Cautious Actor-Critic

Lingwei Zhu[*]   lingwei.andrew.zhu@gmail.com
Toshinori Kitamura[*]   kitamura.toshinori.kt6@is.naist.jp
Takamitsu Matsubara   takam-m@is.naist.jp
Nara Institute of Science and Technology, JAPAN

Editors: Vineeth N Balasubramanian and Ivor Tsang

Abstract

The oscillating performance of off-policy learning and persisting errors in the actor-critic (AC) setting call for algorithms that can conservatively learn to suit the stability-critical applications better. In this paper, we propose a novel off-policy AC algorithm cautious actor-critic (CAC). The name cautious comes from the doubly conservative nature that we exploit the classic policy interpolation from conservative policy iteration for the actor and the entropy-regularization of conservative value iteration for the critic. Our key observation is the entropy-regularized critic facilitates and simplifies the unwieldy interpolated actor update while still ensuring robust policy improvement. We compare CAC to state-of-the-art AC methods on a set of challenging continuous control problems and demonstrate that CAC achieves comparable performance while significantly stabilizes learning.

Keywords: Cautious Reinforcement Learning; Policy Oscillation; Monotonic Improvement; Entropy Regularized Markov Decision Process;

1. Introduction

Actor-critic (AC) methods of reinforcement learning (RL) have been gaining increasing interests recently due to their scalability to large-scale problems: they can learn with both on-policy or/and off-policy samples and handle continuous action spaces (Lillicrap et al., 2015; Schulman et al., 2015); both in model-free or model-based setting (Haarnoja et al., 2018; Hafner et al., 2020). Recently in the model-free setting there has seen a booming in off-policy AC methods (Wang et al., 2017; Haarnoja et al., 2018; Fakoor et al., 2020). However, while these methods are sample-efficient in exploiting off-policy samples for continuous control, it is those samples that often bring oscillating performance during learning as a side-effect due to distribution mismatch. The oscillating performance of off-policy learning and persisting errors in the AC setting (Fujimoto et al., 2018) call for algorithms that can conservatively learn to better suit the stability-critical applications.

The performance oscillation and degradation problems have been widely discussed in the approximate dynamic programming (ADP) literature (Wagner, 2011; Bertsekas, 2011) that has motivated efficient learning algorithms against various sources of error. The seminal work of (Kakade and Langford, 2002) propose a principled approach to tackle performance degradation by leveraging policy interpolation which is *conservative* in that it reduces greediness of the updated policy. However, though it enjoys strong theoretical guarantees, its

---

[*]Equal contribution

drawbacks limit its use in the AC setting: (1) it is difficult to obtain a reliable reference policy in high-dimensional continuous state-action spaces; (2) the interpolation is often regarded inconvenient to use and it is unclear how to design the interpolation in continuous action spaces. In practice, two popular variants (Schulman et al., 2015, 2017) that sidestep the interpolation and directly approximate the updated policy are more often used in the AC setting. On the other hand, the recently booming entropy-regularized ADP literature (Azar et al., 2012; Fox et al., 2016; Kozuno et al., 2019; Vieillard et al., 2020a) also features *conservative learning* (Kozuno et al., 2019) as they average over past value functions (Vieillard et al., 2020a). Though these methods do not explicitly address the performance oscillation problem, they have been empirically verified to be error-tolerant and yield state-of-the-art performance on a wide range of tasks. Extending this conservative learning to AC has been studied (Nachum et al., 2018; Haarnoja et al., 2018). However, the resulted *conservativeness* exists only in the critic: In challenging tasks, the performance degradation and oscillation still occur.

This paper aims to tackle the performance oscillation problem of off-policy AC by proposing a novel algorithm: *cautious actor critic* (CAC), where the naming cautious comes from the *doubly conservative* nature as we combine a conservative actor leveraging the concept of conservative policy iteration (CPI) (Kakade and Langford, 2002) with a conservative critic exploiting the entropy-regularization of conservative value iteration (CVI) (Kozuno et al., 2019). The key observation is that the entropy-regularized critic can find error-tolerant reference policies and simplifies the unwieldy interpolated actor update while still ensures robust policy improvement. CAC leverages automatically adjusted interpolation to reflect the faith during learning: when performance oscillation is likely to happen, CAC behaves cautiously to rely more on validated previous policy rather than on the new policy. Our novel interpolation design is inspired by a very recent study from the ADP literature (Vieillard et al., 2020b) but improved for the continuous AC setting.

The rest of the paper is organized as follows: we provide a short survey on related work in Section 2, followed by the preliminary on RL and relevant AC methods in Section 3. Section 4 presents CAC. Specifically, we discuss our novel design of the interpolation scheme which is central to CAC. We evaluate CAC in Section 5 on a set of challenging continuous control problems. We show that CAC is capable of achieving performance comparable to the state-of-the-art AC methods while significantly stabilizing learning. Ablation study has been conducted to distinguish CAC from existing methods. Discussion and conclusion are given in Section 6. Due to page limits, we present derivations and implementation details in the Appendix of supplemental file.

2. Related Work

It has been noticed that various sources of error such as approximation error in AC algorithms (Fujimoto et al., 2018; Fu et al., 2019) are the cause of performance degradation and oscillation during learning. In this section, we briefly survey some related works (partially) tackling this problem and outline our contributions.

Robust AC. Algorithms learning from off-policy data are sample-efficient but are also at the risk of divergence. To solve the divergence problem, an approach is to incorporate the importance sampling (IS) ratio (Precup et al., 2001). However, the resultant

algorithms typically have large variance as the IS ratio is the product of many potentially unbounded terms. Munos et al. (2016) proposed to clip the IS ratio and proved the resulting algorithm Retrace (λ) can attain the globally optimal policy. Retrace (λ) has motivated recent successful AC methods (Wang et al., 2017; Fakoor et al., 2020) that exploit both on- and off-policy samples for better stability while retaining sample-efficiency. The robustness comes from that any off-policy samples can be used without causing divergence and wild variance thanks to the clipping. However, one still has to trade off the learning speed with learning stability by the user-defined clipping threshold. If we favor more learning stability, the agent might fail to learn meaningful behaviors.

Entropy-regularized AC. The recently booming entropy-regularized ADP literature has established that by augmenting the reward with Shannon entropy, the optimal policy is multi-modal and hence robust against adversarial settings (Nachum et al., 2017; Haarnoja et al., 2017; Ahmed et al., 2019). Another popular candidate entropy is the relative entropy or Kullback-Leibler (KL) divergence that renders the optimal policy an average of all past value functions (Azar et al., 2012; Fox et al., 2016; Kozuno et al., 2019; Vieillard et al., 2020a), which is more conservative and robust under mild assumptions such as the sequence of errors is martingale difference under the natural filtration (Azar et al., 2012). These methods have been extended to the AC setting including state-of-the-art (Haarnoja et al., 2018) that exploits the Shannon entropy and (Nachum et al., 2018) that leverages the KL divergence. Those methods demonstrate strong empirical performance on a wide range of tasks. However, it should be noted that the conservativeness brought by the entropy-regularized reward augmentation exists only in the critic. Since the average is prone to outliers, performance degradation can still happen if the value function estimates at some iterations are poor.

Conservative Policy Iteration. Tackling the performance oscillation problem has been widely discussed in the ADP literature (Wagner, 2011; Bertsekas, 2011), of which the seminal CPI algorithm (Kakade and Langford, 2002) has inspired many conservative learning schemes with strong theoretical guarantees for per-update improvement (Pirotta et al., 2013; Abbasi-Yadkori et al., 2016). However, CPI has seen limited applications to the AC setting due to two main drawbacks: (1) it assumes a *good* reference policy that is typically difficult to obtain in high-dimensional continuous state-action spaces; (2) the interpolation coefficient that interpolates the reference policy and current policy depends on the horizon of learning, which is typically short in ADP scenarios. In the AC setting featuring long learning horizon, this coefficient becomes vanishingly small and hence significantly hinders learning. A very recent work extended CPI to learning with deep networks and has demonstrated good performance on Atari games (Vieillard et al., 2020b). However, it is limited to discrete action spaces while our method mainly focuses on continuous action spaces and can be easily adapted to discrete action setting. The above-mentioned drawbacks render CPI generally perceived as unwieldy (Schulman et al., 2015).

Trust-region Methods. Motivated by the above-mentioned drawbacks of CPI, two popular variants trust region policy optimization (TRPO) (Schulman et al., 2015) and its improved version proximal policy optimization (PPO) (Schulman et al., 2017) sidestep the interpolation and directly approximate the resultant conservative policy. TRPO and PPO are welcomed choices for learning from scratch when the reference policy is unavailable or unreliable, but they also ignore this knowledge when we have a good reference policy at our

disposal. Further, TRPO and PPO require on-policy samples which are expensive since all samples can be used only once and then discarded.

Contribution. The main contributions of this paper are:

- CAC, the first off-policy AC method applying the interpolation of CPI-based algorithms for stabilizing off-policy learning to the best of the authors' knowledge.

- A novel interpolation coefficient design suitable for high dimensional continuous state-action spaces. Previously there was only a design suitable for discrete action spaces (Vieillard et al., 2020b).

- We evaluate CAC on a set of benchmark continuous control problems and demonstrate that CAC achieves comparable performance with state-of-the-art AC methods while significantly stabilizes learning.

3. Preliminary

3.1. Reinforcement Learning

RL problems are mostly formulated by Markov Decision Processes (MDPs) defined by the tuple $(\mathcal{S}, \mathcal{A}, P, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the (possibly continuous) action space, $P(s' \mid s, a)$ is the transition probability from s to s' under action a taken; $\mathcal{R}$ is the reward function with $\mathcal{R}(s, a) \in [-r_{\max}, r_{\max}]$ denoting the immediate reward associated with that transition. We also use r_t as a shorthand for $\mathcal{R}(s_t, a_t)$ at t-th step. $\gamma \in (0, 1)$ is the discount factor. We define $\mathbf{P}$ as a left operator such that $(\mathbf{P}V)(s, a) = \sum_{s'} P(s' \mid s, a) V(s')$ for some V. A policy π maps states to a probability distribution over the action space. We define the stationary state distribution induced by π as the unnormalized occupancy measure $d^\pi(s) = \sum_{t=0}^{\infty} \gamma^t P(s_t = s \mid \pi)$. The goal of RL is to find an optimal policy π^* that maximizes the long term discounted rewards $J^\pi = \mathbb{E}^{d^\pi} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right]$. Equivalently, this optimal policy also maximizes the state value function for all states s:

$$\pi^* = \arg\max_\pi V^\pi(s) = \arg\max_\pi \mathbb{E}^{d^\pi} \left[\sum_{t=0}^{\infty} \gamma^t r_t \big| s_0 = s \right].$$

The state-action value counterpart Q^{π^*} is more frequently used in the control context:

$$Q^{\pi^*}(s, a) = \max_\pi \mathbb{E}^{d^\pi} \left[\sum_{t=0}^{\infty} \gamma^t r_t \big| s_0 = s, a_0 = a \right].$$

We define the advantage function for a policy π as $A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s)$, and the expected advantage function of policy π' over π as:

$$A_\pi^{\pi'}(s) = \mathbb{E}^{\pi'} [Q^\pi(s, a)] - V^\pi(s).$$

3.2. Actor Critic methods

In this section we briefly introduce recent actor-critic algorithms and discuss their pros and cons and shed light on our proposal in Section 4.

3.2.1. TRUST REGION METHODS

TRPO exploits the policy improvement lemma of (Kakade and Langford, 2002) for ensuring approximately monotonic policy improvement. However, unlike in (Kakade and Langford, 2002) that at k-th iteration the policy is updated as $\pi_{k+1} = \zeta\pi' + (1 - \zeta)\pi_k$, where π' is the greedy policy; TRPO constructs an algorithm that directly computes π_{k+1} without resorting to π'. Specifically, TRPO has the following update rule:

$$
\begin{aligned}
J_{\pi_k}^{\mathrm{TRPO}}(\pi) &:= \arg\max_{\pi} \mathbb{E}^{\pi,d^{\pi_k}}\left[A^{\pi_k}\right], \\
\text{subject to} \quad & C_\gamma \, \Delta_\pi \, D_{KL}^{\max}(\pi_k\|\pi) \leq \delta, \\
\text{with} \quad \Delta_\pi &= \max_{s,a}|A^\pi(s,a)|,
\end{aligned}
\tag{1}
$$

where C_γ is a horizon-dependent constant, D_{KL} is the KL divergence and δ is the trust region parameter.

As computing $J_{\pi_k}^{\mathrm{TRPO}}(\pi)$ requires sampling according to the stationary distribution d^{π_k}, it is inherently an on-policy algorithm, which is not sample-efficient as the samples can only be used only once and discarded.

3.2.2. OFF-POLICY MAXIMUM ENTROPY ACTOR-CRITIC

As state-of-the-art model-free off-policy AC algorithm, soft actor-critic (SAC) (Haarnoja et al., 2018) maximizes not only task reward but also the Shannon entropy of policy. The entropy term in the reward function renders the optimal policy multi-modal as opposed to deterministic policies of algorithms that solely maximize task reward, which is beneficial due to the multi-modality (Haarnoja et al., 2017) and has demonstrated superior sample-efficiency due to more effective exploration of the state-action spaces. Writing in the ADP manner, SAC has the following update rule (we drop the state-action pair for the Q function for simplicity):

$$
\begin{cases}
\pi_{k+1} \leftarrow \arg\max_\pi \mathbb{E}^\pi\left[Q_{\mathcal{H}}^{\pi_k}(s,a) + \kappa\mathcal{H}\left(\pi(\cdot|s_t)\right)\right] \\
Q_{\mathcal{H}}^{\pi_{k+1}} \leftarrow \mathcal{R}(s,a) + \gamma\left(\mathbf{P}V_{\mathcal{H}}^{\pi_k}\right)(s,a)
\end{cases}
\tag{2}
$$
$$
\text{where } V_{\mathcal{H}}^\pi(s) = \sum_{t\geq 0}\gamma^t\mathbb{E}^\pi\left[\mathcal{R}(s_t,a_t) + \kappa\mathcal{H}\left(\pi(\cdot|s_t)\right)\Big|s_0 = s\right].
$$

$\mathcal{H}(\pi) := -\sum_a \pi(a|s)\log\pi(a|s)$ denotes the Shannon entropy of policy π, κ denotes the weighting coefficient and $V_{\mathcal{H}}^\pi$ denotes the soft value function when regularized with the Shannon entropy. SAC performs one step look-ahead for updating the actor, where states are randomly sampled from a replay buffer, and then actions are generated by feeding the states into the policy network (Haarnoja et al., 2018). As such, SAC does not need an IS ratio, but it has been demonstrated that SAC often oscillates wildly in performance.

4. Cautious Actor Critic

In this section we present CAC, an off-policy actor-critic method capable of learning conservatively against performance oscillation and degradation.

4.1. CAC Algorithm

For the ease of understanding, we write CAC in the following approximate policy iteration style (Vieillard et al., 2020a). Specifically, the first step corresponds to the policy (actor) improvement and the last step corresponds to the interpolation:

$$
\text{CAC} \begin{cases} \pi_{k+1} \leftarrow \arg\max_\pi \mathbb{E}^\pi\left[Q_{\mathcal{I}}^{\pi_k}(s,a) + \mathcal{I}_{\pi_k}^\pi(s)\right] \\ Q_{\mathcal{I}}^{\pi_{k+1}} \leftarrow \mathcal{R}(s,a) + \gamma\left(\mathbf{P}V_{\mathcal{I}}^{\pi_{k+1}}\right)(s,a) \\ \zeta \leftarrow (\tilde{\Delta}_{\pi_k}^{\pi_{k+1}})^{-1}\left(\mathbb{E}^{\pi_{k+1},\mathcal{B}_K}\left[A^{\pi_k}(s,a)\right]\right) \\ \tilde{\pi}_{k+1} \leftarrow \zeta\pi_{k+1} + (1-\zeta)\pi_k \end{cases}
$$

$$
\text{with } V_{\mathcal{I}}^{\pi_{k+1}}(s) = \sum_{t\geq 0}\gamma^t\mathbb{E}^\pi\left[\mathcal{R}(s_t,a_t) + \mathcal{I}_{\pi_k}^{\pi_{k+1}}(s_t)\,\Big|\,s_0 = s\right],
$$

$$
\mathcal{I}_{\pi_k}^{\pi_{k+1}}(s) = \mathbb{E}^{\pi_{k+1}}\left[-\kappa\log\pi_{k+1}(a|s) - \tau\log\frac{\pi_{k+1}(a|s)}{\pi_k(a|s)}\right],
$$

where ζ is the interpolation coefficient computed by ζ^* in Eq. (7), $\mathcal{B}_K$ denotes the on-policy replay buffer, with K indicating the number of steps up to now. κ, τ denote the Shannon entropy and KL divergence regularization coefficient, respectively. $Q_{\mathcal{I}}, V_{\mathcal{I}}$ (and hence $A_{\mathcal{I}}$) denote the entropy-regularized value functions. For uncluttered notations, in the rest of the paper we drop the subscript $\mathcal{I}$. Except the computation of ζ, all other steps are computed using samples from the off-policy replay buffer $\mathcal{B}$. For later convenience, we define $\alpha := \frac{\kappa}{\kappa+\tau}$ and $\beta := \frac{1}{\kappa+\tau}$. Note our use of both on- and off-policy replay buffers renders CAC similar in spirit to (Gu et al., 2017; Wang et al., 2017; Fakoor et al., 2020).

We first compute the greedy policy π_{k+1}. Due to the Fenchel conjugacy (Geist et al., 2019), when $\mathcal{I}_{\pi_k}^{\pi_{k+1}}$ is included in the $\arg\max$, the maximizer policy can be analytically derived as $\pi_{k+1}(a|s) \propto \pi_k^\alpha(a|s)\exp\left(\beta Q_{\pi_k}(a|s)\right)$ (Kozuno et al., 2019). Then the entropy-regularized action value function $Q^{\pi_{k+1}}$ is evaluated. In the third step, we use the on-policy replay buffer $\mathcal{B}_K$ for computing ζ as described in Eq. (7). Finally, the optimal policy in the sense of guaranteeing policy improvement is obtained by interpolating π_{k+1} with π_k. We present the following theorem of CAC convergence in the tabular case.

Theorem 1 *Repeated application of CAC Eq. (3) on any initial policy π will make it converges to the entropy regularized optimal policy $\pi^*(a|s) = \frac{\exp\left(\frac{1}{\kappa}Q^*(s,a)\right)}{\int_{a\in\mathcal{A}}\exp\left(\frac{1}{\kappa}Q^*(s,a)\right)}$.*

Proof See Section A.1 in the Appendix. ∎

From Theorem 1 we see the optimal policy and corresponding optimum of the MDP is biased by the choice of κ. If we gradually decay the value of κ then we recover the optimum of the non-regularized MDP Vieillard et al. (2020a). In the following sections, we describe in detail the CAC actor and critic, as well as the derivation of actor gradient expression and practical interpolation coefficient ζ design.

4.1.1. CONSERVATIVE ACTOR

As discussed in Section 3.2.1, at k-th iteration TRPO directly constructs a new policy π by maximizing $J_{\pi_k}^{\text{TRPO}}(\pi)$. This is useful if the agent learns from scratch, but it discards the

reference policy when available. On the other hand, we follow the exact form of (Kakade and Langford, 2002) by taking the information of reference policy into account, where we choose π_{k+1} to be the reference policy π':

$$\tilde{\pi}_{k+1} = \zeta\pi_{k+1} + (1-\zeta)\pi_k. \tag{4}$$

Our objective function $J_{\pi_k,\pi_{k+1}}^{\mathrm{CAC}}(\pi)$ explicitly features the knowledge of the reference policy (Pirotta et al., 2013). Specifically, the objective can be lower-bounded as:

$$
\begin{aligned}
J_{\pi_k,\pi_{k+1}}^{\mathrm{CAC}}(\pi) &:= \mathbb{E}^{\pi_{k+1},d^{\pi_{k+1}}}\left[A^{\pi_k}(s,a)\right] \\
&\geq C'_\gamma \, (v\,\tilde{\Delta}_{\pi_k}^{\pi_{k+1}})^{-1}\left(\mathbb{E}^{\pi_{k+1},d^{\pi_k}}\left[A^{\pi_k}(s,a)\right]\right)^2, \\
\text{given } \zeta^* &= 2\,C'_\gamma(v\,\tilde{\Delta}_{\pi_k}^{\pi_{k+1}})^{-1}\left(\mathbb{E}^{\pi_{k+1},d^{\pi_k}}\left[A^{\pi_k}(s,a)\right]\right), \\
\tilde{\Delta}_{\pi_k}^{\pi_{k+1}} &= \max_{s,s'}\left|A_{\pi_k}^{\pi_{k+1}}(s) - A_{\pi_k}^{\pi_{k+1}}(s')\right|, \\
v &= \max_s D_{TV}\left(\pi_{k+1}(\cdot|s)\|\pi_k(\cdot|s)\right),
\end{aligned}
\tag{5}
$$

where D_{TV} denotes the total variation. v, $\tilde{\Delta}_{\pi_k}^{\pi_{k+1}}$ and the expectation wrt π_{k+1}, d^{π_k} require estimation. C'_γ absorbs the horizon-dependent constants. Hence, when optimizing the lower bound of $J_{\pi_k,\pi'}^{\mathrm{CAC}}(\pi)$, we can achieve guaranteed improvement.

In the existing literature (Kakade and Langford, 2002; Pirotta et al., 2013; Abbasi-Yadkori et al., 2016), the difficulties of extending Eq. (5) to large-scale problems are: (1) preparing a reliable reference policy in high dimensional continuous state-action spaces is difficult; (2) it is hard to accurately estimate v, the maximum total variation between two policies without enforcing a gradual change of policies, which is absent in these works. On the other hand, naively using $v \leq 2$ as suggested by (Pirotta et al., 2013) often yields vanishingly small ζ, which significantly hinders learning. (3) the horizon-dependent constant C'_γ developed in the classic ADP literature is not suitable for learning with deep networks that feature long horizon of learning. As will be demonstrated in the following sections, we tackle the first and second problems by leveraging entropy-regularized critic, and the third problem via a novel design of ζ inspired by a very recent work for discrete action problems (Vieillard et al., 2020b).

4.1.2. Conservative Critic

In Eq. (5) we see in order to yield a meaningful interpolation coefficient ζ one is required to accurately estimate the maximum total derivation v, which is intractable in high dimensional continuous action spaces. However, by introducing an entropy-regularized critic, we can leverage the following theorem to avoid estimating v:

Theorem 2 *(Kozuno et al., 2019, Proposition 3) For any two consecutive entropy-regularized policies π_k, π_{k+1} generated by Eq. (3), the following bound for their maximum total deviation holds:*

$$
\begin{aligned}
\max_s D_{TV}\left(\pi_{k+1}(\cdot|s)\|\pi_k(\cdot|s)\right) &\leq \sqrt{4B_K + 2C_K}, \\
\text{where } B_k &:= \frac{1-\gamma^k}{1-\gamma}\epsilon\beta, \quad C_k := r_{max}\beta\sum_{j=0}^{k-1}\alpha^j\gamma^{k-j-1},
\end{aligned}
\tag{6}
$$

ϵ *is the uniform upper bound of errors.*

Recall from Section 4.1 that $\alpha := \frac{\kappa}{\kappa+\tau}, \beta := \frac{1}{\kappa+\tau}$. Specifically, it has been proved in (Kozuno et al., 2019) that this bound is non-improvable, i.e. there exists an MDP such that the inequality becomes equality.

By leveraging an entropy-regularized critic, the objective in Eq. (5) becomes:

$$
\begin{aligned}
J^{\text{CAC}}_{\pi_k,\pi_{k+1}}(\pi) &\geq C'_\gamma C_k \, (\tilde{\Delta}^{\pi_{k+1}}_{\pi_k})^{-1} \left(\mathbb{E}^{\pi_{k+1},d^{\pi_k}} \left[A^{\pi_k}(s,a) \right] \right)^2, \\
\text{given } \zeta^* &= 2\, C'_\gamma C_k (\tilde{\Delta}^{\pi_{k+1}}_{\pi_k})^{-1} \left(\mathbb{E}^{\pi_{k+1},d^{\pi_k}} \left[A^{\pi_k}(s,a) \right] \right),
\end{aligned}
\tag{7}
$$

where C'_γ absorbs horizon-dependent constants and C_k is from Theorem 2. Estimating the optimal ζ^* now requires estimating the expectation wrt π_{k+1}, d^{π_k} and $\tilde{\Delta}^{\pi_{k+1}}_{\pi_k}$ which have been studied by (Vieillard et al., 2020b).

The KL divergence also manifests its importance for generating reasonable reference policies even for high dimensional or continuous action problems. Consider the following upper bound due to (Vieillard et al., 2020a) where reward is augmented by the KL divergence:

$$
\| Q^* - Q^{\pi_{k+1}} \|_\infty \leq \frac{2}{1-\gamma} \left\| \frac{1}{k} \sum_{j=0}^{k} \epsilon_j \right\|_\infty + \frac{4}{1-\gamma} \frac{V_{max}}{k},
$$

where ϵ_j are errors and $V_{max} = \frac{r_{\max}}{1-\gamma}$. By comparing it with the non-improvable approximate modified policy iteration (AMPI) bound where the reward is not augmented (Scherrer et al., 2015):

$$
\| Q^* - Q^{\pi_{k+1}} \|_\infty \leq \left((1-\gamma) \sum_{j=1}^{k} \| \epsilon_j \|_\infty \right) + \frac{2\gamma^{k+1}}{1-\gamma} V_{max},
$$

we see that the error term for the KL regularization case is sup-over-sum. Under mild assumptions such as ϵ_j are iid distributed under the natural filtration (Azar et al., 2012), the summation over errors asymptotically cancels out. On the other hand, the error term for AMPI depends on the summation of *maximum* of every iteration, which is typically large. Further, the dependence of error on the horizon is linear $\frac{1}{1-\gamma}$ rather than quadratic, which is a significant improvement as typically $\gamma \approx 1$.

4.1.3. Network Optimization Perspective

Given the above ADP-style characterization for both the actor and the critic, we now examine Eq. (3) from the optimization perspective. Suppose the critic is parametrized by a network with parameters θ and the actor by a network with parameters ϕ. CAC updates the network weights θ, ϕ by solving the following minimization problems:

$$
y = r + \gamma \left(\mathbb{E}^{a \sim \pi_\phi} \left[Q_{\bar\theta}(s',a) \right] + \mathcal{I}^{\pi_\phi}_{\pi_{\bar\phi}}(s') \right),
\tag{8a}
$$

$$
\theta \leftarrow \arg\min \mathbb{E}^{\mathcal{B}} \left[(Q_\theta(s,a) - y)^2 \right],
\tag{8b}
$$

$$
\phi \leftarrow \arg\min \mathbb{E}^{\mathcal{B}} \left[D_{KL} \left(\pi_\phi \| (1-\zeta)\pi_{\bar\phi} + \zeta \mathcal{G}_{\pi_{\bar\phi}} Q_\theta \right) \right],
\tag{8c}
$$

where the update of ϕ corresponds to solving an *information projection* problem. This is because policies π_{k+1}, π_k are Boltzmann softmax (Geist et al., 2019) but their summation is generally not Boltzmann, which might results in loss of desirable properties. By the following theorem, in the ideal case we can find a Boltzmann policy π_ϕ that perfectly represents the interpolation by solving the information projection problem.

Theorem 3 *(Ziebart, 2010, Theorem 2.8) Let $\pi^{(1)}$, $\pi^{(2)}$, $\ldots, \pi^{(n)}$ be an arbitrary sequence of policies and $\zeta_1, \ldots \zeta_n$ be a sequence of numbers such that $\zeta_i \geq 0, \forall i$, $\sum_{i=1}^n \zeta_i = 1$. Then the policy π' defined by:*

$$\pi'(a|s) := \frac{\sum_{i=1}^n \zeta_i \, P(\mathcal{S} = s, \mathcal{A} = a|\pi^{(i)})}{\sum_{i=1}^n \zeta_i \, P(\mathcal{S} = s|\pi^{(i)})} \tag{9}$$

has same expected number of state-action occurrences when the denominator is nonzero.

In implementation as the states are sampled from the replay buffer, there is an error term in this information projection step. Taking the above information projection into account, we elaborate upon gradient expression of the actor via the following proposition:

Proposition 4 *Let the actor network be parametrized by weights ϕ and critic by θ. Define $\mathcal{G}Q_{\bar{\phi},\theta}$ as the greedy policy with respect to the CAC critic. The subscript $\bar{\phi}$ comes from the baseline policy introduced by KL divergence. Then the gradient of the actor update can be expressed as:*

$$\nabla_\phi \mathbb{E}^{s \sim \mathcal{B}}_{a \sim \pi_\phi} \left[D^\phi_{\bar{\phi}} - \frac{\beta}{1 + \mathcal{X}} Q_\theta(s, a) \right],$$
$$\text{where } D^\phi_{\bar{\phi}} = \log \pi_\phi(a \mid s) - \frac{\alpha + \mathcal{X}}{1 + \mathcal{X}} \log \pi_{\bar{\phi}}(a; s) \tag{10}$$
$$\mathcal{X} = \frac{1 - \zeta}{\zeta} \cdot \frac{\pi_{\bar{\phi}}(a \mid s)}{\mathcal{G}Q_{\bar{\phi},\theta}(a \mid s)}.$$

Proof See Section A.2 in the Appendix. ∎

This gradient expression is similar to SAC (Haarnoja et al., 2018) which is off-policy since states s are sampled from the off-policy replay buffer $\mathcal{B}$. However, CAC has the term $\log \pi_{\bar{\phi}}(a; s)$ from the KL regularization. The term $\mathcal{X}$ in both the Q_θ and $\log \pi_{\bar{\phi}}(a; s)$ involves ζ that encodes the information for guiding the gradient to *cautiously* learn.

4.2. Design of Interpolation Coefficient

One of the main difficulties to extending CPI to learning with deep networks is that ζ becomes vanishingly small due to the typically long horizon in the AC setting. To tackle this problem, (Vieillard et al., 2020b) propose to heuristically design ζ to be a non-trivial value, which features the consideration of *moving averages*.

Recall that in Eq. (7), ζ is computed by a function of the form:

$$\zeta = C'_\gamma C_k \frac{\mathbb{E}^{\pi_{k+1}, d^{\pi_k}} \left[A^{\pi_k}(s, a) \right]}{\tilde{\Delta}^{\pi_{k+1}}_{\pi_k}},$$
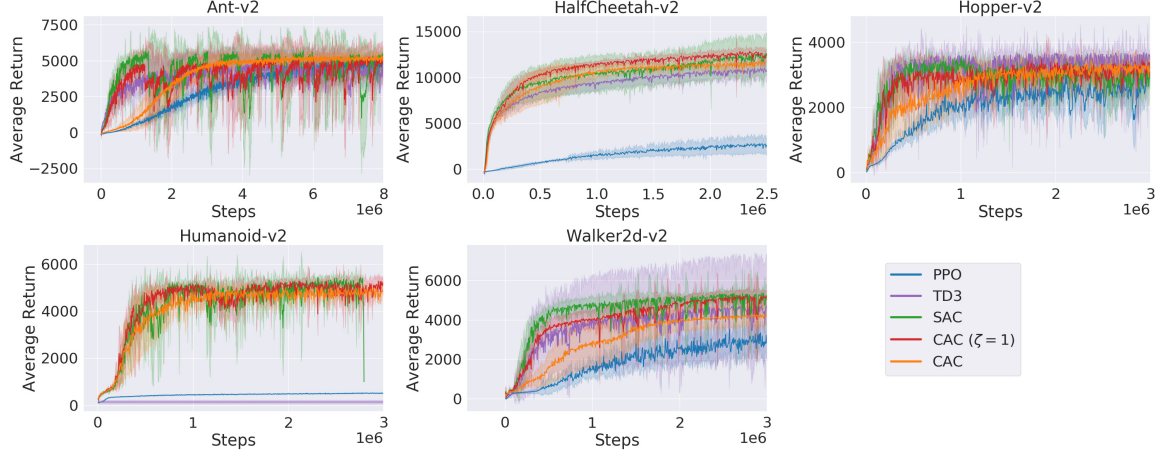
Figure 1: Training curves on the continuous control benchmark problems. The solid curves show the mean and shaded regions the standard deviation over the five independent trials. In all tasks CAC achieved comparable performance but significantly stabilized learning.

where C'_γ absorbs horizon-dependent constants and C_k is defined in Eq. (6). We propose to remove C'_γ since it tends to zero as the horizon increases. Recall also that $\tilde{\Delta}_{\pi_k}^{\pi_{k+1}}$ is the maximum difference of the expected advantage function defined in Eq. (5). We propose the following novel ζ design:

$$\zeta^{\text{CAC}} = \text{clip}\left(\frac{\tilde{\mathbb{A}}}{|\tilde{\mathbb{A}}^{\text{MaxDiff}}|}, 0, \ 1\right), \tag{11}$$

where by following the moving average concept of (Vieillard et al., 2020b), we update $\tilde{\mathbb{A}}$ and $\tilde{\mathbb{A}}^{\text{MaxDiff}}$ as:

$$
\begin{aligned}
M &= \mathbb{E}^{\substack{s \sim \mathcal{B}_K \\ a \sim \mathcal{G}Q^{\pi_k}}}\left[A^{\pi_k}(s,a)\right] \\
\tilde{\mathbb{A}} &\leftarrow \begin{cases} c, & \text{if } M \leq 0 \\ (1 - \nu_A)\,\tilde{\mathbb{A}} + \nu_A M, & \text{else} \end{cases} \\
\tilde{\mathbb{A}}^{\text{MaxDiff}} &\leftarrow \left(1 - \nu_{A^{\text{MaxDiff}}}\right)\tilde{\mathbb{A}}^{\text{MaxDiff}} + \nu_{A^{\text{MaxDiff}}}\, M.
\end{aligned} \tag{12}
$$

Here, M is the current estimate of $\mathbb{E}^{\pi_{k+1},d^{\pi_k}}\left[A^{\pi_k}(s,a)\right]$. We propose to set an if-else judgement here, as $M < 0$ indicates the updated policy has worse performance than the current policy, we let $\tilde{\mathbb{A}}$ be a negative value c, hence enforcing $\zeta = 0$. This information is incorporated into $\tilde{\mathbb{A}}$ by exponential moving average with the previous estimates. $\tilde{\mathbb{A}}^{\text{MaxDiff}}$ attempts to approximate the maximum difference $\tilde{\Delta}_{\pi_k}^{\pi_{k+1}}$. $\mathcal{B}_K$ is a FIFO replay buffer storing K on-policy samples, and $\nu_A, \nu_{A^{\text{MaxDiff}}} \in [0,1]$ are the hyperparameters controlling the average. Following (Vieillard et al., 2020b), it is beneficial to have $\nu_{A^{\text{MaxDiff}}} \leq \nu_A$ for smooth learning.

Computing ζ in Eq. (11) using the moving average in Eq. (12) is in spirit similar to (Vieillard et al., 2020b). However, they focus on general stationary policies. As $\tilde{\mathbb{A}}$ is an off-policy estimate of the on-policy term $\mathbb{E}^{\pi_{k+1},d^{\pi_k}}\left[A^{\pi_k}(s,a)\right]$, it might corrupt the improvement

Table 1: The performance oscillation values of all algorithms for all environments. The bold numbers indicate the smallest performance oscillation values. × indicates the algorithm failed to learn meaningful behaviors. CAC recorded the smallest performance oscillation values for all the environments. PPO is the only on-policy algorithm in the comparison.

| | $\|\mathcal{O}J\|_\infty$ | | | | | $\|\mathcal{O}J\|_2$ | | | | |
	PPO	TD3	SAC	CAC $(\zeta = 1)$	CAC	PPO	TD3	SAC	CAC $(\zeta = 1)$	CAC
ANT	1979	4979	7793	7160	**1811**	359	510	642	591	**297**
HALFCHEETAH	×	2337	3717	4200	**1870**	×	331	425	397	**286**
HOPPER	**1598**	3515	2598	2944	1944	318	609	454	394	**279**
HUMANOID	×	×	4115	3092	**2199**	×	×	645	436	**313**
WALKER2D	1673	3729	4577	4310	**1345**	330	461	499	334	**183**

guarantee. On the other hand, we focus on entropy-regularized policies which allow one to bound the performance loss of leveraging off-policy estimate $\tilde{\mathbb{A}}$ (Zhu and Matsubara, 2020, Theorem 3).

5. Experiments

As CAC combines concepts from ADP literature such as KL regularization and conservative learning that have not seen applications in AC, it is interesting to examine the combination against existing AC methods in challenging tasks. We choose a set of high dimensional continuous control tasks from the OpenAI gym benchmark suite (Brockman et al., 2016).

For comparison, we compare CAC with twin delayed deep deterministic policy gradient (TD3) (Fujimoto et al., 2018) that comprehensively surveys the factors causing poor performance of actor-critic methods, to examine the cautious learning mechanism. As CAC is based on the CPI that has also inspired TRPO and PPO, we compare it with PPO which is improved over TRPO (Schulman et al., 2017). As PPO does not involve computing ζ, we include the curves when $\zeta = 1$. We also compare with SAC (Haarnoja et al., 2018) which has similar architecture.

To better illustrate and quantify the *stability* during learning, we follow (Zhu and Matsubara, 2020) to define the measure of performance oscillation:

$$\begin{aligned} &\forall k, \text{ such that } R_{k+1} - R_k < 0 \\ &\|\mathcal{O}J\|_\infty = \max_k |R_{k+1} - R_k|, \\ &\|\mathcal{O}J\|_2 = \sqrt{\tfrac{1}{N} \sum_{k=1}^N (R_{k+1} - R_k)^2}, \end{aligned} \tag{13}$$

where N is the steps of the learning and R_k refers to the cumulative reward reported at k-th evaluation. Intuitively, $\|\mathcal{O}J\|_\infty$ and $\|\mathcal{O}J\|_2$ measure the maximum and average degradation during learning, respectively.

5.1. Comparative Evaluation

We run all algorithms with the same set of hyperparameters listed in Section A.4 of the Appendix. All figures are plotted with statistics from 10 different random seeds, with each performing 10 evaluation rollouts every 5000 environment steps.

Figure 1 shows the learning curves of the algorithms. CAC achieved comparable performance with other AC algorithms while significantly stabilized learning curves. PPO's learning speed was the slowest among all algorithms on all environments, due to the on-policy nature of PPO which is sample-inefficient. Other methods were able to leverage off-policy samples to quickly learn meaningful behaviors. However, the fast learning came at a cost: except the relatively simple `HalfCheetah-v2`, on all environments these off-policy algorithms oscillated wildly, especially on the challenging `Humanoid-v2` where both PPO and TD3 failed to learn any meaningful behaviors, and the performance of SAC, CAC with $\zeta = 1$ degraded frequently. On the other hand, CAC traded off a little bit slower learning for stability, exhibiting smooth curves. Indeed, the convergence rate of CAC is $O\left(e^{-(1-\gamma)\sum_{j=1}^{k}\zeta_j}\right)$ (Vieillard et al., 2020b), which emphasizes stability more as $\zeta \to 0$.

The comparison on stability of the algorithms can be seen from the Table 1 that summarized the values of $\|\mathcal{O}J\|_\infty$ and $\|\mathcal{O}J\|_2$ for all algorithms. It provided empirical support as CAC showed least oscillation during learning. This is in contrast to other off-policy algorithms oscillated wildly during learning. Since CAC with $\zeta = 1$ still showed huge oscillation, it can be concluded that the mixture coefficient introduced in CAC is effective in preventing significant policy degradation.

5.2. Ablation study on mixture coefficient

In this section we conduct an ablation test to study the effectiveness of CAC as well as the proposed ζ design in Section 4.2. We compare the following setup:

1. **Full.** This is CAC with the proposed ζ^{CAC} in Eq. (12). The curve is same as in Figure 1.

2. **No KL.** We remove KL regularization from Eq. (3). This corresponds to SAC with the cautious actor.

3. **DSPI.** This corresponds to *Deep Safe Policy Iteration* (Zhu and Matsubara, 2020) that uses the ζ suggested by (Vieillard et al., 2020b).

4. **No $\mathcal{B}_K$.** We replace the on-policy replay buffer $\mathcal{B}_K$ with off-policy $\mathcal{B}$ as suggested by (Vieillard et al., 2020b).

5. **No If-Else.** This corresponds to removing the `if` term in Eq. (12) and learns with only the `else` condition.

As is obvious from Figure 2, while removing the `if-else` judgement accelerated learning, it ignored the warning from $M < 0$ that the updated policy was poorer. The consequent curve oscillated drastically as the result of aggressive ζ in the bottom image.

Removing KL regularization induced learning curve similar with **Full** in the beginning, but the performance degraded significantly since the middle stage and failed to recover. This is probably due to the policies were corrupted by error.
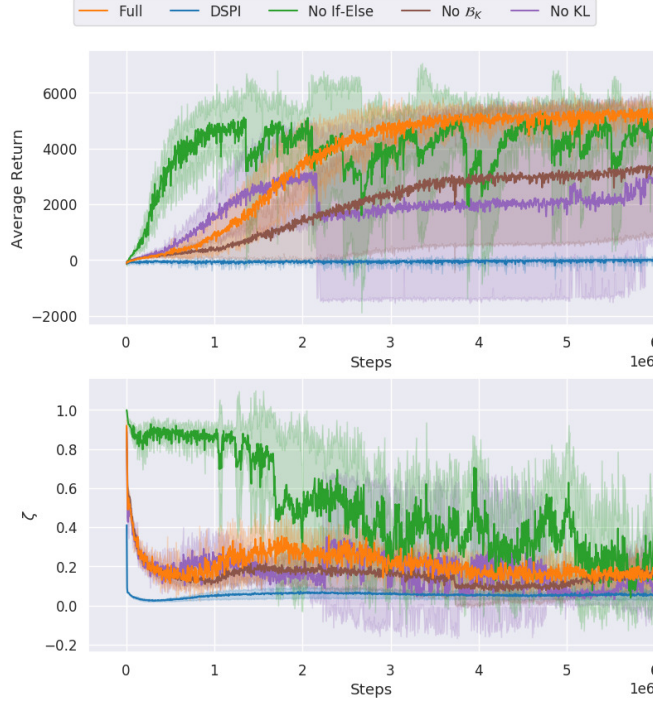
Figure 2: **Top:** The learning curves of derived methods of CAC on Ant-v2. **Bottom:** The value of the mixture coefficient ζ.

Using off-policy replay buffer $\mathcal{B}$ demonstrated stable learning. This is expected as the agent was forced to learn cautiously by the CAC mechanism. On the other hand, the learning was slow as off-policy samples were not as informative as on-policy ones, as we observed small ζ values in the bottom figure.

It is most interesting to examine the DSPI case where ζ was set according to the suggestion of (Vieillard et al., 2020b). Though this scheme works well in Atari games, the resulting algorithm failed to learn any meaningful behaviors in the challenging control tasks with continuous action spaces. This is because they estimate with the entire off-policy replay buffer $\mathcal{B}$ which tends to produce very large estimate of $\tilde{\Delta}_{\pi_k}^{\pi_{k+1}}$, leading to vanishingly small ζ^{DSPI} and subsequent poor performance.

6. Conclusion

We have presented CAC, a novel actor-critic algorithm by introducing several concepts from the approximate dynamic programming literature. The cautiousness in CAC consists in the doubly conservativeness: the actor follows conservative policy iteration (Kakade and Langford, 2002) that ensures monotonic improvement and the critic exploits conservative value iteration (Kozuno et al., 2019; Vieillard et al., 2020a) that has been shown to yield state-of-the-art guarantees in ADP literature.

Our key observation was by introducing an entropy-regularized critic the unwieldy interpolated actor update can be simplified significantly while still ensuring robust policy improvement. CAC performed comparable to the state-of-the-art AC methods while significantly stabilized learning on the benchmark control problems with high dimensional continuous state-action spaces.

An interesting future direction is to incorporate other entropy for different purposes. For example, α-divergence could be used to achieve sparse optimal policies.

Acknowledgments

References

Yasin Abbasi-Yadkori, Peter L. Bartlett, and Stephen J. Wright. A fast and reliable policy improvement algorithm. In *International Conference on Artificial Intelligence and Statistics*, pages 1338–1346, 2016.

Zafarali Ahmed, Nicolas Le Roux, Mohammad Norouzi, and Dale Schuurmans. Understanding the impact of entropy on policy optimization. In *International Conference on Machine Learning*, pages 151–160, 2019.

Mohammad Gheshlaghi Azar, Vicenç Gómez, and Hilbert J Kappen. Dynamic policy programming. *The Journal of Machine Learning Research*, 13(1):3207–3245, 2012.

Dimitri Bertsekas. Approximate policy iteration: A survey and some new methods. *Journal of Control Theory and Applications*, 9:310–335, 2011.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Rasool Fakoor, Pratik Chaudhari, and Alexander J Smola. P3o: Policy-on policy-off policy optimization. In *Conference on Uncertainty in Artificial Intelligence*, pages 1017–1027, 2020.

Roy Fox, Ari Pakman, and Naftali Tishby. Taming the noise in reinforcement learning via soft updates. In *Conference on Uncertainty in Artificial Intelligence*, pages 202–211, 2016.

Justin Fu, Aviral Kumar, Matthew Soh, and Sergey Levine. Diagnosing bottlenecks in deep q-learning algorithms. In *International Conference on Machine Learning*, pages 2021–2030, 2019.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pages 1587–1596, 2018.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized Markov decision processes. In *International Conference on Machine Learning*, pages 2160–2169, 2019.

Shixiang Shane Gu, Timothy Lillicrap, Richard E Turner, Zoubin Ghahramani, Bernhard Schölkopf, and Sergey Levine. Interpolated policy gradient: Merging on-policy and off-policy gradient estimation for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 3846–3855, 2017.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning*, pages 1352–1361, 2017.

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

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint, arXiv:1912.01603*, 2020.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*, pages 267–274, 2002.

Tadashi Kozuno, Eiji Uchibe, and Kenji Doya. Theoretical analysis of efficiency and robustness of softmax and gap-increasing operators in reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2995–3003, 2019.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc Bellemare. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 1054–1062, 2016.

Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. Bridging the gap between value and policy based reinforcement learning. In *Advances in Neural Information Processing Systems 30*, pages 2775–2785. 2017.

Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. Trust-pcl: An off-policy trust region method for continuous control. In *International Conference on Learning Representations*, pages 1–11, 2018.

Matteo Pirotta, Marcello Restelli, Alessio Pecorino, and Daniele Calandriello. Safe policy iteration. In *International Conference on Machine Learning*, pages 307–315, 2013.

Doina Precup, Richard S. Sutton, and Sanjoy Dasgupta. Off-policy temporal difference learning with function approximation. In *International Conference on Machine Learning*, page 417–424, 2001.

Bruno Scherrer and Matthieu Geist. Local policy search in a convex space and conservative policy iteration as boosted policy search. In *Machine Learning and Knowledge Discovery in Databases*, pages 35–50, 2014.

Bruno Scherrer, Mohammad Ghavamzadeh, Victor Gabillon, Boris Lesner, and Matthieu Geist. Approximate modified policy iteration and its application to the game of tetris. *Journal of Machine Learning Research*, 16(1):1629–1676, 2015.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897, 2015.

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

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018.

Nino Vieillard, Tadashi Kozuno, Bruno Scherrer, Olivier Pietquin, Rémi Munos, and Matthieu Geist. Leverage the average: an analysis of kl regularization in reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1–12, 2020a.

Nino Vieillard, Olivier Pietquin, and Matthieu Geist. Deep conservative policy iteration. In *AAAI Conference on Artificial Intelligence*, pages 6070–6077, 04 2020b.

Paul Wagner. A reinterpretation of the policy oscillation phenomenon in approximate policy iteration. In *Advances in Neural Information Processing Systems 24*, pages 2573–2581, 2011.

Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando de Freitas. Sample efficient actor-critic with experience replay. In *International Conference on Learning Representations*, pages 1–13, 2017.

Lingwei Zhu and Takamitsu Matsubara. Ensuring monotonic policy improvement in entropy-regularized value-based reinforcement learning. *arXiv preprint, arXiv:2008.10806*, 2020.

Brian D. Ziebart. *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy*. PhD thesis, Carnegie Mellon University, 2010.