Enhancing Off-Policy Constrained Reinforcement Learning through Adaptive Ensemble C Estimation

Hengrui Zhang[1,2], **Youfang Lin**[1,2], **Shuo Shen**[3], **Sheng Han**[1,2], **Kai Lv**[1,2*]

[1]School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China
[2]Beijing Key Laboratory of Traffic Data Analysis and Mining, Beijing, China
[3]Cooperation Product Department, Interactive Entertainment Group, Tencent
{18112037, yflin, 19140027, shhan, lvkai}@bjtu.edu.cn

Abstract

In the domain of real-world agents, the application of Reinforcement Learning (RL) remains challenging due to the necessity for safety constraints. Previously, Constrained Reinforcement Learning (CRL) has predominantly focused on on-policy algorithms. Although these algorithms exhibit a degree of efficacy, their interactivity efficiency in real-world settings is sub-optimal, highlighting the demand for more efficient off-policy methods. However, off-policy CRL algorithms grapple with challenges in precise estimation of the C-function, particularly due to the fluctuations in the constrained Lagrange multiplier. Addressing this gap, our study focuses on the nuances of C-value estimation in off-policy CRL and introduces the Adaptive Ensemble C-learning (AEC) approach to reduce these inaccuracies. Building on state-of-the-art off-policy algorithms, we propose AEC-based CRL algorithms designed for enhanced task optimization. Extensive experiments on nine constrained robotics tasks reveal the superior interaction efficiency and performance of our algorithms in comparison to preceding methods.

Introduction

Implementing Reinforcement Learning (RL) in real-world tasks presents a formidable challenge, primarily because real-world intelligence systems must comply with myriad safety constraints. For instance, a robotic arm used in an assembly line in manufacturing must work within the constraints of operating at a particular speed, maintaining the integrity of the materials it handles, avoiding collisions with humans and other robots, and adhering to a specified power consumption. Previously, the primary focus of Constrained Reinforcement Learning (CRL) has been on on-policy algorithms (Achiam et al. 2017; Yang et al. 2020; Liu, Ding, and Liu 2020; Zhang, Vuong, and Ross 2020; Yang et al. 2022; Zhang et al. 2022b; Kim and Oh 2022b; Dai et al. 2023; He, Zhao, and Liu 2023). While they have their merits, their interactive efficiency with real-world environments falls significantly short of the demands of practical applications. This inefficiency becomes particularly pronounced when data collection is expensive or risky, as is often the case with real-world robotic tasks. Consequently, there is an urgent need to realize off-policy algorithms that can learn more efficiently from past experiences without the necessity for continuous and costly interaction with the environment.

Designing an off-policy CRL algorithm is generally considered more challenging than designing an on-policy CRL algorithm due to the distinct mechanisms each employs for data handling and policy updates. In on-policy RL, the policy is updated based on the rollout experience obtained during each iteration. This allows the algorithm to directly sample experiences that reflect the current policy. By considering the cumulative discounted cost calculated from this fresh data, it becomes easier to determine whether the current policy violates any given constraint. In contrast, off-policy algorithms utilize a replay buffer to store and sample past experiences. The reliance on older data in the replay buffer makes it more difficult to accurately assess whether the current policy violates the constraint.

One possible solution that has been explored in previous literature is to introduce a C-function (Ma et al. 2021; Yang et al. 2021; Liu et al. 2022; Zhang et al. 2022a), similar to the Q-function, to estimate the cumulative cost and determine whether the current policy satisfies the constraint. However, accurately estimating the C-function presents several challenges. Similar to the estimation of the Q-function, the C-function also involves function approximation, which introduces errors. Additionally, the C-function estimation can result in underestimation or overestimation due to the maximization of the constraint objective.

To resolve this issue, we propose an Adaptive Ensemble C-learning (AEC) technique containing an adaptive variable K denoting the number of C-values. We demonstrate that K can control the deviation between the C-value estimation and its true value. Furthermore, to ensure a precise correspondence between the C-value estimation and the stipulated constraint threshold, we advocate for implementing the First C Buffer (FCB) technique. This approach entails the establishment of a novel buffer pool, archiving the initial state of each episode. Subsequent sampling from this pool enables a refined assessment of whether the current policy aligns with the constraints.

Our main contributions are threefold. Firstly, we underscore the phenomenon in off-policy CRL, where the balance between task goals and constraint goals results in imprecise estimation of the C-value. Subsequently, we intro-

*Corresponding Author: Kai Lv

duce an approach utilizing Adaptive Ensemble C-learning (AEC) to reduce the estimation bias of the C-value. Secondly, we fashion CRL algorithms by extending off-policy algorithms SAC (Haarnoja et al. 2018) and TQC (Kuznetsov et al. 2020). By synergistically integrating AEC and FCB technologies, our algorithms adeptly maximize tasks while operating within prescribed constraints. Lastly, we conduct extensive experimentation on nine distinct robotic constraint optimization tasks. The outcomes signify that our algorithm exhibits superior environmental interaction efficiency and enhanced performance relative to its predecessors.

Related Work

Off-policy Constrained Reinforcement Learning. Exploiting off-policy data in the domain of constrained reinforcement learning (CRL) can substantially enhance the efficiency of environmental interactions. Several research initiatives (Wang et al. 2019; Ha et al. 2020; Peng et al. 2022; Yu, Xu, and Zhang 2022) develop CRL methodologies underpinned by Soft Actor Critic (SAC) method (Haarnoja et al. 2018). These approaches utilize off-policy data in the estimation of safety critics, and apply the Lagrangian approach to echo expectation-based constraints. To reduce the likelihood of constraint violations, some works (Yang et al. 2021; Kim and Oh 2022a; Kim, Lee, and Oh 2023) apply conditional Value-at-Risk to CRL and focus on the upper tail of the cost distribution. CDMPO (Zhang et al. 2022a) learns the full return distribution by using a discrete distribution instead of making the Gaussianity assumption. Taking a different approach, Constrained Variational Policy Optimization (CVPO) (Liu et al. 2022) decomposes the CRL problem into a convex optimization phase with a non-parametric variational distribution and a supervised learning phase.

Overestimation bias. The long-standing challenge of overestimation bias in Q-learning, first highlighted by (Thrun and Schwartz 1993), has been the subject of significant investigation and algorithmic development within the field of RL. The bias is a product of the intrinsic maximization operation in the Q-learning algorithm, which often results in overestimated Q-values, thereby obstructing effective learning. In an effort to mitigate this bias, Double Q-Learning (Hasselt 2010) is proposed to provide a mechanism that reduces maximization bias. The Ensemble Method is believed to improve performance by reducing the variance in Q-value estimation. This has led to the incorporation of ensembles in Q-learning methodologies, such as utilizing the average of multiple Q estimates to reduce variance(Anschel, Baram, and Shimkin 2017), and developing the Random Ensemble Mixture (REM) which enforces optimal Bellman consistency on random convex combinations of multiple Q estimates (Agarwal, Schuurmans, and Norouzi 2020). Other strategies to control these biases include the use of clipped-double Q-learning (CDQ) (Fujimoto, Hoof, and Meger 2018), ensemble size adaptation (Wang, Lin, and Zhang 2021), and truncating sampled Q estimates using distributional networks (Kuznetsov et al. 2020, 2021; Dorka et al. 2023), amongst others. Unlike the previous work on the estimation bias of Q-value, ours focuses on the estimation bias of C-value, which is important in CRL.

Preliminaries

Constrained Markov Decision Processes

CRL can be modeled as a Constrained Markov Decision Process (CMDP). A typical CMDP consists of state $s \in \mathcal{S}$, action $a \in \mathcal{A}$, reward $r(s,a) \in \mathbb{R}$, cost $c(s,a) \in \mathbb{R}$, transition probabilities $p(s'|s,a)$, a discount factor $\gamma \in [0,1)$, and a given safety threshold b. The Q-function can be defined the expected discounted return as $Q(s,a) = \mathbb{E}[\sum_t \gamma^t r(s_t,a_t)|s_0=s,a_0=a]$. Similarly, the C-function is $C(s,a) = \mathbb{E}[\sum_t \gamma^t c(s_t,a_t)|s_0=s,a_0=a]$.

The goal of CRL is to maximize the expected return of the task ($\max_\pi J_R^\pi = \mathbb{E}_\pi[\sum_t \gamma^t r(s_t,a_t)]$) while ensuring that the expected cost satisfies the constraints ($J_C^\pi = \mathbb{E}_\pi[\sum_t \gamma^t c(s_t,a_t)] < b$). This constraint problem can usually be solved using the Lagrange-multiplier method. It can form the unconstrained dual problem as:

$$\min_{\beta\geq 0}\max_{\phi} L(\phi,\beta) = J_R^{\pi_\phi} - \beta\left(J_C^{\pi_\phi} - b\right), \tag{1}$$

Soft Actor-Critic

Soft Actor-Critic (SAC) is an off-policy RL algorithm with entropy regularization. It employs the clipped double-Q technique, selecting the minimum Q-value from two Q approximators as follows:

$$\begin{aligned} y_q(r,s',d) =& r + \gamma(1-d)\Bigg(\min_{j=1,2} Q_{targ,j}(s',a') \\ & -\alpha\log\pi_\phi(a'|s')\Bigg), \end{aligned} \tag{2}$$

where $a' \sim \pi_\phi(\cdot|s')$. The policy is trained to choose actions such that the Q-function is maximized with an additional entropy regularization as follows:

$$J_\pi(\phi,\psi) = \mathbb{E}_{s\sim D, a\sim\pi_\theta}[\lambda\log\pi_\phi(a|s) - Q_\psi(s,a)] \tag{3}$$

Truncated Quantile Critics

Truncated Quantile Critics (TQC) is a distributional off-policy RL algorithm based on SAC. TQC decomposes the expected return into atoms of distributional representation to achieve granularity in controlling the overestimation. Specially, it trains N_1 approximations $Z_{q,\psi_1},\cdots,Z_{q,\psi_{N_1}}$ of the policy conditioned return distribution. Each $Z_{q,\psi_{N_1}}$ maps (s,a) to a probability distribution $Z_{q,\psi_i} = \frac{1}{N_2}\sum_{j=1}^{N_2}\delta\left(\theta_{q,\psi_i}^j(s,a)\right)$ supported on atoms $\theta_{q,\psi_i}^1(s,a),\cdots,\theta_{q,\psi_i}^{N_2}(s,a)$. TQC pools atoms of all N_1 distributions into a set $\mathcal{Z}_q(s',a') = \{\theta_{q,\psi_i}^j(s,a)|i\in[1..N_1], j\in[1..N_2]\}$ and denote elements of $\mathcal{Z}_q(s',a')$ sorted in ascending order by $z_{(l)}(s',a')$, with $l\in[1..N_1N_2]$. To control the overestimation, TQC truncates the right tail of the return distribution approximation by dropping several of the topmost atoms, using the remaining kN_2 smallest elements in $\mathcal{Z}_q(s',a')$ to construct the target distribution $Y(s,a) = \frac{1}{kN_2}\sum_{l=1}^{kN_2}\delta\left(y_l(s,a)\right)$, where $y_l(s,a) := r(s,a) + \gamma\left[z_{(l)}(s',a') - \alpha\log\pi_\phi(a'\mid s')\right]$. The critic loss function is constructed to minimize the 1-Wasserstein between each $Z_{q,\psi_{N_1}}$ and the target distribution

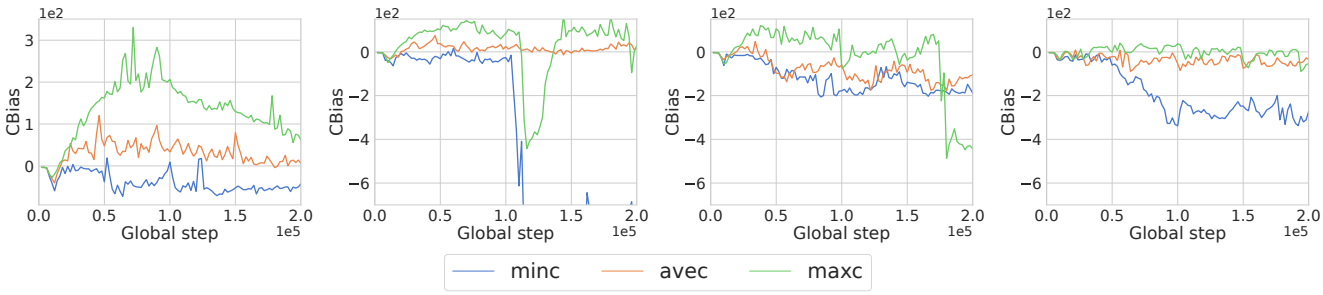

Figure 1: The deviation between the estimated C-value and the true C-value of the SAC algorithm with β under different parameters ($\beta \in \{0, 0.1, 0.5, 1\}$). minc: taking the minimum of two C_i estimates, maxc: taking the maximum of two C_i estimates, avec: taking the average of two C_i estimates.

$Y(s, a)$. The policy parameters ϕ are trained to optimize the nontruncated estimate of Q-value with an additional entropy regularization term like in SAC.

Estimation Bias in C-function

Overestimation of the Q-function has been widely acknowledged as a prevailing challenge within RL. Analogously, the estimation of the C-function grapples with comparable quandaries in CRL. RL primarily concentrates on amplifying expected rewards. This maximization operation and the expectation operation can lead to overestimating Q-values. In contrast, CRL introduces added complexity due to the constraint objective $\max_\pi \mathbb{E}[Q^\pi(s, a) - \beta C^\pi(s, a)]$. Incorporating the β term complicates matters, making it arduous to unequivocally discern whether C-values are subject to overestimation or underestimation. Furthermore, within the constrained optimization process, the magnitude of β remains in a state of flux.

To illustrate this inaccurate estimation, we conduct experiments in the Hopper velocity restriction task. Considering that β constantly changes during the constraint optimization process, we simulated four β settings ($\beta \in \{0, 0.1, 0.5, 1\}$) and calculated the estimated deviation of each C value as shown in Figure 1. In each setting, we illustrate the C-function bias of three schemes, *i.e.,* minc, maxc, avec, which have been utilized in addressing the overestimation of the Q-value. Experimental results reveal that during the constraint optimization process, the C-value could be either overestimated or underestimated as β changes, and a single estimation bias correction strategy cannot accommodate the entirety of the learning process.

Method

Drawing upon our above discourse concerning biases in the C-value, it becomes manifest that methodologies historically employed to rectify overestimation biases in Q-learning are ill-equipped to address similar estimation deviations in C-learning. Previous works (Lan et al. 2019; Chen et al. 2021) highlighted that using different numbers of Q-function approximations to calculate the target Q may lead to overestimation and underestimation of the Q-value estimation. Motivated by these, we propose an ensemble-based approach, leveraging a dynamic number of C-values to refine and correct the C-value estimation.

Adaptive Ensemble C-learning

We formally define an ensemble of N C-function approximations, and each represented as C_i, where $i \in \{1, 2, \cdots, N\}$. We utilize either the several smallest or largest C-values as an estimation of the C-learning target. More specifically, the C-learning target $C_{target}(s, a)$ is determined as $c + \gamma C^{\{N,K\}}(s, a')$, with the restriction of $|K| \leq N - 1$. In cases where $K > 0$, AEC applies the mean of the highest $N - K$ C-values as $C^{\{N,K\}}(s, a)$. Alternatively, when $K < 0$, the AEC resorts to the mean of the smallest $N + K$ cost values to compute $C^{\{N,K\}}(s, a)$.

Then, we clarify the relationship between every C-value approximation within the AEC framework and the true C-value. In alignment with the suppositions concerning the bias of Q-value estimates in the works (Thrun and Schwartz 1993; Lan et al. 2019; Chen et al. 2021), we posit that each $C_i(s, a)$ possesses a random approximation error, $e_i(s, a)$, such that $C_i(s, a) = C^*(s, a) + e_i(s, a)$. This error conforms to a uniform random distribution, $U(-\tau, \tau)$. We then define the estimation bias $Z^{\{N,K\}}$ for $C^{\{N,K\}}(s, a)$ and the true C-value $C^*(s, a)$ as follows:

$$\begin{aligned} Z^{\{N,K\}} &= \left(c + \gamma C^{\{N,K,\max\}}\left(s', a'\right)\right) \\ &\quad - \left(c + \gamma C^{\{*,\max\}}\left(s', a'\right)\right) \\ &= \gamma\left(C^{\{N,K,\max\}}\left(s', a'\right) - C^{\{*,\max\}}\left(s', a'\right)\right), \end{aligned} \tag{4}$$

$$\begin{aligned} C^{\{N,K,\max\}}\left(s', a'\right) = & C^{\{N,K\}}\left(s', \arg\max_{a'} Q\left(s', a'\right)\right. \\ & \left. - \beta C^{\{N,K\}}\left(s', a'\right)\right), \end{aligned} \tag{5}$$

$$\begin{aligned} C^{\{*,\max\}}\left(s', a'\right) = & C^{\{*\}}\left(s', \arg\max_{a'} Q\left(s', a'\right)\right. \\ & \left. - \beta C^{\{*\}}\left(s', a'\right)\right), \end{aligned} \tag{6}$$

Note that the above definition differs slightly from that of RL in the choice of the next action a' to be evaluated, since CRL

requires maximizing the constrained optimization objective $\max_\pi \mathbb{E}[Q^\pi(s,a) - \beta C^\pi(s,a)]$ rather than just the task objective $\max_\pi \mathbb{E}[Q^\pi(s,a)]$ to obtain the optimal action. Furthermore, the introduction of the Lagrange multiplier, β, imparts a level of complexity when attempting to quantify the magnitude of $C^{\{*,\max\}}(s',a')$. To comprehend the meaning of the above inequality, we can envisage a scenario wherein β approaches infinity, causing $C^{\{*,\max\}}(s',a')$ to converge towards $\min_{a'} C^*(s',a')$.

By utilizing $\mathbb{E}[Z^{\{N,K\}}]$, the expected discrepancy of $C^{\{N,K\}}(s,a)$ following each update from the true $C^*(s,a)$ can be determined. Consequently, when $\mathbb{E}[Z^{\{N,K\}}] > 0$, a positive expected deviation denotes a bias in the C-value estimation towards overestimation. Inversely, when $\mathbb{E}[Z^{\{N,K\}}] < 0$, a negative expected deviation suggests a bias in the C-value estimation towards underestimation. Unlike the bias of Q-value expectation in RL, the C-value expectation bias assumes a more critical role in CRL. This is due to the necessity to discern whether the current policy satisfies the constraint based on the C-value, where both overestimation and underestimation could precipitate significant adverse effects on policy updates.

Theorem 1. *Let M denote the number of actions applicable at state s', if all M actions share the same true Q-value and true C-value, i.e., $\exists\ q:\ \forall a':\ q = Q^*(s',a')$, $\exists\ c:\ \forall a':\ c = C^*(s',a')$. We then have:*

1. $\mathbb{E}\left[Z^{\{N,K\}}\right] < \mathbb{E}\left[Z^{\{N,K+1\}}\right]$ *for all* $N \geq 1$.
2. $\mathbb{E}\left[Z^{\{N,N-1\}}\right] \geq \gamma\tau\left[1 + 2t_{M,N}\right]$.
3. $\mathbb{E}\left[Z^{\{N,-N+1\}}\right] \leq \gamma\tau\left[1 - 2t_{M,N}\right]$.

$t_{M,N} = \frac{M(M-1)\cdots 1}{(M+\frac{1}{N})(M-1+\frac{1}{N})\cdots(1+\frac{1}{N})}$.

Proof. See Appendix A.1.

Theorem 1 clarifies that the overestimation and underestimation of the C-value can be controlled by adjusting the parameter K. Specifically, the second and third points within Theorem 1 present the lower bound of the expected bias $\mathbb{E}\left[Z^{\{N,K\}}\right]$ under the circumstance of $N-1$ and the upper bound in $-N+1$ situation as $\gamma\tau[1+2t_{M,N}]$ and $\gamma\tau[1-2t_{M,N}]$. These bounds are contingent on parameter N, suggesting that we can select a relatively large hyperparameter N, such that the lower bound $\gamma\tau[1+2t_{M,N}] > 0$, and the upper bound $\gamma\tau[1-2t_{M,N}] < 0$ (*e.g.*, as N approaches positive infinity, $t_{M,N}$ equals 1). Consequently, we can derive that $\mathbb{E}\left[Z^{\{N,N-1\}}\right] > 0$ and $\mathbb{E}\left[Z^{\{N,-N+1\}}\right] < 0$. With the first point from Theorem 1, we can infer that as K increases, the expected bias transitions from underestimation to overestimation. This theory provides us with an assurance for our subsequent learning of the parameter K, aiming to minimize the expected bias of the C-value.

Adaptive Adjustment of Parameter K via Error Feedback Mechanism. A primary step in AEC is to discern the discrepancy between the current C-value estimation and its true value. To this end, Monte Carlo simulations serve as an instrumental tool for estimating the C-value corresponding to each state-action pair under the current policy. A test trajectory of length H, represented by $\tau = (s_0, a_0, s_1, a_1, \cdots, s_H, a_H)$, is initiated from a randomly selected starting state following the current policy π. This process involves the computation of the discounted Monte Carlo cost-return $C^\pi(s,a)$ and the estimated C-function value $C_i(s,a), i = 1, \cdots, N$ for each encountered state-action pair (s,a). Subsequently, the difference between the C-function value $C_i(s,a)$ and the Monte Carlo cost-return $C^\pi(s,a)$ is calculated for all state-action pairs (s,a) within the trajectory, and then averaged as below:

$$\delta = \frac{1}{N}\sum_{i=1}^{N}\left(C_i(s,a) - C^\pi(s,a)\right), \quad (s,a) \in \tau. \tag{7}$$

This deviation δ facilitates the update of the parameter K. We learn from (Stooke, Achiam, and Abbeel 2020) and adopt a Proportional Integral Derivative (PID) controller to update as follows:

$$K \leftarrow \text{int}(K_p\delta + K_i\int_{i=1}^{k}\delta di + K_d\frac{\delta}{di}), \tag{8}$$

where K_p, K_i, K_d are the hyper-parameters, int indicates that the results need to be converted to integers.

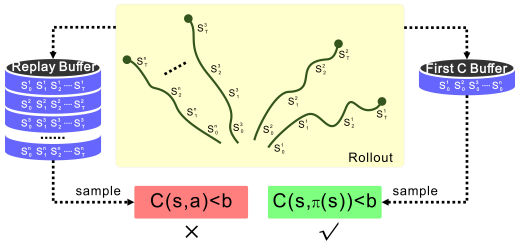

Figure 2: The contrast between the Replay Buffer and the First C Buffer during the process of updating the Lagrangian multiplier β. It is imperative to note that for illustrative simplicity, details pertaining to (a,r,d) within the Replay Buffer have been eschewed.

First C Buffer Technique

In this subsection, we delve into the process of updating the Lagrangian multiplier. The determination of β hinges upon the interplay of $(b - J_c^\pi)$, with b delineating the constraint threshold. If $b \geq J_c^\pi$, β is reduced. Conversely, if $b < J_c^\pi$, β is increased to augment the emphasis on safety. As discussed earlier, it is feasible to approximate J_c^π using the C-value. Preceding algorithms (Yang et al. 2021; Liu et al. 2022) typically employ $\mathbb{E}_{(s,a)\sim D}[C(s,a)]$ to approximate J_c^π. However, it should be emphasized that b describes the constraint threshold starting from the initial state. As the replay buffer D contains a multitude of non-initial states, utilizing $\mathbb{E}_{(s,a)\sim D}[C(s,a)]$ to approximate J_c^π may not be congruent with the given threshold b. Recognizing this difference, we introduce the First C Buffer (FCB) technique, capitalizing on the initial state s_0 from each episode during environment interactions as a representative approximation for J_c^π. Specifically, we maintain a buffer to store the initial state from each interaction trajectory. On each instance, a batch of state data is sampled from this buffer, and

$\mathbb{E}_{s\sim D_0}[C(s,\pi(s))]$ is employed to approximate J_c^π. Subsequently we calculate $\delta = b - \mathbb{E}_{s\sim D_0}[C(s,\pi(s))]$ and employ the PID control formula $\beta \leftarrow K_p\delta + K_i\int_{i=1}^{k}\delta di + K_d\frac{\delta}{di}$, to facilitate the update of β. Figure 2 explains the disparities between the FCB and the conventional replay buffer during the β update process.

Applay AEC and FCB to Off-Policy Algorithm

For the off-policy algorithms SAC and TQC, we commence by utilizing the Lagrangian relaxation technique, transforming the task objective into its corresponding constrained objective. Thereafter, we deploy the previously outlined FCB method in conjunction with PID control to precisely learn the parameter β. We then proceed with value function updates and policy refinement for each respective algorithm. The C-function loss is supplemented by the AEC method to rectify the C target, building upon the Q-function loss foundation inherent within each algorithm.

AECSAC. Mirroring the Q target computation strategy in SAC, as expressed in Eq. (1), AECSAC employs a dynamic estimation $C^{\{N,K\}}$ to calculate the C target in lieu of the clipped double-Q estimation:

$$y_c\left(c,s'\right) = c + \gamma\left(C_{targ}^{\{N,K\}}\left(s',a'\right)\right), \tag{9}$$

where $a' \sim \pi_\phi\left(\cdot|s'\right)$. Moreover, AECSAC introduces an additional constraint term through β to the actor loss in SAC:

$$J_\pi(\phi,\psi) = \mathbb{E}[\lambda\log\pi_\phi(a|s) - Q_\psi(s,a) + \beta C_\psi(s,a)]. \tag{10}$$

AECTQC. For the C-function, AECTQC also trains N_1 approximations $Z_{c,\psi_1},\cdots,Z_{c,\psi_{N_1}}$ of the cost distribution. Then we denote the elements in the set $\left\{\theta^j_{c,\psi_i}(s,a)|i\in[1\cdots N_1, j\in[1\cdots N_2]]\right\}$ in ascending order as $z_i(a,b)$ and in descending order as $z_{(i)}(a,b)$. Define $Z_i^K(a,b)$ with the following rule, when $K>0$, $Z_i^K(a,b) = z_{(i)}(a,b)$, and when $K<0$, $Z_i^K(a,b) = z_i(a,b)$. Then, the target distribution can be calculated as follows:

$$y_c\left(c,s'\right) = c + \gamma\left(\frac{1}{N_1N_2-|K|}\sum_{i=1}^{N_1N_2-|K|} z_i^K\left(s',a'\right)\right), \tag{11}$$

where $a' \sim \pi_\phi\left(\cdot|s'\right)$. The actor loss for AECTQC is:

$$J_\pi(\phi) = \mathbb{E}\left[\lambda\log\pi_\phi(a|s) - \frac{1}{N_1N_2}\sum_{i,j=1}^{N_1,N_2}\theta^j_{q,\psi_i}(s,a) + \frac{\beta}{N_1N_2}\sum_{i,j=1}^{N_1,N_2}\theta^j_{c,\psi_i}(s,a)\right]. \tag{12}$$

Experiments

In this section, we conduct experiments on nine tasks that derive from three different sets: 1) five MuJoCo tasks with speed limit (Ant-v3, Halfcheetah-v3, Humanoid-v3, Swimmer-v3, Walker2d-v3) (Todorov, Erez, and Tassa 2012; Zhang, Vuong, and Ross 2020), 2) Circle tasks (AntCircle, HumanoidCircle) (Achiam et al. 2017), and 3) Safe exploration tasks (CarGoal1, PointGoal1) in Safety-Gymnasium (Ji et al. 2023).

For task 1), the agent is directed to move in a straight line or within a two-dimensional plane. However, the robot's speed is restricted to maintain safety. For task 2), the agent earns rewards for circling within a set radius, with rewards tied to its speed and proximity to the circle's edge. Exiting the safety zone incurs costs. For task 3), the agent navigates in a 2D map to reach the goal position while trying to avoid hazardous areas. Rewards promote goal-directed movement, with a significant bonus upon reaching the goal. Additionally, the location of the robot and the goal will be reset randomly when the robot reaches a goal during training.

Comparison With Other Algorithms

We compare our approach against on-policy algorithms that include FOCOPS (Zhang, Vuong, and Ross 2020), CUP (Yang et al. 2022), CPO (Chow et al. 2017), PPOLag (Ray, Achiam, and Amodei 2019), and P3O (Zhang et al. 2022b). A summary of the performance of all the evaluated algorithms is presented in Table 1, 2. On evaluating all tasks, it becomes apparent that both AECSAC and AECTQC surpass other algorithms in most tasks with respect to reward, while still adhering to the cost constraint. In comparison to on-policy algorithms, AECSAC and AECTQC, as off-policy methods, demonstrate remarkable efficiency in interacting with the environment and better performance in the final outcome, particularly in complex constraint scenarios such as Humanoid-v3, Walker2d-v3, AntCircle, and HumanoidCircle. This efficiency and performance are important for successfully implementing RL in real-world scenarios.

We compare our approach against off-policy algorithms that include CVPO (Liu et al. 2022), SACLag (Ha et al. 2020), and CDMPO (Zhang et al. 2022a). Figure 3 illustrates the challenges that other off-policy CRL algorithms face in achieving an optimal balance between task objectives and constraint objectives. CVPO and SACLag do not address the issue of C-value estimation. As a result, due to inaccurate C-value estimation, it is difficult to ascertain whether the constraints are satisfied, which ultimately leads to subpar performance. However, within the CVPO experiments, in an attempt to circumvent the C-value estimation conundrum, they opt to modify the environment such that a fixed layout was maintained across episodes, eschewing randomized ones. Additionally, they impose uniform interaction lengths for each episode, refraining from issuing arbitrary termination signals. However, such a setup proves challenging to actualize in real-world scenarios. CDMPO endeavors to reduce the variance of estimates via value distribution methods (Bellemare, Dabney, and Munos 2017), but the persisting bias hinders them from achieving results that rival those of our algorithm.

Exploratory Experiments

In this part, we delve into the impact of our proposed components. We particularly address the ensuing considerations:

Algorithm		Ant-v3 (103.12)	HalfCheetah-v3 (151.99)	Humanoid-v3 (20.14)	Swimmer-v3 (24.52)	Walker2d-v3 (81.89)
FOCOPS	Reward	1469.299±59.839	1485.953±47.143	4577.634±108.064	30.165±2.451	2186.234±316.851
	Cost	101.417±0.5082	151.81±0.5086	19.42±0.3854	26.765±4.008	79.462±0.998
CUP	Reward	1288.84±200.584	1365.173±33.208	4567.943±240.874	36.669±5.651	760.098±446.716
	Cost	95.578±5.392	150.121±0.5088	18.351±0.6918	23.49±1.536	80.149±2.766
CPO	Reward	1301.991±193.17	19.992±315.826	443.012±19.048	30.802±4.512	638.964±337.717
	Cost	87.851±0.5038	78.144±4.556	18.707±0.7814	32.664±10.015	66.729±5.125
PPOLag	Reward	1270.81±72.884	-189.822±80.877	613.224±32.468	25.634±2.554	590.695±175.164
	Cost	91.194±16.343	42.09±3.206	25.952±3.304	29.499±7.525	76.327±5.635
P3O	Reward	1169.873±195.948	1111.309±372.506	4551.988±333.16	28.885±1.332	1063.666±414.351
	Cost	91.942±6.094	135.408±11.102	21.05±2.218	24.593±2.373	76.919±5.143
AECSAC	Reward	1645.73±125.594	**1561.42±20.705**	**5102.084±235.089**	45.787±3.478	**2954.377±54.398**
	Cost	93.017±0.1223	150.757±1.98	18.664±3.97	18.134±0.647	77.945±0.661
AECTQC	Reward	**1765.112±53.938**	1463.664±83.381	4959.277±225.09	**51.775±0.091**	2826.165±181.34
	Cost	103.172±0.309	141.01±11.347	20.162±1.854	24.132±0.3727	80.326±1.512

Table 1: Average results in five MuJoCo tasks for different algorithms over five seeds. Cost thresholds are in brackets.

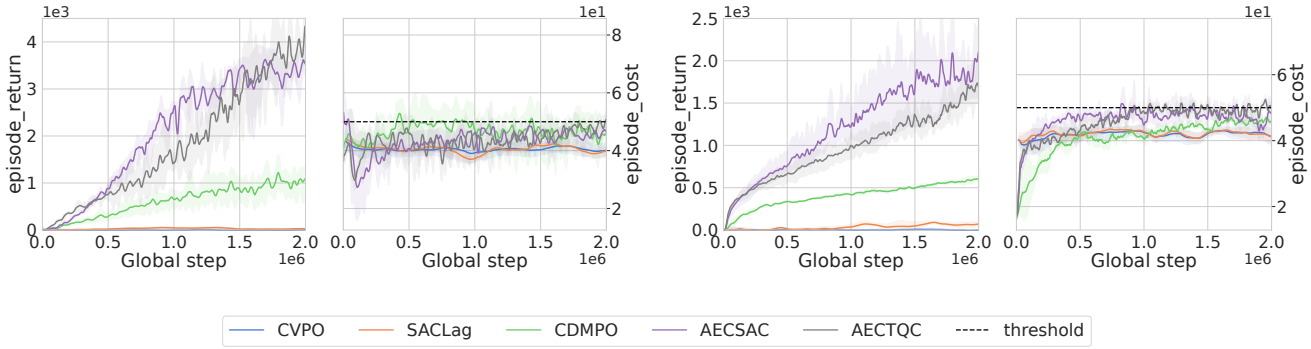

Figure 3: The performance of CVPO, SACLg, CDMPO, AECSAC, and AECTQC on two MuJoCo-circle tasks.

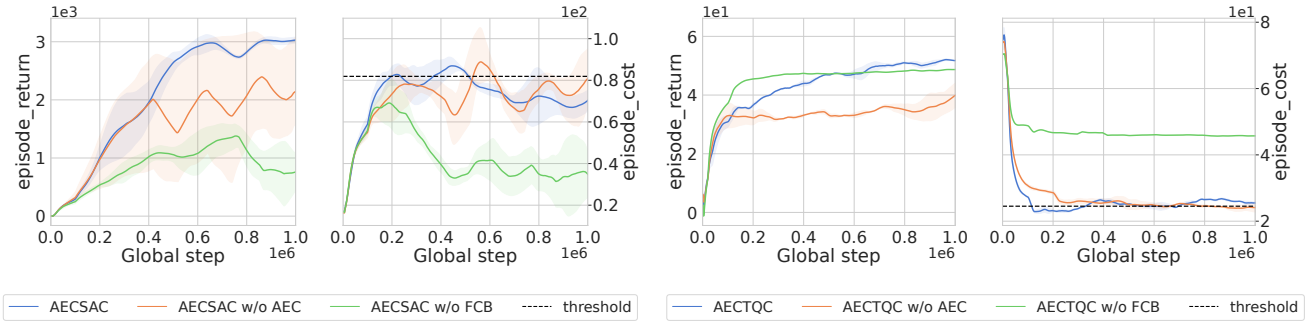

Figure 4: Ablation experiments for AECSAC and AECTQC.

(1) How instrumental is the AEC technology for AECSAC and AECTQC? Can it effectively rectify the biases encountered in C-value estimation throughout constrained optimization? To discern this, Figures 4 offer a comparative analysis of AECSAC against AECSAC w/o AEC and similarly, AECTQC against AECTQC w/o AEC on the Walker2d and Swimmer tasks. In the cases w/o AEC, an identical number of C-values was employed; however, the C-target computation relied on an aggregate average of all C-values. As the algorithm progresses through its learn-

Algorithm		Ant-Circle (50)	Humanoid-Circle (50)	PointGoal1 (3)	CarGoal1 (3)
FOCOPS	Reward	301.124±131.053	537.273±40.612	2.262±3.696	2.946±3.03
	Cost	44.669±4.313	43.476±1.651	2.799±1.848	2.302±0.414
CUP	Reward	763.043±276.272	592.554±30.721	0.959±3.635	1.509±1.753
	Cost	46.032±2.388	47.539±2.155	2.074±0.348	1.351±0.122
CPO	Reward	116.284±35.307	143.855±5.881	5.773±7.541	3.541±5.49
	Cost	42.446±3.814	28.995±3.013	2.772±0.942	2.403±0.129
PPOLag	Reward	122.858±29.244	246.837±32.25	0.3238±4.549	2.618±3.816
	Cost	24.116±6.419	23.923±3.159	3.092±1.818	1.482±0.223
P3O	Reward	121.206±92.478	591.393±14.52	5.266±1.504	1.684±0.115
	Cost	41.871±6.644	43.988±2.122	1.549±1.373	2.401±1.463
AECSAC	Reward	3636.119±268.687	**1935.683±89.38**	**22.243±2.38**	**7.429±0.38**
	Cost	48.272±2.271	48.4888±1.271	2.896±1.271	1.315±0.271
AECTQC	Reward	**4021.963±474.553**	1692.08±278.025	8.056±9.368	4.892±1.55
	Cost	49.714±2.134	50.075±1.898	3.353±1.674	0.424±0.838

Table 2: Average results in two circle tasks and two safe exploration tasks for different algorithms over five seeds.

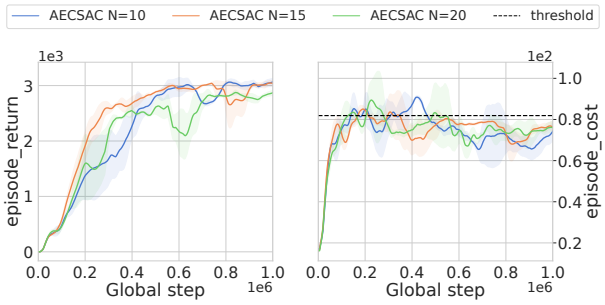

Figure 5: Parameter stability experiments for AECSAC.

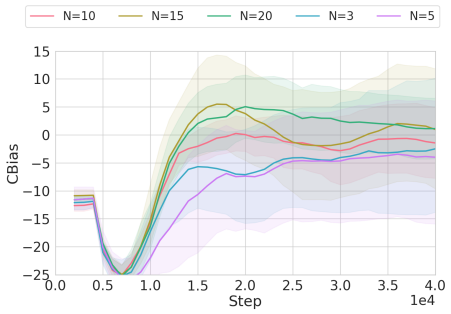

Figure 6: C-value bias in the Walker2d.

ing trajectory, the Lagrange multiplier may undergo fluctuations, causing shifts in the C-value deviations. Without AEC, the algorithm would grapple with accurately discerning if the prevailing policy adheres to the constraints. This can result in pronounced fluctuations in the learning curve, which in turn can detrimentally affect the overall performance.

(2) How integral is the FCB technology in ensuring compliance with the designated constraint threshold? In scenarios without the FCB, akin to traditional methodologies, we draw batches of state-action (s,a) pairs from the replay buffer during each iteration to compute $\mathbb{E}_{(s,a)\sim D}[C(s,a)]$, serving as an approximation for J_c^π. As illustrated in Figure 4, the incorporation of the FCB method to approximate J_c^π ensures that the algorithm's perception of constraints aligns meticulously with the predetermined threshold. In the absence of FCB, there's a risk that the algorithm could misconstrue the constraint threshold, leading to either an overly cautious approach, significantly undershooting the constraint threshold (Walker2d task), or an overly optimistic one, greatly surpassing the constraint threshold (Swimmer task).

(3) How does the quantity of C-value estimates (N for AECSAC) shape the efficacy of the algorithm? Theorem 1 posits that opting for a relatively large hyperparameter N can ensure that K modulates the C-value from underestimation to overestimation. However, our empirical investigations revealed a pronounced robustness with respect to N. As illustrated in Figure 5 and Figure 6, variations in the value of N (e.g., 10, 15, or 20) do not induce significant perturbations in the performance of AECSAC.

Conclusion

In this study, we propose the Adaptive Ensemble C-learning (AEC) technique to mitigate the issue of imprecise C-value estimation. Leveraging the power of state-of-the-art off-policy algorithms, we integrate AEC to enhance the task maximization under constraints. The proposed algorithms were tested on nine robotic constraint optimization tasks, where they showcased superior performance and improved efficiency in terms of environmental interaction. The research contributes significantly to the advancement of CRL, providing a viable solution for applying RL in real-world agents where safety constraints are paramount.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62206013. Thanks to Chuanhao Zhou for drawing pictures of some experiments in this paper.

References

Achiam, J.; Held, D.; Tamar, A.; and Abbeel, P. 2017. Constrained policy optimization. In *International Conference on Machine Learning*, 22–31. PMLR.

Agarwal, R.; Schuurmans, D.; and Norouzi, M. 2020. An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning*, 104–114. PMLR.

Anschel, O.; Baram, N.; and Shimkin, N. 2017. Averaged-dqn: Variance reduction and stabilization for deep reinforcement learning. In *International conference on machine learning*, 176–185. PMLR.

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

Chen, X.; Wang, C.; Zhou, Z.; and Ross, K. W. 2021. Randomized Ensembled Double Q-Learning: Learning Fast Without a Model. In *International Conference on Learning Representations*.

Chow, Y.; Ghavamzadeh, M.; Janson, L.; and Pavone, M. 2017. Risk-constrained reinforcement learning with percentile risk criteria. *The Journal of Machine Learning Research*, 18(1): 6070–6120.

Dai, J.; Ji, J.; Yang, L.; Zheng, Q.; and Pan, G. 2023. Augmented Proximal Policy Optimization for Safe Reinforcement Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 7288–7295.

Dorka, N.; Welschehold, T.; Bödecker, J.; and Burgard, W. 2023. Adaptively Calibrated Critic Estimates for Deep Reinforcement Learning. *IEEE Robotics and Automation Letters*, 8(2): 624–631.

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

Ha, S.; Xu, P.; Tan, Z.; Levine, S.; and Tan, J. 2020. Learning to Walk in the Real World with Minimal Human Effort. In *Conference on Robot Learning*, 1110–1120. PMLR.

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

Hasselt, H. 2010. Double Q-learning. *Advances in neural information processing systems*, 23.

He, T.; Zhao, W.; and Liu, C. 2023. Autocost: Evolving intrinsic cost for zero-violation reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Ji, J.; Zhang, B.; Pan, X.; Zhou, J.; Dai, J.; and Yang, Y. 2023. Safety-Gymnasium. *GitHub repository*.

Kim, D.; Lee, K.; and Oh, S. 2023. Efficient Trust Region-Based Safe Reinforcement Learning with Low-Bias Distributional Actor-Critic. *arXiv preprint arXiv:2301.10923*.

Kim, D.; and Oh, S. 2022a. Efficient Off-Policy Safe Reinforcement Learning Using Trust Region Conditional Value At Risk. *IEEE Robotics and Automation Letters*, 7(3): 7644–7651.

Kim, D.; and Oh, S. 2022b. TRC: Trust region conditional value at risk for safe reinforcement learning. *IEEE Robotics and Automation Letters*, 7(2): 2621–2628.

Kuznetsov, A.; Grishin, A.; Tsypin, A.; Ashukha, A.; and Vetrov, D. P. 2021. Automating Control of Overestimation Bias for Continuous Reinforcement Learning. *ArXiv abs/2110.13523*.

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. PMLR.

Lan, Q.; Pan, Y.; Fyshe, A.; and White, M. 2019. Maxmin Q-learning: Controlling the Estimation Bias of Q-learning. In *International Conference on Learning Representations*.

Liu, Y.; Ding, J.; and Liu, X. 2020. IPO: Interior-point policy optimization under constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 4940–4947.

Liu, Z.; Cen, Z.; Isenbaev, V.; Liu, W.; Wu, S.; Li, B.; and Zhao, D. 2022. Constrained variational policy optimization for safe reinforcement learning. In *International Conference on Machine Learning*, 13644–13668. PMLR.

Ma, H.; Guan, Y.; Li, S. E.; Zhang, X.; Zheng, S.; and Chen, J. 2021. Feasible Actor-Critic: Constrained Reinforcement Learning for Ensuring Statewise Safety. *arXiv preprint arXiv:2105.10682*.

Peng, Z.; Li, Q.; Liu, C.; and Zhou, B. 2022. Safe driving via expert guided policy optimization. In *Conference on Robot Learning*, 1554–1563. PMLR.

Ray, A.; Achiam, J.; and Amodei, D. 2019. Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1910.01708*, 7: 1.

Stooke, A.; Achiam, J.; and Abbeel, P. 2020. Responsive safety in reinforcement learning by pid lagrangian methods. In *International Conference on Machine Learning*, 9133–9143. PMLR.

Thrun, S.; and Schwartz, A. 1993. Issues in using function approximation for reinforcement learning. In *Proceedings of the Fourth Connectionist Models Summer School*, volume 255, 263. Hillsdale, NJ.

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

Wang, H.; Lin, S.; and Zhang, J. 2021. Adaptive Ensemble Q-learning: Minimizing Estimation Bias via Error Feedback. *Advances in Neural Information Processing Systems*, 34: 24778–24790.

Wang, W.; Yu, N.; Gao, Y.; and Shi, J. 2019. Safe off-policy deep reinforcement learning algorithm for volt-var control in power distribution systems. *IEEE Transactions on Smart Grid*, 11(4): 3008–3018.

Yang, L.; Ji, J.; Dai, J.; Zhang, L.; Zhou, B.; Li, P.; Yang, Y.; and Pan, G. 2022. Constrained update projection approach to safe policy optimization. *Advances in Neural Information Processing Systems*, 35: 9111–9124.

Yang, Q.; Simão, T. D.; Tindemans, S. H.; and Spaan, M. T. 2021. WCSAC: Worst-case soft actor critic for safety-constrained reinforcement learning. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence. AAAI Press, online*.

Yang, T.-Y.; Rosca, J.; Narasimhan, K.; and Ramadge, P. J. 2020. Projection-Based Constrained Policy Optimization. In *International Conference on Learning Representations*.

Yu, H.; Xu, W.; and Zhang, H. 2022. Towards Safe Reinforcement Learning with a Safety Editor Policy. In *NeurIPS*.

Zhang, H.; Lin, Y.; Han, S.; Wang, S.; and Lv, K. 2022a. Conservative distributional reinforcement learning with safety constraints. *arXiv preprint arXiv:2201.07286*.

Zhang, L.; Shen, L.; Yang, L.; Chen, S.; Wang, X.; Yuan, B.; and Tao, D. 2022b. Penalized Proximal Policy Optimization for Safe Reinforcement Learning. In *International Joint Conference on Artificial Intelligence*, 3744–3750.

Zhang, Y.; Vuong, Q.; and Ross, K. 2020. First order constrained optimization in policy space. *Advances in Neural Information Processing Systems*, 33: 15338–15349.