DIVERSITY ACTOR-CRITIC: SAMPLE-AWARE ENTROPY REGULARIZATION FOR SAMPLE-EFFICIENT EXPLORATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Policy entropy regularization is commonly used for better exploration in deep reinforcement learning (RL). However, policy entropy regularization is sample-inefficient in off-policy learning since it does not take the distribution of previous samples stored in the replay buffer into account. In order to take advantage of the previous sample distribution from the replay buffer for sample-efficient exploration, we propose sample-aware entropy regularization which maximizes the entropy of weighted sum of the policy action distribution and the sample action distribution from the replay buffer. We formulate the problem of sample-aware entropy regularized policy iteration, prove its convergence, and provide a practical algorithm named diversity actor-critic (DAC) which is a generalization of soft actor-critic (SAC). Numerical results show that DAC significantly outperforms SAC baselines and other state-of-the-art RL algorithms.

1 INTRODUCTION

Reinforcement learning (RL) aims to maximize the discounted sum of rewards under Markov decision process (MDP) environments (Sutton & Barto, 1998). When the given task is complex, i.e. the environment has high action-dimensions or sparse rewards, it is important to well explore state-action pairs for high performance (Agre & Rosenschein, 1996). For better exploration, recent RL considers various methods: maximizing the policy entropy to take actions more uniformly (Ziebart et al., 2008; Fox et al., 2015; Haarnoja et al., 2017), maximizing diversity gain that yields intrinsic rewards to explore rare states by counting the number of visiting states (Strehl & Littman, 2008; Lopes et al., 2012), maximizing information gain (Houthooft et al., 2016; Hong et al., 2018), maximizing model prediction error (Achiam & Sastry, 2017; Pathak et al., 2017), and so on. In particular, based on policy iteration for soft Q-learning, (Haarnoja et al., 2018a) considered an off-policy actor-critic framework for maximum entropy RL and proposed the soft actor-critic (SAC) algorithm, which has competitive performance for challenging continuous control tasks.

In this paper, we reconsider the problem of policy entropy regularization in off-policy learning and propose a generalized approach to policy entropy regularization. In off-policy learning, we store and reuse old samples to update the current policy (Mnih et al., 2015), and it is preferable that the old sample distribution in the replay buffer is uniformly distributed for better performance. However, the simple policy entropy regularization tries to maximize the entropy of the current policy irrespective of the distribution of previous samples. Since the uniform distribution has maximum entropy, the current policy will choose previously less-sampled actions and more-sampled actions with the same probability and hence the simple policy entropy regularization is sample-unaware and sample-inefficient. In order to overcome this drawback, we propose sample-aware entropy regularization, which tries to maximize the weighted sum of the current policy action distribution and the sample action distribution from the replay buffer. We will show that the proposed sample-aware entropy regularization reduces to maximizing the sum of the policy entropy and the α-skewed Jensen-Shannon divergence (Nielsen, 2019) between the policy distribution and the buffer sample action distribution, and hence it generalizes SAC. We will also show that properly exploiting the sample action distribution in addition to the policy entropy over learning phases will yield far better performance.

2 RELATED WORKS

Entropy regularization: Entropy regularization maximizes the sum of the expected return and the policy action entropy. It encourages the agent to visit the action space uniformly for each given state, and the regularized policy is robust to modeling error (Ziebart, 2010). Entropy regularization is considered in various domains for better optimization: inverse reinforcement learning (Ziebart et al., 2008), stochastic optimal control problems (Todorov, 2008; Toussaint, 2009; Rawlik et al., 2013), and off-policy reinforcement learning (Fox et al., 2015; Haarnoja et al., 2017; 2018a). An adaptation method for the entropy regularization coefficient is proposed by (Haarnoja et al., 2018b). (Lee et al., 2019) shows that Tsallis entropy regularization that generalizes usual Shannon-entropy regularization is helpful. (Nachum et al., 2017a) shows that there exists a connection between value-based and policy-based RL under entropy regularization. (O'Donoghue et al., 2016) proposed an algorithm combining them, and it is proven that they are equivalent (Schulman et al., 2017a).

Diversity gain: Diversity gain is used to provide a guidance for exploration to the agent. To achieve diversity gain, many intrinsically-motivated approaches and intrinsic reward design methods have been considered, e.g., intrinsic reward based on curiosity (Chentanez et al., 2005; Baldassarre & Mirolli, 2013), model prediction error (Achiam & Sastry, 2017; Pathak et al., 2017; Burda et al., 2018), divergence/information gain (Houthooft et al., 2016; Hong et al., 2018), counting (Strehl & Littman, 2008; Lopes et al., 2012; Tang et al., 2017; Martin et al., 2017), and unification of them (Bellemare et al., 2016). For self-imitation learning, (Gangwani et al., 2018) considered the Stein-variational gradient decent with the Jensen-Shannon kennel.

Off-policy learning: Off-policy learning reuses previous samples for current policy update (Sutton & Barto, 1998; Degris et al., 2012; Munos et al., 2016), so it is sample-efficient as compared to on-policy learning. In order to reuse old samples, a replay buffer that stores trajectories generated by previous policies is used for Q-learning (Mnih et al., 2015; Lillicrap et al., 2015; Fujimoto et al., 2018; Haarnoja et al., 2018a). To enhance both stability and sample-efficiency, several methods are considered, e.g., combining on-policy and off-policy (Wang et al., 2016; Gu et al., 2016; 2017), and generalization from on-policy to off-policy (Nachum et al., 2017b; Han & Sung, 2019).

3 BACKGROUND

In this section, we briefly introduce the basic setup and recapitulate the soft actor-critic (SAC) algorithm.

3.1 SETUP

We assume a basic RL setup composed of an environment and an agent. The environment follows a Markov decision process $(\mathcal{S}, \mathcal{A}, P, \gamma, r)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the continuous action space, P is the transition probability, γ is the discount factor, and $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function. The agent has a policy $\pi : \mathcal{S} \to \mathcal{A}$ which selects an action a_t for a given state s_t at each time step t, and the agent interacts with the environment and receives reward $r_t := r(s_t, a_t)$ from the environment. Standard RL aims to maximize the discounted return $\mathbb{E}_{\tau_0 \sim \pi}[\sum_{t=0}^{T-1} \gamma^t r_t]$ for each episode of length T, where $\tau_t = (s_t, a_t, \cdots, s_{T-1}, a_{T-1})$ is an episode trajectory.

3.2 SOFT ACTOR-CRITIC

Soft actor-critic (SAC) (Haarnoja et al., 2018a) includes a policy entropy regularization term in the objective function for better exploration by visiting the action space uniformly for each given state. The entropy-augmented policy objective function of SAC is given by

$$J(\pi) = \mathbb{E}_{\tau_0 \sim \pi} \left[\sum_{t=0}^{T-1} \gamma^t (r_t + \beta \mathcal{H}(\pi(\cdot|s_t))) \right], \tag{1}$$

where $\mathcal{H}$ is the entropy function and $\beta \in (0, \infty)$ is the entropy weighting factor. SAC generalizes off-policy actor-critic based on soft policy iteration to find the policy that maximizes (1). Soft policy iteration alternates soft policy evaluation to estimate the true soft Q-function and soft policy improvement to find the optimal policy that maximizes (1) (Haarnoja et al., 2018a). In addition, soft

policy iteration theoretically guarantees convergence to optimal policy π^*. To reduce overestimation bias, SAC uses double Q-function as suggested in (Fujimoto et al., 2018).

4 THE DIVERSITY ACTOR-CRITIC ALGORITHM

In off-policy learning, previous samples are stored in the replay buffer and reused to learn the policy. The policy entropy regularization in Section 3.2 enhances exploration to yield better performance, but it does not take advantage of the previous sample action distribution obtainable from the replay buffer. Even if there already exist many similar actions in the replay buffer, the policy with the simple entropy regularization will choose these actions with the same probability as less-sampled actions from the perspective of exploration, as seen by the term $\mathcal{H}(\pi(\cdot|s_t))$ in (1) because the uniform distribution has maximum entropy. Thus, it is sample-inefficient, so we aim to consider and exploit the sample action distribution in the replay buffer for better sample efficiency.

4.1 SAMPLE-AWARE ENTROPY REGULARIZATION

In order to overcome the limitation of the simple entropy regularization, we consider the entropy of the mixture of the policy distribution and the sample action distribution in the replay buffer. The rationale behind this is that it is preferable to have as diverse actions stored in the replay buffer as possible for better Q estimation in off-policy learning.

First, we define the current sample action distribution based on old samples in the replay buffer as

$$q(\cdot|s_t) := \frac{\sum_{a\in\mathcal{D}} N(s_t,a)\delta_a(\cdot)}{\sum_{a'\in\mathcal{D}} N(s_t,a')}, \tag{2}$$

where $\mathcal{D}$ is the replay buffer that stores previous samples (s_t, a_t, r_t, s_{t+1}), $\delta_a(\cdot)$ is the Dirac measure at $a \in \mathcal{A}$, and $N(s_t, a_t)$ is the number of state-action pair (s_t, a_t) in $\mathcal{D}$. Note that if we draw current samples from policy π and store them in the replay buffer, the updated sample action distribution will be a mixture of π and q. Thus, we define the target sample action distribution as the weighted sum of the policy distribution π and the current sample action distribution q, which is expressed as $q^{\pi,\alpha}_{target} := \alpha\pi + (1-\alpha)q$, where $\alpha \in [0, 1]$ is the weighting factor.

Our approach is to maximize the return while maximizing the target sample action distribution. Under this approach, previously many times sampled actions will be given low probabilities and previously less taken actions will be given high probabilities by the current policy π so that the entropy of the target or mixture distribution $\alpha\pi+(1-\alpha)q$ is maximized. Hence, we set the objective function for the proposed sample-aware entropy regularization as

$$J(\pi) = \mathbb{E}_{\tau_0\sim\pi}\left[\sum_{t=0}^{T-1}\gamma^t(r_t + \beta\mathcal{H}(q^{\pi,\alpha}_{target}))\right]. \tag{3}$$

Here, the sample-aware entropy $\mathcal{H}(q^{\pi,\alpha}_{target})$ for given s_t can be decomposed as

$$\mathcal{H}(q^{\pi,\alpha}_{target}) = -\int_{a\in\mathcal{A}}(\alpha\pi + (1-\alpha)q)\log(\alpha\pi + (1-\alpha)q) = \alpha\mathcal{H}(\pi) + D^{\alpha}_{JS}(\pi||q) + (1-\alpha)\mathcal{H}(q), \tag{4}$$

where $D^{\alpha}_{JS}(\pi||q) := \alpha D_{KL}(\pi||q^{\pi,\alpha}_{target}) + (1-\alpha)D_{KL}(q||q^{\pi,\alpha}_{target})$ is the α skew-symmetric Jensen-Shannon (JS) divergence (Nielsen, 2019). Note that D^{α}_{JS} reduces to the standard JS divergence for $\alpha = \frac{1}{2}$ and to zero for $\alpha = 0$ or 1. Hence, for $\alpha = 1$, (4) reduces to the simple entropy, but for $\alpha \neq 1$, it is a generalization incorporating the distribution q. Thus, our objective function aims to maximize the return while simultaneously maximizing the discounted sum of policy entropy, sample entropy, and the divergence between π and q. In this way, the policy will choose more diverse actions that are far from the samples stored in the replay buffer while maintaining its entropy for better exploration.

4.2 DIVERSE POLICY ITERATION WITH THE PROPOSED OBJECTIVE

In this section, we derive the diverse policy evaluation and diverse policy improvement to maximize the objective function with the sample-aware entropy regularization (3). Note that the sample action distribution q is updated as the iteration goes on. However, it changes very slowly since the buffer

size is much larger than the time steps of one iteration. Hence, for the purpose of proof we regard the action distribution q as a fixed distribution in this section.

First, we define the true diverse Q-function Q^π as $Q^\pi(s_t, a_t) := \frac{1}{\beta} r_t + \mathbb{E}_{\tau_{t+1}\sim\pi}\left[\sum_{l=t+1}^{T-1} \gamma^{l-t-1}\left(\frac{1}{\beta} r_l + \alpha\mathcal{H}(\pi(\cdot|s_l)) + D^\alpha_{JS}(\pi(\cdot|s_l)||q(\cdot|s_l)) + (1-\alpha)\mathcal{H}(q(\cdot|s_l))\right)\right]$. To facilitate implementation, we decompose $D^\alpha_{JS}(\pi(\cdot|s_l)||q(\cdot|s_l))$ as

$$D^\alpha_{JS}(\pi||q) = \alpha\mathbb{E}_{a_l\sim\pi(\cdot|s_l)}[\log R^{\pi,\alpha}(s_l, a_l)] + (1-\alpha)\mathbb{E}_{a_l\sim q(\cdot|s_l)}[\log(1 - R^{\pi,\alpha}(s_l, a_l))] + H(\alpha),$$

where H is the binary entropy function and $R^{\pi,\alpha}$ is the ratio of $\alpha\pi$ to $q^{\pi,\alpha}_{target}$, i.e.,

$$R^{\pi,\alpha}(s_t, a_t) = \frac{\alpha\pi(a_t|s_t)}{\alpha\pi(a_t|s_t) + (1-\alpha)q(a_t|s_t)}. \tag{5}$$

The modified Bellman backup operator for Q^π estimation is given by

$$\mathcal{T}^\pi Q(s_t, a_t) := \frac{1}{\beta} r_t + \gamma\mathbb{E}_{s_{t+1}\sim P}[V(s_{t+1})], \tag{6}$$

where $V(s_t) = \mathbb{E}_{a_t\sim\pi}[Q(s_t, a_t) + \alpha\log R^\alpha(s_t, a_t) - \alpha\log\alpha\pi(a_t|s_t)] + (1-\alpha)\mathbb{E}_{a_t\sim q}[\log(1 - R^\alpha(s_t, a_t)) - (1-\alpha)\log(1-\alpha)q(a_t|s_t)]$ is an estimated diverse state value function, $Q : \mathcal{S}\times\mathcal{A} \to \mathbb{R}$ is an estimated diverse state-action value function, and R^α is an estimate for the ratio function $R^{\pi,\alpha}$.

Proof of the convergence of diverse policy evaluation that estimates Q^π by repeating the Bellman operator (6) is provided in Appendix A. Then, the policy is updated from π_{old} to π_{new} as $\pi_{new} = \arg\max_\pi \; J_{\pi_{old}}(\pi)$, where $J_{\pi_{old}}(\pi)$ is defined as[1]

$$\begin{aligned} J_{\pi_{old}}(\pi(\cdot|s_t)) := \beta\{&\mathbb{E}_{a_t\sim\pi}\left[Q^{\pi_{old}}(s_t, a_t) + \alpha\log R^{\pi,\alpha}(s_t, a_t) - \alpha\log\alpha\pi(a_t|s_t)\right] \\ &+ (1-\alpha)\mathbb{E}_{a_t\sim q}\left[\log(1 - R^{\pi,\alpha}(s_t, a_t)) - \log(1-\alpha)q(a_t|s_t)\right]\}. \end{aligned} \tag{7}$$

The monotone improvement of this step is proved in Appendix A. Now, we can find the optimal policy that maximizes $J(\pi)(= J_\pi(\pi))$ by the following theorem:

Theorem 1 (Diverse Policy Iteration) *By repeating iteration of the diverse policy evaluation and the diverse policy improvement, any initial policy converges to the optimal policy π^* s.t. $Q^{\pi^*}(s_t, a_t) \geq Q^{\pi'}(s_t, a_t)$, $\forall\, \pi' \in \Pi$, $\forall\, (s_t, a_t) \in \mathcal{S}\times\mathcal{A}$. Also, such π^* achieves maximum J, i.e., $J_{\pi^*}(\pi^*) \geq J_\pi(\pi)$ for any $\pi \in \Pi$.*

Proof. See Appendix A.1.

Note that $J_{\pi_{old}}(\pi)$ for diverse policy iteration above requires the ratio function $R^{\pi,\alpha}$ of the current policy π, but we can only estimate $R^{\pi_{old},\alpha}$ for the previous policy π_{old} in practice. Thus, we circumvent this difficulty by defining a practical objective function $J_{\pi_{old}}(\pi)$ given by

$$\tilde{J}_{\pi_{old}}(\pi(\cdot|s_t)) := \beta\mathbb{E}_{a_t\sim\pi}[Q^{\pi_{old}}(s_t, a_t) + \alpha\log R^{\pi_{old},\alpha}(s_t, a_t) - \alpha\log\pi(a_t|s_t)], \tag{8}$$

Regarding the practically computable objective function $\tilde{J}_{\pi_{old}}(\pi)$, we have the following result:

Theorem 2 *Suppose that the policy is parameterized with parameter θ. For parameterized policy π_θ, two objective functions $J_{\pi_{\theta_{old}}}(\pi_\theta(\cdot|s_t))$ and $\tilde{J}_{\pi_{\theta_{old}}}(\pi_\theta(\cdot|s_t))$ have the same gradient direction for θ at $\theta = \theta_{old}$ for all $s_t \in \mathcal{S}$.*

Proof. See Appendix A.2.

By Theorem 2, we can replace the objective function $J_{\pi_{old}}(\pi)$ of policy improvement with the practically computable objective function $\tilde{J}_{\pi_{old}}(\pi)$ for parameterized policy without loss of optimality.

4.3 Diversity Actor Critic Implementation

For implementation, we parameterize π, R^α, Q, and V by neural network parameters θ, η, ϕ, and ψ, respectively. Then, we setup the practical objective (or loss) functions $\hat{J}_\pi(\theta)$, $\hat{J}_{R^\alpha}(\eta)$, $\hat{L}_Q(\phi)$, and $\hat{L}_V(\psi)$ for the parameter update. Detailed DAC implementation based on Section 4 is provided in Appendix B. The propoed DAC algorithm is summarized in Appendix C. Note that DAC becomes SAC when $\alpha = 1$, and becomes standard off-policy RL without entropy regularization when $\alpha = 0$.

[1]Note that if we replace π_{old} with π and view every state s_t as an initial state, then (7) reduces to $J(\pi)$.

5 α-ADAPTATION

In the proposed sample-aware entropy regularization, the weighting factor α plays an important role in controlling the ratio between the policy distribution π and the sample action distribution q. However, it is difficult to estimate optimal α directly. Hence, we further propose an adaptation method for α based on max-min principle widely considered in game theory, robust learning, and decision making problems (Chinchuluun et al., 2008). Since we do not know optimal α, an alternative formulation is that we maximize the return while maximizing the worst-case sample-aware entropy, i.e., $\min_\alpha \mathcal{H}(q^{\pi,\alpha}_{target})$. Then, the max-min approach can be formulated as follows:

$$\max_\pi \mathbb{E}_{\tau_0\sim\pi}\left[\sum_t \gamma^t (r_t + \beta \min_\alpha [\mathcal{H}(q^{\pi,\alpha}_{target}) - \alpha c])\right] \tag{9}$$

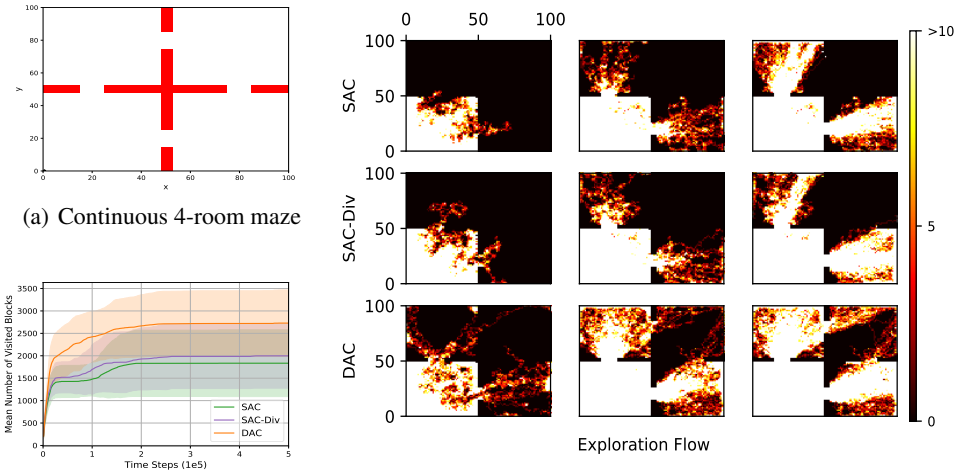
where c is a control hyperparameter for α adaptation. In order to learn α, we parameterize α as a function s_t using parameter ξ, i.e., $\alpha = \alpha_\xi(s_t)$, and implement $\alpha_\xi(s_t)$ with a neural network. Then, ξ is updated to minimize the following loss function deduced from (9):

$$\hat{L}_\alpha(\xi) = \mathbb{E}_{s_t\sim\mathcal{D}}[\alpha_\xi \mathcal{H}(\pi_\theta) + D^{\alpha_\xi}_{JS}(\pi_\theta||q) + (1-\alpha_\xi)\mathcal{H}(q) - \alpha_\xi c] \tag{10}$$

Then, the gradient of $\hat{L}_\alpha(\xi)$ with respect to ξ can be estimated as given in Appendix B.1. Here, all the updates for diverse policy iteration is the same except that α is replaced with $\alpha_\xi(s_t)$.

6 EXPERIMENTS

In this section, we evaluate the proposed DAC algorithm on various continuous-action control tasks and provide ablation study. In order to see the superiority of the sample-aware entropy regularization, we here focus on comparison with two SAC baselines: SAC and SAC-Div. SAC-Div is SAC combined with the method in (Hong et al., 2018) that diversifies policies from buffer distribution by simply maximizing $J(\pi) - \alpha_d D(\pi||q)$ for some divergence D. Note that the key difference between SAC-Div and DAC is that SAC-Div simply adds the single divergence term to the policy objective function $J(\pi)$ in (1), whereas DAC considers the discounted sum of target entropy terms as seen in (3). For SAC-Div, we consider KL divergence (MSE if the policy is Gaussian) and adaptive scale α_d with $\delta = 0.2$ for the divergence term as suggested in (Hong et al., 2018), and its parameter setup is all the same as DAC/SAC. In addition, we provide comparison of DAC to random network distillation (RND) (Burda et al., 2018), which gives an intrinsic reward to explore rare states based on model prediction error, in Appendix F.2 and to other recent RL algorithms in Appendix F.3. The result shows that DAC yields the best performance for all considered tasks as compared to recent RL algorithms. We also provide the source code of DAC implementation that requires Python Tensorflow. Detailed simulation setup for experiments is summarized in Appendix E.



(a) Continuous 4-room maze

(b) Number of state visitation

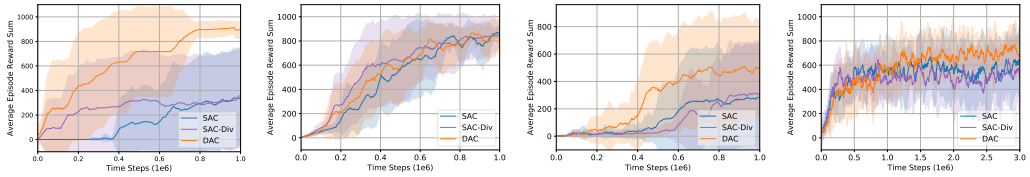
(c) State visit histogram at 5k (left) 50k (middle) 500k (right) steps

Figure 1: Pure exploration task: Continuous 4-room maze

6.1 PURE EXPLORATION COMPARISON

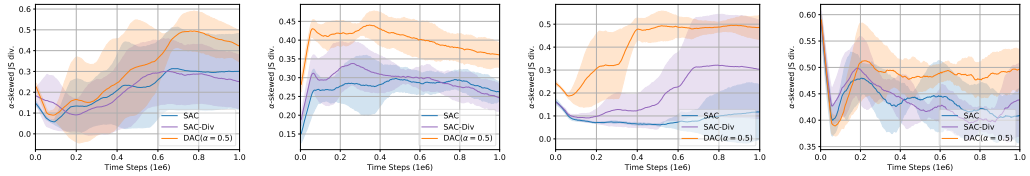
In order to see the exploration performance of DAC ($\alpha = 0.5$) as compared to the SAC baselines, we compare state visitation on a 100×100 continuous 4-room maze task. The maze environment is made by modifying a continuous grid map available at `https://github.com/huyaoyu/GridMap`, and it is shown in Fig. 1(a). State is (x, y) position in the maze, action is (dx, dy) bounded by $[-1, 1]$, and the agent location after the action becomes $(x + dx, y + dy)$. The agent starts from the left lower corner $(0.5, 0.5)$ and explores the maze without any reward, and Fig. 1(b) shows the mean number of new state visitations over 30 seeds, where the number of state visitation is obtained for each integer interval. As seen in Fig. 1(b), DAC visited much more states than SAC/SAC-Div, which means that the exploration performance of DAC is superior to that of the SAC baselines. In addition, Fig. 1(c) shows the corresponding state visit histogram of all seeds. Here, as the color of the state becomes brighter, the state is visited more times. Note that SAC/SAC-Div rarely visit the right upper room even at 500k time steps for all seeds, but DAC starts visiting the right upper room at 5k time steps and frequently visit the right upper room at 500k time steps. Thus, Fig. 1(c) clearly shows that DAC has better sample-efficiency for exploration than SAC/SAC-Div.

6.2 PERFORMANCE COMPARISON WITH THE SAC BASELINES



(a) SparseHalfCheetah-v1 (b) SparseHopper-v1 (c) SparseWalker2d-v1 (d) SparseAnt-v1

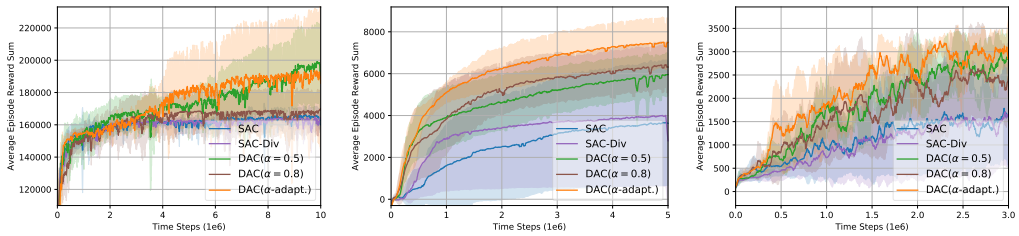
Figure 2: Performance comparison: Fixed α case



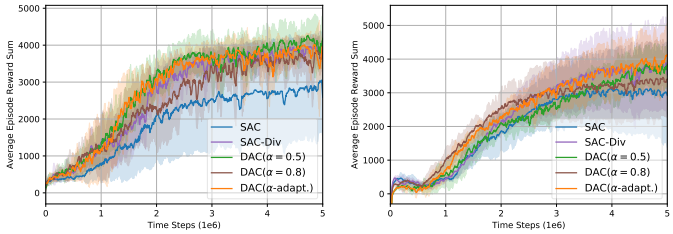
(a) SparseHalfCheetah-v1 (b) SparseHopper-v1 (c) SparseWalker-v1 (d) SparseAnt-v1

Figure 3: α-skewed JS divergence for DAC and SAC/SAC-Div

Fixed α case: Next, in order to see the advantage of the sample-aware entropy regularization for rewarded tasks, we compare the performance of DAC with $\alpha = 0.5$ and the SAC baselines on simple MDP tasks: SparseMujoco tasks. SparseMujoco is a sparse version of Mujoco and the reward is 1 if the agent exceeds the x-axis threshold, otherwise 0 (Hong et al., 2018; Mazoure et al., 2019). The performance results averaged over 10 random seeds are shown in Fig. 2. As seen in Fig. 2, DAC has significant performance gain for most tasks as compared to SAC. On the other hand, SAC-Div also enhances the convergence speed compared to SAC for some tasks, but it fails to enhance the final performance. Fig. 3 shows the α-skewed JS divergence curve ($\alpha = 0.5$) of DAC and SAC/SAC-Div for sparse Mujoco tasks. For SAC/SAC-Div, the ratio function R is estimated separately from (B.2) in Appendix B and the divergence is computed from R. The performance table for all tasks is given by Table F.1 in Appendix F.1. As seen in Fig. 3, the divergence of DAC is much higher than that of SAC/SAC-Div throughout the learning time. It means that the policy distribution of DAC is much far away from the sample action distribution in the replay buffer, so it yields better exploration and performance. Note that the divergence of SAC-Div is also higher than that of SAC in SparseWalker, but it does not lead to performance improvement as seen in Fig. 2(c). On the contrary, the higher divergence of DAC leads to higher performance for most tasks since it guarantees the policy improvement based on diverse policy iteration. Thus, it is seen that the proposed sample-aware entropy regularization is superior to the simple policy entropy regularization of SAC and single divergence regularization of SAC-Div.



(a) HumanoidStandup-v1 (b) DelayedHalfCheetah-v1 (c) DelayedHopper-v1



(d) DelayedWalker2d-v1 (e) DelayedAnt-v1

Figure 4: Performance comparison: Adaptive α case

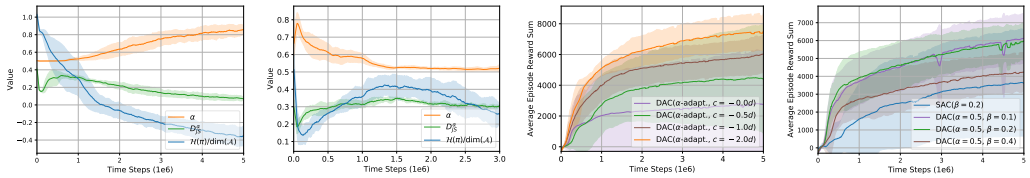
Adaptive α case: Now, we compare the performance of DAC with $\alpha = 0.5, 0.8$, α-adaptation, and the SAC baselines to see the need of α-adaptation. To maintain controllability and prevent saturation of R_η^α, we used regularization for α learning and restricted the range of α as $0.5 \leq \alpha \leq 0.99$ for α adaptation so that a certain level of entropy regularization is enforced. Here, we consider more complicated tasks: HumanoidStandup and delayed Mujoco tasks (DelayedHalfCheetah, DelayedHopper, DelayedWalker2d, and DelayedAnt). HumanoidStandup is one of high-action dimensional Mujoco tasks. Delayed Mujoco tasks suggested by (Zheng et al., 2018; Guo et al., 2018) have the same state-action spaces with original Mujoco tasks but reward is sparsified. That is, rewards for D time steps are accumulated and the accumulated sum is delivered to the agent once every D time steps, so the agent receives no reward during the accumulation time. The performance results averaged over 5 random seeds are shown in Fig. 4. The result of the max average return of these Mujoco tasks for DAC and SAC/SAC-Div is provided in Table F.2 in Appendix F.1. As seen in Fig. 4, all versions of DAC outperform SAC. Here, SAC-Div also outperforms SAC for several tasks, but the performance gain by DAC is much higher. In addition, it is seen that the best α depends on the tasks in the fixed α case. For example, $\alpha = 0.8$ is the best for DelayedHalfCheetah, but $\alpha = 0.5$ is the best for DelayedAnt. Thus, we need to adapt α for each task. Finally, DAC with α-adaptation has the top-level performance for most tasks and the best performance for HumanoidStandup and DelayedHopper tasks. Further consideration for α is provided in Section 6.3.

6.3 Ablation Study

In this section, we provide ablation study for important parameters in the sample-aware entropy regularization.

Weighting factor α: As seen in Section 6.2, α-adaptation is necessary because one particular value of α is not best for all environments. Although the proposed α-adaptation in Section 5 is sub-optimal, it shows good performance across all the considered tasks. Thus, we study more on the proposed α-adaptation and the sensible behavior of sample-awareness in entropy regularization.

Fig. 5(b) and 5(a) shows the averaged learning curve of α, α-skewed JS divergence $D_{JS}(\pi||q)$ and the policy entropy $\mathcal{H}(\pi)$ for DAC with the proposed α-adaptation method on DelayedHalfCheetah and DelayedHopper tasks. Here, we fix the control coefficient c as $-2.0\dim(\mathcal{A})$. As seen in (3), the return, the policy entropy and the JS divergence are intertwined in the cost function and hence their learning curves over time steps are also intertwined through the α-adaptation method as well as the cost function. In the case of DelayedHalfCheetah, the learned policy entropy term decreases



(a) DelayedHalfCheetah-v1 (b) DelayedHopper-v1 (c) Control coefficient c (d) Entropy weighting factor β

Figure 5: Averaged learning curve for ablation study

and the learned α increases as time step goes on. As α increases further to one, the initially nonzero JS divergence term $D_{JS}(\pi||q)$ diminishes to zero. Thus, the sample action distribution from the replay buffer is exploited for roughly initial 2.5M time steps, and then DAC operates like SAC. On the other hand, in the case of DelayedHopper, the learned α gradually decreases approximately from 0.9 to 0.6, and the JS divergence term $D_{JS}(\pi||q)$ is non-zero throughout the learning period. Hence, DAC exploits the sample action distribution for better exploration through the learning period. This adaptive exploitation of the sample-aware entropy from the replay buffer leads to better overall performance across time steps as seen in Fig. 4. Thus, DAC with α-adaptation seems to properly exploit both the policy entropy and the sample action distribution depending on the learning stage.

Control coefficient c: In the proposed α-adaptation (9) and (10), the control coefficient c affects the learning behavior of α. Since $\mathcal{H}(\pi)$ and D_{JS}^{α} are proportional to the action dimension, we tried a few values such as 0, $-0.5d$, $-1.0d$ and $-2.0d$ where $d = \dim(\mathcal{A})$. Fig. 5(c) shows the corresponding performance of DAC with α-adaptation on DelayedHalfCheetah. As seen in Fig. 5(c), the performance depends on the change of c as expected, and $c = -2.0 \cdot \dim(\mathcal{A})$ performs well. The performance comparison with respect to c in other delayed Mujoco environments is provided in Appendix G. We observed that $-2.0d$ performed well for all considered tasks, thus we set $c = -2.0\dim(\mathcal{A})$ in (10).

Entropy weighting factor β: As mentioned in (Haarnoja et al., 2018a), the performance of SAC depends on β. It is expected that the performance of DAC depends on β too. Fig 5(d) shows the performance of DAC with fixed $\alpha = 0.5$ for three different values of β: $\beta = 0.1$, 0.2 and 0.4 on DelayedHalfCheetah. It is seen that the performance of DAC indeed depends on β. Although there exists performance difference for DAC depending on β, the performance of DAC is much better than SAC for a wide range of β. One thing to note is that the coefficient of pure policy entropy regularization term for DAC is $\alpha\beta$, as seen in (3). Thus, DAC with $\alpha = 0.5$ and $\beta = 0.4$ has the same amount of pure policy entropy regularization as SAC with $\beta = 0.2$. However, DAC with $\alpha = 0.5$ and $\beta = 0.4$ has much higher performance than SAC with $\beta = 0.2$, as seen in Fig. 5(d). So, we can see that the performance improvement of DAC comes from joint use of policy entropy $\mathcal{H}(\pi)$ and the sample action distribution from the replay buffer via $D_{JS}^{\alpha}(\pi||q)$. The performance comparison with respect to β for remaining tasks is provided in Appendix G.

7 Conclusion

In this paper, we have proposed a sample-aware entropy framework for off-policy RL to overcome the limitation of simple policy entropy for sample-efficient exploration. With the sample-aware entropy regularization, we can achieve diversity gain by exploiting sample history in the replay buffer in addition to policy entropy. For practical implementation of sample-aware entropy regularized policy optimization, we have proposed the DAC algorithm with convergence proof. We have also provided an adaptation method for DAC to control the ratio of the sample action distribution to the policy action entropy. DAC is an actor-critic algorithm for sample-aware regularized policy optimization and generalizes SAC. Numerical results show that DAC significantly outperforms SAC baselines in Maze exploration and various Mujoco tasks.

REFERENCES

Joshua Achiam and Shankar Sastry. Surprise-based intrinsic motivation for deep reinforcement learning. *arXiv preprint arXiv:1703.01732*, 2017.

Philip Agre and Stanley J Rosenschein. *Computational theories of interaction and agency*. Mit Press, 1996.

Gianluca Baldassarre and Marco Mirolli. *Intrinsically motivated learning in natural and artificial systems*. Springer, 2013.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pp. 1471–1479, 2016.

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

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

Nuttapong Chentanez, Andrew G Barto, and Satinder P Singh. Intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pp. 1281–1288, 2005.

Altannar Chinchuluun, Panos M Pardalos, Athanasios Migdalas, and Leonidas Pitsoulis. *Pareto optimality, game theory and equilibria*. Springer, 2008.

Thomas Degris, Martha White, and Richard S Sutton. Off-policy actor-critic. *arXiv preprint arXiv:1205.4839*, 2012.

Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, Yuhuai Wu, and Peter Zhokhov. Openai baselines. `https://github.com/openai/baselines`, 2017.

Roy Fox, Ari Pakman, and Naftali Tishby. Taming the noise in reinforcement learning via soft updates. *arXiv preprint arXiv:1512.08562*, 2015.

Scott Fujimoto, Herke van Hoof, and Dave Meger. Addressing function approximation error in actor-critic methods. *arXiv preprint arXiv:1802.09477*, 2018.

Tanmay Gangwani, Qiang Liu, and Jian Peng. Learning self-imitating diverse policies. *arXiv preprint arXiv:1805.10309*, 2018.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Shixiang Gu, Timothy Lillicrap, Zoubin Ghahramani, Richard E Turner, and Sergey Levine. Q-prop: Sample-efficient policy gradient with an off-policy critic. *arXiv preprint arXiv:1611.02247*, 2016.

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*, pp. 3846–3855, 2017.

Yijie Guo, Junhyuk Oh, Satinder Singh, and Honglak Lee. Generative adversarial self-imitation learning. *arXiv preprint arXiv:1812.00950*, 2018.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. *arXiv preprint arXiv:1702.08165*, 2017.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018a.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018b.

Seungyul Han and Youngchul Sung. Dimension-wise importance sampling weight clipping for sample-efficient reinforcement learning. *International Conference on Machine Learning*, 2019.

Zhang-Wei Hong, Tzu-Yun Shann, Shih-Yang Su, Yi-Hsiang Chang, Tsu-Jui Fu, and Chun-Yi Lee. Diversity-driven exploration strategy for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 10489–10500, 2018.

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. In *Advances in Neural Information Processing Systems*, pp. 1109–1117, 2016.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kyungjae Lee, Sungyub Kim, Sungbin Lim, Sungjoon Choi, and Songhwai Oh. Tsallis reinforcement learning: A unified framework for maximum entropy reinforcement learning. *arXiv preprint arXiv:1902.00137*, 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.

Manuel Lopes, Tobias Lang, Marc Toussaint, and Pierre-Yves Oudeyer. Exploration in model-based reinforcement learning by empirically estimating learning progress. In *Advances in neural information processing systems*, pp. 206–214, 2012.

Jarryd Martin, Suraj Narayanan Sasikumar, Tom Everitt, and Marcus Hutter. Count-based exploration in feature space for reinforcement learning. *arXiv preprint arXiv:1706.08090*, 2017.

Bogdan Mazoure, Thang Doan, Audrey Durand, R Devon Hjelm, and Joelle Pineau. Leveraging exploration in off-policy algorithms via normalizing flows. *arXiv preprint arXiv:1905.06893*, 2019.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc Bellemare. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 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*, pp. 2775–2785, 2017a.

Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. Trust-pcl: An off-policy trust region method for continuous control. *arXiv preprint arXiv:1707.01891*, 2017b.

Frank Nielsen. On the jensen–shannon symmetrization of distances relying on abstract means. *Entropy*, 21(5):485, 2019.

Brendan O'Donoghue, Remi Munos, Koray Kavukcuoglu, and Volodymyr Mnih. Combining policy gradient and q-learning. *arXiv preprint arXiv:1611.01626*, 2016.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 16–17, 2017.

Konrad Rawlik, Marc Toussaint, and Sethu Vijayakumar. On stochastic optimal control and reinforcement learning by approximate inference. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

John Schulman, Xi Chen, and Pieter Abbeel. Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*, 2017a.

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

Alexander L Strehl and Michael L Littman. An analysis of model-based interval estimation for markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008.

R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. The MIT Press, Cambridge, MA, 1998.

Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, OpenAI Xi Chen, Yan Duan, John Schulman, Filip DeTurck, and Pieter Abbeel. # exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in neural information processing systems*, pp. 2753–2762, 2017.

Emanuel Todorov. General duality between optimal control and estimation. In *2008 47th IEEE Conference on Decision and Control*, pp. 4286–4292. IEEE, 2008.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 5026–5033. IEEE, 2012.

Marc Toussaint. Robot trajectory optimization using approximate inference. In *Proceedings of the 26th annual international conference on machine learning*, pp. 1049–1056. ACM, 2009.

Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando de Freitas. Sample efficient actor-critic with experience replay. *arXiv preprint arXiv:1611.01224*, 2016.

Yuhuai Wu, Elman Mansimov, Roger B Grosse, Shun Liao, and Jimmy Ba. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. In *Advances in neural information processing systems*, pp. 5279–5288, 2017.

Zeyu Zheng, Junhyuk Oh, and Satinder Singh. On learning intrinsic rewards for policy gradient methods. In *Advances in Neural Information Processing Systems*, pp. 4644–4654, 2018.

Brian D Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. PhD thesis, figshare, 2010.

Brian D Ziebart, Andrew Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. 2008.