Appendix A. Appendix

In this appendix we provide missing proofs and implementation details. Specifically, we present Theorem 1 for CAC convergenc in Section A.1, Proposition 4 for calculating CAC actor gradient expression in Section A.2 and implementation details in Section A.3.

A.1. Proof for the CAC convergence

We prove the convergence of CAC using policy iteration style argument. Similar proofs have also been used in (Haarnoja et al., 2018, Theorem 4). The following lemmas establish the convergence of the policy evaluation and policy improvement of CAC, respectively.

Lemma 5 (CAC Policy Evaluation) *Given the current policy π and a baseline policy $\tilde{\pi}$, the policy evaluation step of CAC is formulated as:*

$$Q_{\pi_{k+1}} \leftarrow \mathcal{R}(s,a) + \gamma \left(\mathbf{P}\mathbb{E}^{\pi}[Q_{\pi_k}(s,a)] + \mathcal{I}^{\pi}_{\tilde{\pi}}(s)\right), \tag{14}$$

Consider an initial Q value $Q_0 : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ with $|\mathcal{A}| < \infty$. With the repeated application of Eq. (14), the sequence Q_k converges to the following entropy-regularized Q-value $Q^{\pi}_{\tilde{\pi}}$ as $k \to \infty$.

$$Q^{\pi}_{\tilde{\pi}}(s,a) := \mathbb{E}^{d^{\pi}}\left[\sum_{t=0}^{\infty} \gamma^t \big(r_t + \mathcal{I}^{\pi}_{\tilde{\pi}}\left(s_{t+1}\right)\big) \big| s_0 = s, a_0 = a\right]. \tag{15}$$

Proof Define the entropy augmented reward as $\mathcal{R}^{\pi}_{\tilde{\pi}}(s,a) \triangleq \mathcal{R}(s,a) + \mathcal{I}^{\pi}_{\tilde{\pi}}(s)$ and rewrite the update rule as:

$$Q_{\pi_{k+1}} \leftarrow \mathcal{R}^{\pi}_{\tilde{\pi}}(s,a) + \gamma \mathbf{P}\mathbb{E}^{\pi}[Q_{\pi_k}(s,a)]. \tag{16}$$

With the assumption $|\mathcal{A}| < \infty$ for bounded reward, we can apply the standard convergence results for policy evaluation (Sutton and Barto, 2018). ■

Lemma 6 (CAC Policy Improvement) *Given the current policy π, a baseline policy $\tilde{\pi}$ and the updated policy π_{new}. CAC has the following policy update:*

$$\begin{aligned} &\pi_{new} = (1-\zeta)\pi + \zeta\hat{\pi}, \\ &\textit{where} \quad \hat{\pi}(a \mid s) = \frac{\tilde{\pi}^{\alpha}(a \mid s) \exp\left(\beta Q^{\pi}_{\tilde{\pi}}(s,a)\right)}{Z(s)}, \end{aligned} \tag{17}$$

with $\zeta \in [0,1]$. Then, $Q^{\pi_{new}}_{\tilde{\pi}}(s,a) \geq Q^{\pi}_{\tilde{\pi}}(s,a)$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$ with $|\mathcal{A}| < \infty$.

Proof Consider a function $f : \zeta \to \mathbb{R}$ with $\zeta \in [0,1]$:

$$f(\zeta) = \mathbb{E}^{a \sim \pi_{\text{new}}}\left[Q^{\pi}_{\tilde{\pi}}(s,a)\right] + \mathcal{I}^{\pi_{\text{new}}}_{\tilde{\pi}}\left(s\right). \tag{18}$$

From the definition of $\hat{\pi} = \arg\max_{\pi} \mathbb{E}^{a\sim\pi}\left[Q^{\pi}_{\tilde{\pi}}(s,a)\right] + \mathcal{I}^{\pi}_{\tilde{\pi}}(s)$, $f(\zeta)$ takes the maximum value when $\zeta = 1$. The first and the second derivative of $f(\zeta)$ w.r.t. ζ are:

$$\begin{aligned} f^{'}(\zeta) = \sum_{a} &\left(\hat{\pi}(a|s) - \pi(a|s)\right)\left(Q^{\pi}_{\tilde{\pi}}(s,a) + \tau \log \tilde{\pi}(a|s)\right. \\ &\left. -(\sigma+\tau)\log((1-\zeta)\pi(a|s) + \zeta\hat{\pi}(a|s)\right). \end{aligned} \tag{19}$$

$$f^{''}(\zeta) = -(\sigma+\tau)\sum_{a} \frac{\left(\hat{\pi}(a|s) - \pi(a|s)\right)^2}{(1-\zeta)\pi(a|s) + \zeta\hat{\pi}(a|s)} \leq 0. \tag{20}$$

Thus, the function f is concave in $\zeta \in [0,1]$. Since $f(\zeta)$ takes the maximum value with $\zeta = 1$, $f(\zeta)$ is monotonically increasing in ζ and $f(0) \leq f(\zeta)$.

Therefore, the following inequality about the entropy-regularized V-value $V^{\pi}_{\tilde{\pi}}(s)$ holds:

$$\begin{aligned} V^{\pi}_{\tilde{\pi}}(s) =& \mathbb{E}^{a\sim\pi}\left[Q^{\pi}_{\tilde{\pi}}(s,a)\right] + \mathcal{I}^{\pi}_{\tilde{\pi}}(s) \\ \leq& \mathbb{E}^{a\sim\pi_{\text{new}}}\left[Q^{\pi}_{\tilde{\pi}}(s,a)\right] + \mathcal{I}^{\pi_{\text{new}}}_{\tilde{\pi}}(s). \end{aligned} \tag{21}$$

By repeatedly applying Eq. (21), we obtain the following inequalities:

$$\begin{aligned} Q^{\pi}_{\tilde{\pi}}(s,a) &= \mathcal{R}(s,a) + \gamma\mathbf{P}\left[Q^{\pi}_{\tilde{\pi}}(s,a) + \mathcal{I}^{\pi}_{\tilde{\pi}}(s)\right] \\ &\leq \mathcal{R}(s,a) + \gamma\mathbf{P}\left[\mathbb{E}^{\pi_{\text{new}}}\left[Q^{\pi}_{\tilde{\pi}}(s,a)\right] + \mathcal{I}^{\pi_{\text{new}}}_{\tilde{\pi}}(s)\right] \\ &\vdots \\ &\leq Q^{\pi_{\text{new}}}_{\tilde{\pi}}(s,a). \end{aligned} \tag{22}$$

Convergence to $Q^{\pi_{\text{new}}}_{\tilde{\pi}}$ follows from Lemma 5. ■

Combining the policy evaluation and policy improvement, we are now ready to prove Theorem 1.

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 According to Lemma 5 and Lemma 6, the entropy-regularized Q-value at k-th update satisfies $Q^{\pi_k}_{\pi_{k-1}}(s,a) \geq Q^{\pi_{k-1}}_{\pi_{k-1}}(s,a)$. Given bounded reward, $Q^{\pi_k}_{\pi_k}$ is also bounded from above and the sequence converges to a unique π^*. Note that when reaching the optimum the KL regularization term becomes 0. Hence, using the same iterative argument as in the proof of Lemma 6, we get $Q^{\pi^*}_{\pi^*}(s,a) > Q^{\pi}_{\pi}(s,a)$ for all $(s,a) \in \mathcal{S}\times\mathcal{A}$ and any π. By Ziebart (2010); Haarnoja et al. (2018), the optimal policy is entropy-regularized and hence has the softmax form $\pi^*(a|s) = \frac{\exp\left(\frac{1}{\kappa}Q^*(s,a)\right)}{\int_{a\in\mathcal{A}}\exp\left(\frac{1}{\kappa}Q^*(s,a)\right)}$ (recall from Eq. (3) that κ is the weight coefficient of entropy). The convergence of general interpolated policy to the optimal policy follows the argument of Scherrer and Geist (2014). ■

A.2. Proof for the CAC gradient

In this subsection we derive the gradient expression for CAC. For the ease of reading we rephrase the proposition here:

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:*

$$\begin{aligned}
&\nabla_\phi \mathbb{E}^{s\sim\mathcal{B}}_{a\sim\pi_\phi}\left[D^\phi_{\bar{\phi}} - \frac{\beta}{1+\mathcal{X}} Q_\theta(s,a)\right], \\
&\text{where } D^\phi_{\bar{\phi}} = \log \pi_\phi\left(a \mid s\right) - \frac{\alpha+\mathcal{X}}{1+\mathcal{X}} \log \pi_{\bar{\phi}}(a;s) \\
&\mathcal{X} = \frac{1-\zeta}{\zeta}\cdot\frac{\pi_{\bar{\phi}}(a \mid s)}{\mathcal{G}Q_{\bar{\phi},\theta}(a \mid s)}.
\end{aligned} \tag{23}$$

Proof Using the reparameterization trick $a = f_\phi(\epsilon; s_t)$ with ϵ a noise vector (Haarnoja et al., 2018), the gradient of Eq. (8c) can be expressed as:

$$\begin{aligned}
\hat{\nabla}_\phi J_\pi(\phi) = &\nabla_\phi \log \pi_\phi\left(a \mid s\right) + \nabla_a \log \pi_\phi\left(a \mid s\right) \nabla_\phi f_\phi\left(\epsilon; s_t\right) \\
&- \nabla_a \log\left((1-\zeta)\pi_{\bar{\phi}}(a \mid s) + \zeta \mathcal{G}Q_{\bar{\phi},\theta}\left(a \mid s\right)\right) \nabla_\phi f_\phi\left(\epsilon; s_t\right).
\end{aligned} \tag{24}$$

We expand the term $\nabla_a \log\left((1-\zeta)\pi_{\bar{\phi}}(a \mid s) + \zeta \mathcal{G}Q_{\bar{\phi},\theta}\left(a \mid s\right)\right)$ by using that $\nabla_{x_i} \log\left(\sum_i \exp x_i\right) = \frac{\exp x_i}{\sum_j \exp x_j}$.

Let $\exp\left(C_1(a)\right) = (1-\zeta)\pi_{\bar{\phi}}(a \mid s)$ and $\exp\left(C_2(a)\right) = \zeta \mathcal{G}_{\pi_{\bar{\phi}}} Q_\theta\left(a \mid s\right)$. We have the following transformation:

$$\begin{aligned}
&\nabla_a \log\left((1-\zeta)\pi_{\bar{\phi}}(a \mid s) + \zeta \mathcal{G}Q_{\bar{\phi},\theta}\left(a \mid s\right)\right) \\
=&\nabla_a \log\left(\exp\left(C_1(a)\right) + \exp\left(C_2(a)\right)\right) \\
=&\frac{\left(D\frac{\partial}{\partial a}\pi_{\bar{\phi}}(a \mid s) + \alpha\frac{\partial}{\partial a}\pi_{\bar{\phi}}(a \mid s) + \beta\frac{\partial}{\partial a}Q_\theta(s,a)\right)}{1+D}, \\
&\text{where } D = \exp\left(C_1(a) - C_2(a)\right).
\end{aligned} \tag{25}$$

After replacing D with $\mathcal{X}$ and inserting Eq. (25) into Eq. (24), we obtain Eq. (10). ■

A.3. Implementation details

This section presents implementation details of CAC with deep networks. Pseudo-code is provided in Algorithm 1.

On-policy replay buffer To make the algorithm off-policy, we approximate the on-policy samples with on-policy replay buffer $\mathcal{B}_K$ which stores K recent samples where K is smaller than the size of the main replay buffer $\mathcal{B}$.

Advantage estimation While it is possible to simply use the entropy-regularized advantage function $A_{\mathcal{I}}(s,a)=Q_{\mathcal{I}}(s,a)-V_{\mathcal{I}}(s)$ for computing ζ, we are interested in studying the guidance of ζ when no entropy is involved since it might provide a more informative gradient improving direction. This corresponds to the case of (Kakade and Langford, 2002; Pirotta et al., 2013). To this end, we train another Q-network Q_ω by solving:

$$\begin{aligned} \omega &\leftarrow \arg\min \mathbb{E}^{\mathcal{B}}\left[\left(Q_\omega\left(s,a\right)-y\right)^2\right], \\ \text{where} \quad y &= r+\gamma\left(\mathbb{E}^{a\sim\pi_\phi(\cdot|s')}\left[Q_{\bar{\omega}}(s',a)\right]\right), \end{aligned} \tag{26}$$

where $\bar{\omega}$ is the target network. Then we approximate the advantage as $A^\pi(s,a) = Q_\omega(s,a) - \mathbb{E}^{\pi_\phi}[Q_\omega(s,a)]$. While the advantage estimation is expected to be further improved with the recent generalized advantage estimation, we found that the above simple implementation is sufficient to stabilize the learning.

Target smoothing For the target Q-networks $Q_{\bar{\theta}}$ and $Q_{\bar{\omega}}$, we update the parameters using the moving average (Haarnoja et al., 2018):

$$\begin{aligned} \bar{\theta} &\leftarrow \nu_{\bar{\theta}}\,\theta + (1-\nu_{\bar{\theta}})\bar{\theta}, \\ \bar{\omega} &\leftarrow \nu_{\bar{\omega}}\,\omega + (1-\nu_{\bar{\omega}})\bar{\omega}, \end{aligned} \tag{27}$$

where $\nu_{\bar{\theta}}$ and $\nu_{\bar{\omega}}$ are the target smoothing coefficients. In the mixing step, we use the previous policy $\pi_{\bar{\phi}}$ rather than the current policy π_ϕ to stabilize the training:

$$\mathbb{E}^{\mathcal{B}}\left[D_{KL}\left(\pi_\phi \| (1-\zeta)\pi_{\bar{\phi}} + \zeta\hat{\pi}\right)\right].$$

Thus, the target policy $\pi_{\bar{\phi}}$ corresponds to the monotonically improved policy in the CPI algorithm that is not updated when performance oscillation happens. To reflect this fact, we update the weight of the target policy network as:

$$\bar{\phi} \leftarrow \zeta\nu_{\bar{\phi}}\,\theta + (1-\zeta\nu_{\bar{\phi}})\bar{\phi}, \tag{28}$$

where $\nu_{\bar{\phi}}$ is the target smoothing coefficient.

Normalization factor estimation Since CAC algorithm requires the density of the reference policy $\hat{\pi}(a \mid s) = \pi_{\bar{\phi}}^{\alpha}(a \mid s)\exp\left(\beta Q_\theta(s,a)\right)(Z(s))^{-1}$, we need to estimate the normalization factor $Z(s)$.

A simple approach to estimate $Z(s)$ is by Monte-Carlo sampling with some distribution q that is easier to sample from:

$$Z(s) = \mathbb{E}^q\left[\frac{\pi_{\bar{\phi}}(a \mid s)^\alpha \exp\left(\beta Q_\theta(s,a)\right)}{q(a \mid s)}\right]. \tag{29}$$

The closer $q(\cdot \mid s)$ and the reference policy $\hat{\pi}(\cdot \mid s)$ are, the better the accuracy of the $Z(s)$ approximation.

Theorem 2 indicates that by choosing the current policy π as the proposal distribution, we can control the closeness of the two distributions and the accuracy of the MC approximation via changing the entropy regularization weighting coefficients. We empirically study the effectiveness of entropy regularization against the closeness and the accuracy when $\zeta \leq 1$

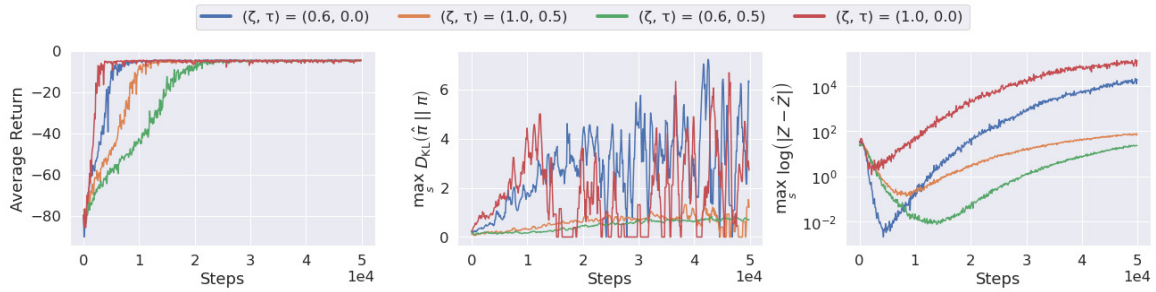

Figure 3: **Left:** The learning curves of CAC with different parameters on pendulum. **Middle:** The maximum KL divergence over all the states. **Right:** The maximum error of log-scaled $Z(s)$ approximation over all the states.

Algorithm 1 Cautious Actor-Critic

Initialize parameter vectors θ, ϕ, $\bar{\theta}$, $\bar{\phi}$, ω, $\bar{\omega}$
Initialize variable $\tilde{\mathbb{A}}$ and $\tilde{\mathbb{A}}^{\text{MaxDiff}}$
for each iteration **do**
  Collect transitions by π_θ and add them to $\mathcal{B}$ and $\mathcal{B}_K$
  for each gradient step **do**
    Update θ with one step of SGD using Eq. (8b)
    Update ω with one step of SGD using Eq. (26)
    Update $\tilde{\mathbb{A}}$ and $\tilde{\mathbb{A}}^{\text{MaxDiff}}$ using Eq. (12)
    Update ϕ with one step of SGD using Eq. (8c)
    Update ζ using Eq. (11)
    $\bar{\theta} \leftarrow \nu_{\bar{\theta}}\theta + (1-\nu_{\bar{\theta}})\bar{\theta}$
    $\bar{\omega} \leftarrow \nu_{\bar{\omega}}\omega + (1-\nu_{\bar{\theta}})\bar{\omega}$
    $\bar{\phi} \leftarrow \zeta\nu_{\bar{\phi}}\phi + (1-\zeta\nu_{\bar{\phi}})\bar{\phi}$
  end for
end for

We use the pendulum environment from Fu et al. (2019) where the dynamics are discretized so that we can compute the oracle values such as the KL divergence between the current and the reference policy. The hyperparameters used in the experiment is listed in Section A.4. Figure 3 shows how the learning behavior of CAC changes when the interpolation coefficient ζ and KL regularization weight τ vary: When KL regularization is present, the approximation quality of $Z(s)$ is improved significantly.

A.4. Hyperparameters

This section lists the hyperparameters used in the comparative evaluation Section 5.1.

Table 2: Hyperparameters of off-policy algorithms in mujoco tasks

Parameter	Value
Shared	
optimizer	Adam
learning rate	10^{-3}
discount factor (γ)	0.99
replay buffer size ($\mathcal{B}$)	10^6
number of hidden layers	2
number of hidden units per layer	256
number of samples per minibatch	100
activations	ReLU
TD3	
Stddev for Gaussian noise	0.1
Stddev for target smoothing noise	0.2
policy delay	2
SAC	
entropy coefficient (κ)	0.2
$\bar{\theta}$ smoothing coefficient	0.995
CAC	
entropy coefficient (κ)	0.2
KL coefficient (τ)	0.1
$\bar{\theta}$ smoothing coefficient ($\nu_{\bar{\theta}}$)	0.995
$\bar{\omega}$ smoothing coefficient ($\nu_{\bar{\omega}}$)	0.995
$\bar{\phi}$ smoothing coefficient ($\nu_{\bar{\phi}}$)	0.9999
ν_A	0.01
$\nu_{A^{\mathrm{MaxDiff}}}$	0.001
size of $\mathcal{B}_K$	1000
`if-else` update	$c = M$

Table 3: Hyperparameters of PPO in mujoco tasks

Parameter	Value
PPO	
optimizer	Adam
value function learning rate	10^{-3}
policy learning rate	3×10^{-4}
discount factor (γ)	0.99
number of hidden layers	2
number of hidden units per layer	256
number of samples per minibatch	100
activations	ReLU
Number of samples per update	80
Policy objective clipping coefficient	0.2

Table 4: Hyperparameters of CAC in pendulum task

Parameter	Value
CAC	
optimizer	Adam
learning rate	10^{-3}
discount factor (γ)	0.99
replay buffer size ($\mathcal{B}$)	10^6
number of hidden layers	2
number of hidden units per layer	256
number of samples per minibatch	32
activations	ReLU
entropy coefficient (κ)	0.2
$\bar{\theta}$ smoothing coefficient	0.995
$\bar{\phi}$ smoothing coefficient	0.995
size of $\mathcal{B}_K$	1000
`if-else` update	$c = M$