Curiosity augmented exploration for off-policy reinforcement learning

Ashish Malik *
College of Engineering
Oregon State University
Oregon, OR 97330
malikas@oregonstate.edu

Abstract

Actor-critic methods with greedy explorations have become a standard for achieving state of the art performance in model-free reinforcement learning. Prior works have identified and tackled a few shortcomings of greedy exploration. In this work, we avoid the common pitfalls of greedy exploration using a novel method that decouples exploration-exploitation objectives. The exploration objective is trained with intrinsic rewards (curiosity) to maximize learning and exploitation objective is trained with extrinsic rewards to maximize returns. We show that this decoupling addresses the limitations of greedy exploration. We also show that our method can be combined with any model-free actor critic method. We test its performance on several challenging continuous control tasks. Our empirical results show that it achieves state of the art performance on continuous control benchmarks.

1 Introduction

Reinforcement learning algorithms have gained tremendous success in challenging high-dimensional continuous control problems. State of the art algorithms such as Soft Actor-Critic (SAC, Haarnoja et al. [2018]), Twin Delayed Deterministic Policy Gradient (TD3, Fujimoto et al. [2018]) and Proximal Policy Optimization (PPO, Schulman et al. [2017]) are actor critic methods that rely on greedy exploration for generating trajectories for learning. SAC maximizes the trade-off between expected extrinsic returns and entropy of the learned soft-policy. The entropy augmented learning objective ensures that policy's covariance does not collapse to zero and thus maintains high exploration rate and avoids local minimas. Still, SAC (and other state of the art algorithms) require millions of interactions to learn viable policies in high-dimensional settings. This poor-sample efficiency is a direct result of limitations of greedy exploration, some of which are:

1. *Directionally uninformed exploration*: Adding uncorrelated noise in actions either by sampling from a normal distribution or by adding random noise is a standard practice in deep reinforcement learning. These practices lead to actions from opposite side of the mean to be selected with equal probabilities. High frequency perturbations around mean actions acts as a low pass filter and tend cancel each other out, leading to poor exploration Stulp and Sigaud [2013] Kober and Peters [2014]. Additionally, policies are obtained after incremental updates over previous policies using temporal difference methods. So, action-spaces where the past-policies had high densities are likely explored extensively. Therefore exploration efforts should be focused towards novel action-spaces.
2. *Pessimistic under-exploration* Ciosek et al. [2019]: SAC and TD3 avoid overestimation of Q-estimates and stabilize learning Van Hasselt et al. [2016] using greedy maximization of lower bounds of action-value estimates. Exploration in state-action space with spuriously high estimation

*Please refer to the provided disclaimer in Section 1 of appendix at the end of the document before proceeding.

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

of this lower bound causes the policy's covariance to collapse in that region. This may discourages exploration of new actions and prevent improvement of Q-estimate. SAC's entropy regularlization prevents policy's covariance from reducing to zero but it does'nt directly address this problem.

Similarly, few limitation of standard practices in deep reinforcement learning (DRL) are:

1. *Goal focused exploration*: "Goal-less" exploration based on intrinsic rewards (curiosity) is an active research area in Deep reinforcement learning (DRL). It is inspired by the natural learning in infants and is shown to bring about useful behaviors without any external rewards Burda et al. [2018]. Goal-focused exploration methods (using extrinsic reward signals) explores in an on-policy manner that narrows the active area of exploration. Off-policy exploration on the other hand considerably slows down and tend to destabilize learning Sutton and Barto [2018].
2. *Unimodal policy distribution*: Standard implementations of RL algorithms employ unimodal action-probability distributions (ex. Gaussians). These distributions limits the expressivity of policies to only a single area of interest. Multi-modal action-probability distributions increase exploration efficiency and thus accelerates discovery of good policies Mazoure et al. [2020].

In this work, we present a novel method of off-policy exploration that can be used for any temporal difference based model-free RL algorithms that address the above-mentioned shortcomings. The method is based upon disentangling exploration and exploitation trade-off. Our method learns a separate exploration policy in addition to a target policy (the exploitation policy). The exploration policy is trained with intrinsic rewards and aims at maximizing learning, whereas the target policy is trained by the base-algorithm and focuses on extrinsic rewards. Both exploration policy and target policies are used for active exploration of different areas of interests.

2 Related Works

Existing reinforcement learning methods can be broadly classified in on-policy or off-policy learning methods. On-policy methods update their policy using the data collected from the same policy while off-policy methods can use a different policy or data collected from a previous policy. Both these methods have their pros and cons. Nonetheless both these methods can gain substantial performance benifits by improving their exploration methods to collect data for training. Improving exploration for both on and off-policy RL methods is an active area of research. Some of the prior works include REPS Peters et al. [2010] which formulates exploration as a policy search problem that is iteratively solved using supervised regression. MPO [Abdolmaleki et al., 2018] extends REPS to deep reinforcement learning. Recent work of AWR [Peng et al., 2019] combines the formulation of MPO with soft policy search by using weighted supervised regression with soft policy iteration updates similar to Soft actor critic [Haarnoja et al., 2018]. Similar to our work, Optimisitic actor critic (OAC, Ciosek et al. [2019]) proposes a directionally informed explorer for RL and a KL constraint beteen the target and exploration policy. However, OAC adds a plethora of new hyperparameters which can be diffocult to tune in the real life. Other works that use a KL constraint between the exploration policy and the target policy are MOTO [Mazoure et al., 2020] and MORE [Raffin and Stulp, 2020].

3 Curiosity augmentation

3.1 Notation

Reinforcement learning is a learning paradigm in which an agent interacts the environment to learn extrinsic rewards maximization behavior over a given time-horizon. At each timestep t, the agent observes a state $s_t \sim S$, perform an action $a_t \sim A$ which results in a new environment state $s_{t+1} \sim S$ and a reward signal r_{t+1} from the environment. The agent's mapping of states to actions is called the agent's action policy $\pi : S \rightarrow A$. Returns are defined as the cummulative sum of discounted rewards observed by the agent over a finite or infite temporal horizon,i.e., $G = \sum_{t=0}^{T} \gamma^t r_t$, where $\gamma \in (0, 1]$ is the discount factor. The agent is tasked with learning an action policy that maximizes G. $\rho(s_{t+1}, r_{t+1}|s_t, a_t)$ referes to the environment model which probabilistically maps current states and actions to future states and rewards. We define $z_t \sim Z(s_t, a_t, \mathcal{U})$ as the intrinsic reward that the agent observes by taking action a_t in state s_t, with $\mathcal{U}$ representing all the past state-action transitions observed by the agent. Here $Z(\cdot)$ is the intrinsic reward model whoes output depends on the novelty of the transition, that is, a measure of fulfillment of agent's intrinsic curiosity.

3.2 Curiosity guided exploration policy

Instead of just learning a single action policy, our proposed method jointly trains an exploration focused policy along with the conventional RL policy that maximizes returns. For clarity we refer to the former as exploration policy and the later as target policy. The exploration policy maximizes intrinsic returns under constraints on its entropy and Kullback–Leibler divergence from the target policy. Let π_ϕ be the exploration policy parameterized using ϕ, π_θ be the target policy parameterized using θ and z_t be the intrinsic reward received by the agent at time-step t. The learning/optimization objective for π_ϕ is:

$$\begin{aligned} \text{Maximize}: \quad & J(\pi_\phi) = \sum_{t=0}^{T} \mathbb{E}_{(s_t \sim \rho_{\pi_\phi}, a_t \sim \pi_\phi)}[z(\mathbf{s}_t, \mathbf{a}_t, \mathcal{D})] \\ \textit{s.t.}: \quad & \mathcal{H}\big(\pi_\phi(\cdot|\mathbf{s}_t)\big) \geq \alpha, \quad \forall \mathbf{s}_t \sim \pi_\phi \\ \text{and}, \quad & D_{KL}\Big(\pi_\phi(\cdot|\mathbf{s}_t)\big|\big|\pi_{\boldsymbol{\theta}}(\cdot|\mathbf{s}_t)\Big) \leq \beta \quad \forall \mathbf{s}_t \sim \pi_\phi \end{aligned}$$

Here, T is the maximum trajectory length, $\mathcal{D}$ is the replay buffer that stores the past experiences, and:

- $J(\pi_\phi)$: The objective function can be concave (maximization objective) depending upon the right parameterization class of the exploration policy, intrinsic reward model (and whether it is stationary or non-stationary) and the MDP design. However, in general the objective function is non-convex.
- $\mathcal{H}\big(\pi_\phi(\cdot|\mathbf{s}_t)\big) = \mathbb{E}_{(s_t \sim \rho_{\pi_\phi}, a_t \sim \pi_\phi)}[-\log\big(\pi_\phi(\cdot|\mathbf{s}_t)\big)]$ is the expected entropy of the exploration policy, which is a strictly concave function of exploration policy's parameters for linear parameterization of the exploration policy.
- $D_{KL}\Big(\pi_\phi(\cdot|\mathbf{s}_t)\big|\big|\pi_{\boldsymbol{\theta}}(\cdot|\mathbf{s}_t)\Big)$ is the Kullback–Leibler divergence of the exploration policy from the target policy. It is a convex function of exploration policy's parameters for linear parameterization of exploration policy, and strictly convex if the target policy is held constant [Soch, 2020].

As both π_ϕ and π_θ are parameterized approximations, we modify the above hard constraints into soft penalties to make the optimization problem more tractable for gradient based learning. The above maximization objective is updated to the following (from here onwards, α and β carry a different meaning than above):

$$J(\pi_\phi) = \sum_{t=0}^{T} \mathbb{E}_{(s_t \sim \rho_{\pi_\phi}, a_t \sim \pi_\phi)}\Big[z(\mathbf{s}_t, \mathbf{a}_t, \mathcal{D}) + \alpha\mathcal{H}\big(\pi_\phi(\cdot|\mathbf{s}_t)\big) - \beta D_{KL}\Big(\pi_\phi(\cdot|\mathbf{s}_t)\big|\big|\pi_{\boldsymbol{\theta}}(\cdot|\mathbf{s}_t)\Big)\Big] \quad (1)$$

Here, α and β determines the relative importance of intrinsic rewards, entropy of the exploration policy and Kullback–Leibler divergence of the exploration policy from the target policy. Parameter β puts a soft penalty on straying too far from the target policy. Therefore, the exploration policy explores regions in the vicinity of the target policy, which increases the stability of learning in the target policy (Off policy temporal different learning with function approximiation destabilizes learning, refer Sutton and Barto [2018]). Such KL divergence constraints between exploration and target policies have previously been employed to improve exploration Ciosek et al. [2019]. Using 1, state and state-action value functions for the exploration policy are obtained as:

$$\begin{aligned} V_{\pi_\phi}(s) = \mathbb{E}_{(s_t \sim \rho_{\pi_\phi}, a_t \sim \pi_\phi)}\Bigg[\sum_{t=0}^{T} \gamma^t \Big(z(\mathbf{s}_t, \mathbf{a}_t, \mathcal{D}) + \alpha\mathcal{H}\big(\pi_\phi(\cdot|\mathbf{s}_t)\big) \\ - \beta D_{KL}\Big(\pi_\phi(\cdot|\mathbf{s}_t)\big|\big|\pi_{\boldsymbol{\theta}}(\cdot|\mathbf{s}_t)\Big)\Big|s_0 = s\Bigg] \end{aligned} \quad (2)$$

$$\begin{aligned} Q_{\pi_\phi}(s,a) = \mathbb{E}_{(s_t \sim \rho_{\pi_\phi}, a_t \sim \pi_\phi)}\Bigg[\sum_{t=0}^{T} \gamma^t z(\mathbf{s}_t, \mathbf{a}_t, \mathcal{D}) + \sum_{t=1}^{T} \gamma^t \Big(\alpha\mathcal{H}\big(\pi_\phi(\cdot|\mathbf{s}_t)\big) \\ - \beta D_{KL}\Big(\pi_\phi(\cdot|\mathbf{s}_t)\big|\big|\pi_{\boldsymbol{\theta}}(\cdot|\mathbf{s}_t)\Big)\Big)\Big|s_0 = s, a_0 = a\Bigg] \end{aligned} \quad (3)$$

Using 2 and 3, state value function and state action value function for the exploration policy are related as:

$$V_{\pi_\phi}(\mathbf{s}) = \mathbb{E}_{\mathbf{a}\sim\pi_\phi}\Big[Q_{\pi_\phi}(\mathbf{s},\mathbf{a})\Big] + \alpha\mathcal{H}\big(\pi_\phi(\cdot|\mathbf{s})\big) - \beta D_{KL}\Big(\pi_\phi(\cdot|\mathbf{s})\Big|\Big|\pi_{\boldsymbol{\theta}}(\cdot|\mathbf{s})\Big)$$
$$= \mathbb{E}_{\mathbf{a}\sim\pi_\phi}\Big[Q_{\pi_\phi}(\mathbf{s},\mathbf{a}) - \Big(\alpha+\beta\Big)\log\pi_\phi(\mathbf{a}|\mathbf{s}) + \beta\log\pi_\theta(\mathbf{a}|\mathbf{s})\Big]$$

3.3 Soft policy iteration for exploration policy

The value functions of the exploration policy are calculated based on the objective 1. Iterative computation of the state-value function can be done by repeatedly applying the modified Bellmann operator $\mathcal{T}^{\pi_\phi}$ as:

$$\mathcal{T}^{\pi_\phi}Q_{\pi_\phi}(\mathbf{s}_t,\mathbf{a}_t) \triangleq z(\mathbf{s}_t,\mathbf{a}_t,\mathcal{D}) + \gamma\mathbb{E}_{(s_{t+1}\sim\rho_{\pi_\phi},a_{t+1}\sim\pi_\phi)}\Big[Q_{\pi_\phi}(\mathbf{s}_{t+1},\mathbf{a}_{t+1}) + \alpha\mathcal{H}\big(\pi_\phi(\cdot|\mathbf{s}_{t+1})\big)$$
$$- \beta D_{KL}\Big(\pi_\phi(\cdot|\mathbf{s}_{t+1})\Big|\Big|\pi_{\boldsymbol{\theta}}(\cdot|\mathbf{s}_{t+1})\Big)\Big] \quad (4)$$

Lemma 1. *(Exploration-policy evaluation) Consider the Bellman backup operator in 4 , stationary soft-policies π_ϕ and π_θ, stationary intrinsic reward model z and fixed buffer B and an initial mapping $Q^0_{\pi_\phi}: \mathcal{S}\times\mathcal{A}\to\mathbb{R}$ with $|\mathcal{A}|<\infty$. Let $Q^{k+1}_{\pi_\phi} = \mathcal{T}^{\pi_\phi}Q^k_{\pi_\phi}$ as defined in 4. The sequence $Q^k_{\pi_\phi}$ will converge to the soft Q-value of π_ϕ as $k\to\infty$.*

See appendix for proof.

Using the definition of state value functions and the results of lemma 3, we can express the exploration policy optimization objective's update rule 1 as:

$$\pi^{\text{new}}_\phi = \arg\min_{\pi'_\phi\in\boldsymbol{\Pi}} D_{KL}\left(\pi'_\phi(\cdot|\mathbf{s}_t)\Big|\Big|\frac{\exp\Big(\frac{1}{(\alpha+\beta)}\times\Big[Q_{\pi'_\phi}(\mathbf{s}_t,\cdot) + \beta\mathcal{H}\big(\pi_\theta(\cdot|\mathbf{s}_t)\big)\Big]\Big)}{Z_{\pi_\phi}(\mathbf{s}_t)}\right) \quad (5)$$

where, $\boldsymbol{\Pi}$ is the distribution class for projecting the new policy π^{new}_ϕ, i.e., the set of acceptable exploration policies. Partition function $Z_{\pi_\phi}(\mathbf{s}_t)$ normalizes the exponential of exploration Q-function distribution. Notice that the multiplication factor $\frac{1}{(\alpha+\beta)}$ can be absorbed in the partition function, but we explicitly write it for clarity.

Lemma 2. *(Exploration-policy improvement) Let $\pi'_\phi\in\boldsymbol{\Pi}$ and π^{new}_ϕ be the solution of the minimization objective 5. Then, $Q_{\pi^{new}_\phi}(\mathbf{s}_t,\mathbf{a}_t)\geq Q_{\pi'_\phi}(\mathbf{s}_t,\mathbf{a}_t)$, assuming $|\mathcal{A}|<\infty$, $\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)>0$ and $\pi'_\phi(\mathbf{a}_t|\mathbf{s}_t)>0$, for all $\mathbf{s}_t\in\mathcal{S}$, $\mathbf{a}_t\in\mathcal{A}$ and $\pi'_\phi\in\boldsymbol{\Pi}$.*

See appendix for proof.

Theorem 1. *(Exploration-policy iteration) Repeated applications of policy evaluation (as given in lemma 3) and policy improvement (as given in lemma 4) on any $\pi_\phi\in\Pi$ will converge it to π^*_ϕ such that $Q^{\pi^*_\phi}(s_t,a_t)\geq Q^{\pi_\phi}(s_t,a_t)$ assuming $|\mathcal{A}|<\infty$, $\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)>0$ and $\pi_\phi(\mathbf{a}_t|\mathbf{s}_t)>0$, for all $\mathbf{s}_t\in\mathcal{S}$ and $\mathbf{a}_t\in\mathcal{A}$.*

Proof. Refer Sutton and Barto [2018] for proof of policy iteration. The assumption $|\mathcal{A}|$ bounds the modified intrinsic reward. □

4 Practical curiosity augmentation

4.1 Approximate policy iteration

The above algorithm should be practically approximated to be applicable to large and/or continuous reinforcement learning domains. For that, instead of exact policy iteration which alternates between

running policy evaluation and policy improvement steps till convergence, we will employ approximate policy iteration by partial policy evaluation and policy improvement. Next, we employ function approximators for Q-function and the policies. We also use two separate Q-functions and use the minimum of the two to mitigate the positive bias in the policy improvement step. We also employ target Q networks for bootstrapping Q value learning targets so that we can use first order gradients for updating Q function, which has been shown to improve stability of the learning process. Mnih et al. [2015]. These target networks are updated using exponential moving average of the corresponding Q function parameters.

Consider function approximators for state-action value functions $Q_{\boldsymbol{\psi}_1}$, $Q_{\boldsymbol{\psi}_2}$ parameterized using $\boldsymbol{\psi}_1$, $\boldsymbol{\psi}_2$, exploration-policy $\pi_{\boldsymbol{\phi}}$ parameterized using $\boldsymbol{\phi}$ and target-policy $\pi_{\boldsymbol{\theta}}$ parameterized using $\boldsymbol{\theta}$. The soft-Q functions are trained to minimize the following error:

$$J_Q(\boldsymbol{\psi}_i) = \mathbb{E}_{(s_t,a_t)\sim\mathcal{D}}\left[\frac{1}{2}\Big(Q_{\boldsymbol{\psi}_i}(s_t,a_t) - \hat{Q}(s_t,a_t)\Big)^2\right], \qquad \text{For } i = 1,2 \tag{6}$$

with,

$$\hat{Q}(s_t,a_t) = z(s_t,a_t,\mathcal{D}) + \gamma\mathbb{E}_{s_{t+1}\sim\mathcal{D}}\Bigg[\min\Big(Q_{\bar{\boldsymbol{\psi}}_i}(s_{t+1},a_{t+1})\Big) - (\alpha+\beta)\log\pi_\phi(a_{t+1}|s_{t+1}) + \beta\log\pi_\theta(a_{t+1}|s_{t+1})\Bigg], \qquad \text{For } i = 1,2 \text{ and } a_{t+1}\sim\pi_\phi(s_{t+1}) \tag{7}$$

where, $\mathcal{D}$ is the replay buffer and $Q_{\bar{\boldsymbol{\psi}}_i}$ are the target networks that are updated using a moving average of the Q-function parameters. The above fixed point equation formulation is a quadratic program for linear parameterization of Q function and Q targets and stationary intrinsic reward model. The above equation provides an unbiased estimate for updating the Q-funciton approximators. However, the variance of the KL-divergence estimate increases with the increase in the dimensionality of the policy which can destabilize the training process. Estimates with considerably lesser variance can be computed using the Bregman divergences [Nielsen and Nock, 2010].

The exploration policy is obtained by minimizing the expected KL-divergence with respect out the past experiences of the optimization objective given in 5:

$$J_\pi(\phi) = \arg\min_{\pi'_\phi\in\boldsymbol{\Pi}} D_{KL}\left(\pi'_\phi(\cdot|s_t)\Big|\Big|\frac{\exp\Big(Q_{\pi'_\phi}(s_t,\cdot) + \frac{\beta}{(\alpha+\beta)}\mathcal{H}\big(\pi_\theta(\cdot|s_t)\big)\Big)}{Z_{\pi_\phi}(s_t)}\right) \tag{8}$$

We can use reparameterization to minimize this objective as each component of the target density function is differentiable. Using the reparameterization trick to write expectations over actions as expectations over noise ϵ, the above objective is translated to:

$$\mathbb{E}_{s_t\sim\mathcal{D},\epsilon\sim\mathcal{N}}\Big[(\alpha+\beta)\log\pi_\phi(f_\phi(\epsilon_t,s_t)|s_t) - Q_{\pi_{\boldsymbol{\phi}}}(s_t,f_\phi(\epsilon_t,s_t)) + \beta\log\pi_\theta\big(f_\phi(\epsilon_t,s_t)\big)\Big] \tag{9}$$

with,

$$a_t(s_t,\epsilon) = f_\phi(\epsilon_t,s_t)$$

In our case, we used a squashed gaussian policy, which gives:

$$a_t(s_t,\epsilon) = \tanh\big(\mu_\phi(s_t) + \sigma_\phi(s_t)\odot\epsilon\big), \qquad \epsilon\sim\mathcal{N}(0,I)$$

A two component Gaussian-mixture model $\mathcal{M}_{(\pi_{\boldsymbol{\phi}},\pi_{\boldsymbol{\theta}})}$ is used for sampling actions for generating exploration trajectories. The probability distribution for sampling actions is given as:

$$\mathcal{M}_{(\pi_{\boldsymbol{\phi}},\pi_{\boldsymbol{\theta}})}(a_t|s_t) = \eta\pi_{\boldsymbol{\phi}}(a_t|s_t) + (1-\eta)\pi_{\boldsymbol{\theta}}(a_t|s_t) \tag{10}$$

where, $\eta\in[0,1]$. This Bi-modal action probability distribution better expresses the high-priority action-spaces for exploration in a natural manner. Humans tend to take actions that either lead greatest utilization of learned knowledge or actions that can potentially result in greatest learning of knowledge. The above model explicitly captures both these aspects of exploration. If needed the value of η can be slowly reduced as the agent's learning progresses to change the relative weightage of exploration and exploitation. The complete algorithm is described in Algorithm 1.

4.2 Intrinsic reward model

Algorithm 1: Curiosity augmented off-policy learning

Initialize $\boldsymbol{\phi}$, $\boldsymbol{\theta}$, $\boldsymbol{\psi}_{i=1,2}$, $\bar{\boldsymbol{\psi}}_{i=1,2}$ and $\mathcal{D}$;
Initialize $\bar{\boldsymbol{\psi}}_{i=1,2} = \boldsymbol{\psi}_{i=1,2}$ and set $\boldsymbol{\phi} = \boldsymbol{\theta}$;
Initialize off-policy learning algorithm $\mathcal{O}$, instrinsic reward-model $\mathcal{Z}$;
for *each iteration* **do**
  Initialize $\mathcal{M}$ using η, $\pi_{\boldsymbol{\phi}}$ and $\pi_{\boldsymbol{\theta}}$;
  for *each environment step* **do**
    $a_t \sim \mathcal{M}_{(\pi_{\boldsymbol{\phi}}, \pi_{\boldsymbol{\theta}})}(a_t|s_t)$;
    $s_{t+1} \sim p(s_{t+1}|s_t, a_t)$;
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, a_t, r(s_t, a_t), s_{t+1}))\}$;
    Update $\mathcal{Z}$;
    if *Time to update exploration policy* **then**
      for *Each gradient step* **do**
        $\psi_i \leftarrow \psi_i - \lambda_Q \hat{\nabla}_{\psi_i} J_Q(\psi_i)$, for $i = 1, 2$;
        $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_{\phi_i} J_\pi(\phi)$;
        $\bar{\psi}_i \leftarrow \tau\psi_i + (1-\tau)\bar{\psi}_i$, for $i = 1, 2$;
      end
    end
    if *Time to update target policy* **then**
      $\theta \leftarrow \mathcal{O}(\theta)$;
    end
  end
end

We use a psuedo count based neural density model to quantify the uncertainity in agent's knowledge about the states. We use this uncertainity as a measure of novelty of the visited states and use it to derive intrinsic curiosity rewards for the agent. This method allows us to approximate count based exploration rewards to non-tabular / continuous domain reinforcement learning [Ostrovski et al., 2017].

This method works as follows: We train an underparameterized model to predict next states based on agent current state and actions using the same replay buffer as the exploration policy ($\mathcal{D}$). Underparameterization of the function approximation ensures that the network does not remember all the past transitions and the relative error between different prediction is based on their representation density in the replay buffer (how many times such a transition has been previously encountered). The magnitute of the prediction error is an estimate of the novelty of the state. The neural network for the intrinsic reward model ($\mathcal{Z}$) is therefore trained to minimize the following error:

$$J\big(\mathcal{Z}(\cdot)\big) = \arg\min_{Z} \Big[\frac{1}{2}\big(s_{t+1} - \mathcal{Z}(s_t, a_t)\big)^2\Big], \quad \forall s_t, a_t \sim \mathcal{D}$$

And the intrinsic reward model is obtained as:

$$\begin{aligned} Z(s_{t+1}, s_t, a_t) &= f(s_{t+1}, \mathcal{Z}(s_t, a_t)) \\ &= ||s_{t+1} - \mathcal{Z}(s_t, a_t)||_2^2 \end{aligned}$$

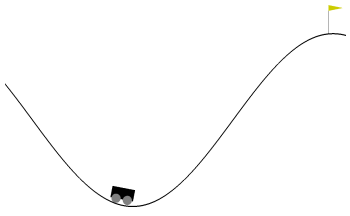
5 Experiments

In this section, we will provide the experimental results of our proposed exploration model and compare it with the state of the art methods in reinforcement learning. Although our proposed method can be used to augment any model-free reinforcement learning algorithm, we will only focus on Soft Actor Critic [Haarnoja et al., 2018] (SAC) (mostly due to time constraints). SAC is an offline reinforcement learing algorithm that provides state of the art leanring performance in multiple continuous control tasks benchmarks. SAC trains a soft-gaussian policy whoes covariance is managed using the hyperparameter α. Higher α means large policy coovariance and thus more exploration, and vice versa. We divide our experimental section into 2 sub-sections: Experiments with the mountainCar environment and popular continuous control benchmarks.

5.1 Mountain car

The continuous variant for the popular mountain car task (shown in Fig. 1) provides the perfect opportunity to gauge the prospects of our proposed model. The agent is tasked to control a car that is initially located at the center of a valley. The goal for the is to reach the peak of the mountain on the right side. The state at any given position is the x position of the car and the car's velocity. The agent is can select an action $a \in [-1, 1]$ to select the directio and magnitude of the force to apply to the car. For each action, the agent recieves a negative reward which is equal to the magnitude of the force taken. The force is not stong enough to take the car all the way up to the mountain by itself, so the agent has to learn to gain momentum by swinging from mountain to mountain to reach the goal.

Figure 1: The MountainCar environmnment



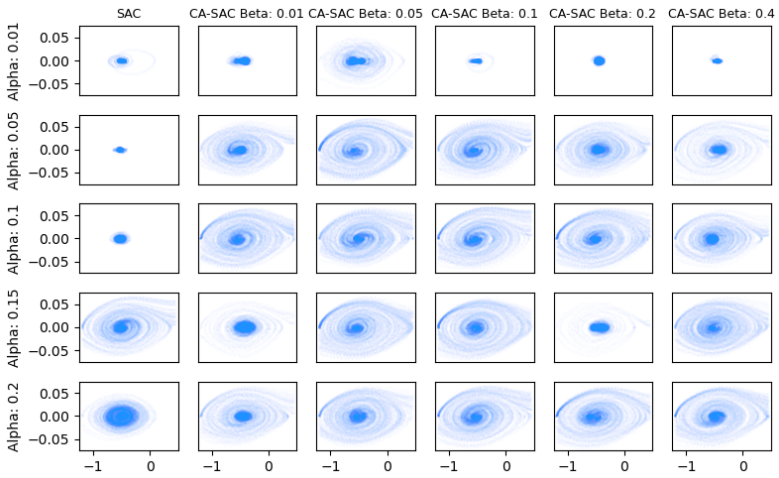


Figure 2: Scatter plots of states visited by different agents with different hyperparameter settings for the first 10^4 environment steps during learning. The x axis marks the position of the car and the y-axis is for the car's velocity.

Table 1: Success rates of finding the goal in the first 10^4 environment steps in the mountain car tasks using different agents and hyperparameters. Each experiment was done using 10 different seed values.

Algorithm	α	β	Success rate
SAC	0.01	-	0.1
CA-SAC	0.01	0.01	0.2
CA-SAC	0.01	0.05	0.1
CA-SAC	0.01	0.1	0.2
CA-SAC	0.01	0.2	0.1
CA-SAC	0.01	0.4	0.2
SAC	0.05	-	0.1
CA-SAC	0.05	0.01	0.6
CA-SAC	0.05	0.05	0.7
CA-SAC	0.05	0.1	0.5
CA-SAC	0.05	0.2	0.5
CA-SAC	0.05	0.4	0.5
SAC	0.1	-	0.2
CA-SAC	0.1	0.01	1.0
CA-SAC	0.1	0.05	0.9
CA-SAC	0.1	0.1	0.6
CA-SAC	0.1	0.2	0.8
CA-SAC	0.1	0.4	0.8
SAC	0.15	-	0.1
CA-SAC	0.15	0.01	0.9
CA-SAC	0.15	0.05	0.9
CA-SAC	0.15	0.1	0.9
CA-SAC	0.15	0.2	0.9
CA-SAC	0.15	0.4	0.8
SAC	0.2	-	0.4
CA-SAC	0.2	0.01	0.9
CA-SAC	0.2	0.05	0.9
CA-SAC	0.2	0.1	1.0
CA-SAC	0.2	0.2	1.0
CA-SAC	0.2	0.4	0.8

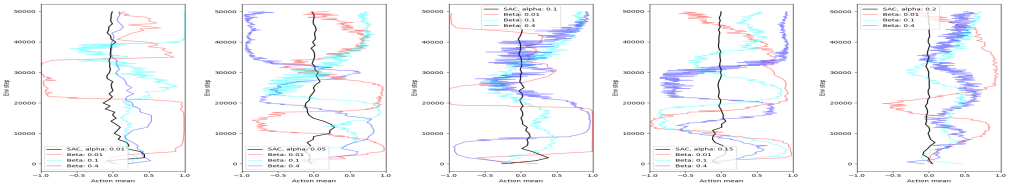
The task is challenging because if the agent doesn't quickly discover that there is a postive reward to be had by reaching the goal, then it decide to not take any actions as all the actions lead to a negative reward. This setting makes the task an excellent test case for our proposed method. Moreover, low dimensional states and actions makes the agent behavior easy to visualize.

The states explored by the pure SAC agent with different α values and curiosity-augmented SAC agent (CA-SAC) with different α and β (KL-divergence penalty coefficient) are shown if Fig. 2. We notice that for small α values, SAC's exploration is focused at the bottom of the valley with small car velocties. As the value of α is increased, the agent begin exploring the sorrounding regions of state-space more actively. For any given α value, the CA-SAC agent explores more actively compared to the vanilla SAC agent. As expected, the exploration incentive for the CA-SAC decreases with increasing β values which corresponds to more concentration in the center of the plots in Fig. 2. The success rates for finding the goal within the first 10^4 environment steps for different agents and hyperparameter settings (using 10 different seed values) is given in table 1. We see that the CA-SAC agents outperforms the vanilla SAC agent in each experiment. Also, the success rate descreases with increasing β values as expected.

Another method to investigate our effect of our proposed method is to visualize how the mean of the soft policies evolve with training for any given state. For this, we plot the mean of the policies of different agents at the starting state of the tasks (Fig. 3). We can draw the following conclusions from the figure:

- SAC agent's mean line is relatively the same for different α values, that is, the SAC agent has a non-goal-focused exploration method. It explores by random sampling. However, the CA-SAC agents acticely look for new states to explores.
- Increasing the β values decreases how agressively the agent looks for novel states. This affect is most prominent with $\alpha = 0.01$ (see next point for the reasoning).

- The underlying policy with greater α becomes inherrently better at randomly encountering novel states, thereby reducing the burden on the exploration policy to look for novelty. This can be seen by the less agressive changes in the mean of CA-SAC agents with $\alpha = 0.2$ compared to the others. However, it should be noted that the CA-SAC agents are still much more efficient at exploration.



(a) $\alpha = 0.01$ (b) $\alpha = 0.05$ (c) $\alpha = 0.1$ (d) $\alpha = 0.15$ (e) $\alpha = 0.2$

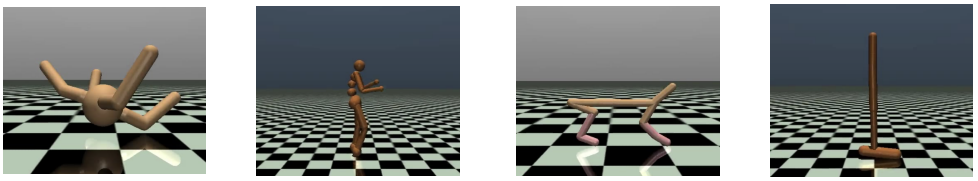
Figure 3: Evolution of the mean of the soft policy for the mountain car task for different agents. The x-axis the action value and the y axis is the environment step number. Zig-zagging in CA-SAC agents is a result of the agent learning a near optimal policy which results in early termination of episodes.

5.2 RL benchmarks

Environment	VGP	PPO	DDPG	TD3	SAC	AWR	CA-SAC (Ours)
Ant	-50	2950	350	4950	5000	5060	**6050**
Hopper	850	2450	1700	2950	3450	3405	**3550**
Humanoid	-	700	4350	80	**8050**	4990	7150
HalfCheetah	850	3200	11600	11200	**13000**	9130	5050

Table 2: Performance comparision of our proposed method with SOTA RL methods on the selected benchmarks. The data is compiled from our own testing, OpenAI [2020] and Peng et al. [2019].

In this subsection, we compare the performance of the agent learned using CA-SAC with state of the art reinforcement learning methods. We evaulate the agents on several popular continouous control benchmarks from the openAI's gym suite. Performance of several popular RL methods and their comparision is provided in Table 2. We see that our proposed method either beats or is comparable to other state of the art methods in high dimensional control tasks. It should be noted here that because CA-SAC explores more widely, its convergence rate, specially in the begining of the training process is slower than other methods. However, this allows it to find more optimal solutions in the long run. The learning curves for the CA-SAC agents are provided in Fig. **??**.



(a) Ant env. (b) Humanoid env. (c) HalfCheetah env. (d) Hopper env.

Figure 4: Suite of continuous control environments used for benchmarking.

6 Conclusions

In this work, we proposed a novel method for exploration using model-free reinforcement learning methods that addresses some of the known shortcomings of current exploration methods. We formulated our method as a contraint optimization problem, discussed the reasoning behind the applied constraints. We also derived ideal case and pratical algorithms for applying the formulation with any off-the-shelf model free reinforcement learning method. We also performed several experiments with the popular continous control tasks. From the benchmarks, we see that our proposed curiosity augmentation method trains policies that surpass or meet the performance of policies trained using the state of the art reinforcement learning methods. However, the benifits come at the cost of increased computations and convergence time. The better performance of the proposed method comes directly from active, novelty focused exploration instead of randomly sampling actions around a mean for exploration.

References

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

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

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

Freek Stulp and Olivier Sigaud. Robot skill learning: From reinforcement learning to evolution strategies. *Paladyn, Journal of Behavioral Robotics*, 4(1):49–61, 2013.

Jens Kober and Jan Peters. Policy search for motor primitives in robotics. In *Learning Motor Skills*, pages 83–117. Springer, 2014.

Kamil Ciosek, Quan Vuong, Robert Loftin, and Katja Hofmann. Better exploration with optimistic actor-critic. *arXiv preprint arXiv:1910.12807*, 2019.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*, 2018.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Bogdan Mazoure, Thang Doan, Audrey Durand, Joelle Pineau, and R Devon Hjelm. Leveraging exploration in off-policy algorithms via normalizing flows. In *Conference on Robot Learning*, pages 430–444. PMLR, 2020.

Jan Peters, Katharina Mulling, and Yasemin Altun. Relative entropy policy search. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. *arXiv preprint arXiv:1806.06920*, 2018.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Antonin Raffin and Freek Stulp. Generalized state-dependent exploration for deep reinforcement learning in robotics. *Arxiv*, 2020.

Joram Soch. General theorems. *The book of statistical proofs*, https://statproofbook.github.io/P/kl-conv.html(P148), 2020.

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.

Frank Nielsen and Richard Nock. Entropies and cross-entropies of exponential families. In *2010 IEEE International Conference on Image Processing*, pages 3621–3624. IEEE, 2010.

Georg Ostrovski, Marc G Bellemare, Aäron Oord, and Rémi Munos. Count-based exploration with neural density models. In *International conference on machine learning*, pages 2721–2730. PMLR, 2017.

OpenAI. Benchmarks for spinning up RL: Popular rl methods. `https://spinningup.openai.com/en/latest/spinningup/bench.html`, 2020.

A Appendix

A.1 Disclaimer

I declare the following to be true to the best of my knowledge:

A.1.1 Novelty of the work

The novelty of the work lies in 3 different areas:

1. Formulation of the exploration policy as a constraint optimization problem with entropy based and KL-divergence based constraints. The formulation derives inspiration from Entropy regularized policy literature and curiosity in reinforcement learning. However, in its entirety the idea is novel.
2. Proof of exploration policy evaluation for the given exploration policy formulation.
3. Proof of exploration policy improvement for the given exploration policy formulation.

Policy iteration can be derived from the policy evaluation and policy improvement as shown in many previous works, so the provided theorem 1 is **not** novel.

This work takes inspiration from the Soft Actor Critic literature but is different from it in many significant ways.

A.1.2 Conception of the idea

The idea of the project was conceived by myself (without any contribution or disscussion with any other person or groups) before the term began. I had some vague ideas of how to prove policy evaluation and improvement but nothing concrete. So most of the theoretical work for this project was done for the class after the final project was announced.

The code base is also self programmed without any significant (> 5 lines of code) help for outside. I used my own previously coded repository of SAC for the barebone starting code.

A.2 Proofs and Figures

Lemma 3. *(Exploration-policy evaluation) Consider the Bellman backup operator in 4 , stationary soft-policies π_ϕ and π_θ, stationary intrinsic reward model z and fixed buffer B and an initial mapping $Q^0_{\pi_\phi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ with $|\mathcal{A}| < \infty$. Let $Q^{k+1}_{\pi_\phi} = \mathcal{T}^{\pi_\phi} Q^k_{\pi_\phi}$ as defined in 4. The sequence $Q^k_{\pi_\phi}$ will converge to the soft Q-value of π_ϕ as $k \to \infty$.*

Proof. Using the definitions of entropy and Kullback–Leibler divergence, we can write 4 as:

$$\mathcal{T}^{\pi_\phi} Q_{\pi_\phi}(\mathbf{s}_t, \mathbf{a}_t) \triangleq z(\mathbf{s}_t, \mathbf{a}_t, \mathcal{D}) + \gamma \mathbb{E}_{(s_{t+1} \sim \rho_{\pi_\phi}, a_{t+1} \sim \pi_\phi)} \Big[Q_{\pi_\phi}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}) - (\alpha + \beta) \log(\pi_\phi(\mathbf{a}_{t+1}|\mathbf{s}_{t+1})) + \beta \log(\pi_{\boldsymbol{\theta}}(\mathbf{a}_{t+1}|\mathbf{s}_{t+1})) \Big] \quad (11)$$

Let,

$$z'(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, \mathcal{D}) = z(\mathbf{s}_t, \mathbf{a}_t, B) + \frac{\gamma}{\beta} \mathbb{E}_{(s_{t+1} \sim \rho_{\pi_\phi}, a_{t+1} \sim \pi_\phi)} \Big[\log(\pi_{\boldsymbol{\theta}}(\mathbf{a}_{t+1}|\mathbf{s}_{t+1})) - (\alpha + \beta) \log(\pi_\phi(\mathbf{a}_{t+1}|\mathbf{s}_{t+1})) \Big] \quad (12)$$

be the modified intrinsic reward signal. The update rule 4 can then be written as:

$$Q_{\pi_\phi}(\mathbf{s}_t, \mathbf{a}_t) \leftarrow z'(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, \mathcal{D}) + \gamma \mathbb{E}_{(s_{t+1} \sim \rho_{\pi_\phi}, a_{t+1} \sim \pi_\phi)} \Big[Q_{\pi_\phi}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}) \Big] \quad (13)$$

Finally, apply the standard policy evaluation convergence results [Sutton and Barto, 2018]. The assumption $|\mathcal{A}| < \infty$ ensures that the modified intrinsic reward signal z' is bounded. □

Lemma 4. *(Exploration-policy improvement) Let $\pi'_\phi \in \mathbf{\Pi}$ and π^{new}_ϕ be the solution of the minimization objective 5. Then, $Q_{\pi^{new}_\phi}(\mathbf{s}_t, \mathbf{a}_t) \geq Q_{\pi'_\phi}(\mathbf{s}_t, \mathbf{a}_t)$, assuming $|\mathcal{A}| < \infty$, $\pi_\theta(\mathbf{a}_t|\mathbf{s}_t) > 0$ and $\pi'_\phi(\mathbf{a}_t|\mathbf{s}_t) > 0$, for all $\mathbf{s}_t \in \mathcal{S}$, $\mathbf{a}_t \in \mathcal{A}$ and $\pi'_\phi \in \mathbf{\Pi}$.*

Proof. The update rule 5 can be rewritten as:

$$\begin{aligned}\pi^{\text{new}}_\phi(\cdot|s_t) &= \arg\min_{\pi'_\phi \in \mathbf{\Pi}} D_{KL}\Big(\pi'_\phi(\cdot|\mathbf{s}_t)\Big|\Big| \exp\Big(\frac{1}{(\alpha+\beta)} \times \Big[Q_{\pi'_\phi}(\mathbf{s}_t, \cdot) + \beta\mathcal{H}\big(\pi_\theta(\cdot|\mathbf{s}_t)\big)\Big]\Big) - \log Z_{\pi_\phi}(\mathbf{s}_t)\Big)\Big) \\ &= \arg\min_{\pi'_\phi \in \mathbf{\Pi}} J_{\pi_\phi}(\pi'_\phi(\cdot|\mathbf{s}_t))\end{aligned}$$

We have, $J_{\pi_\phi}(\pi^{\text{new}}_\phi(\cdot|\mathbf{s}_t)) \leq J_{\pi'_\phi}(\pi_\phi(\cdot|\mathbf{s}_t))$ as we can always select $\pi^{new}_\phi = \pi'_\phi \in \mathbf{\Pi}$. So,

$$\begin{aligned}&\mathbb{E}_{a_t \sim \pi^{\text{new}}_\phi}\Big[\log \pi^{\text{new}}_\phi(\mathbf{a}_t|\mathbf{s}_t) - \frac{1}{(\alpha+\beta)} Q_{\pi_\phi}(\mathbf{s}_t, \mathbf{a}_t) - \frac{\beta}{(\alpha+\beta)} \log \pi_\theta(a_t|s_t) + \log Z_{\pi_\phi}(\mathbf{s}_t)\Big] \\ \leq\quad &\mathbb{E}_{\mathbf{a}_t \sim \pi'_\phi}\Big[\log \pi'_\phi(\mathbf{a}_t|\mathbf{s}_t) - \frac{1}{(\alpha+\beta)} Q_{\pi_\phi}(\mathbf{s}_t, \mathbf{a}_t) - \frac{\beta}{(\alpha+\beta)} \log \pi_\theta(a_t|s_t) + \log Z_{\pi_\phi}(\mathbf{s}_t)\Big]\end{aligned}$$

As partition function Z_{π_ϕ} does not depend on $\mathbf{a}_t$ and $(\alpha+\beta) > 0$,

$$\begin{aligned}&\mathbb{E}_{a_t \sim \pi^{\text{new}}_\phi}\Big[\log \pi^{\text{new}}_\phi(\mathbf{a}_t|\mathbf{s}_t) - \frac{1}{(\alpha+\beta)} Q_{\pi_\phi}(\mathbf{s}_t, \mathbf{a}_t) - \beta \log \pi_\theta(a_t|s_t)\Big] \\ &\qquad\qquad \leq \mathbb{E}_{\mathbf{a}_t \sim \pi'_\phi}\Big[\log \pi'_\phi(\mathbf{a}_t|\mathbf{s}_t) - \frac{1}{(\alpha+\beta)} Q_{\pi_\phi}(\mathbf{s}_t, \mathbf{a}_t) - \beta \log \pi_\theta(a_t|s_t)\Big]\end{aligned}$$

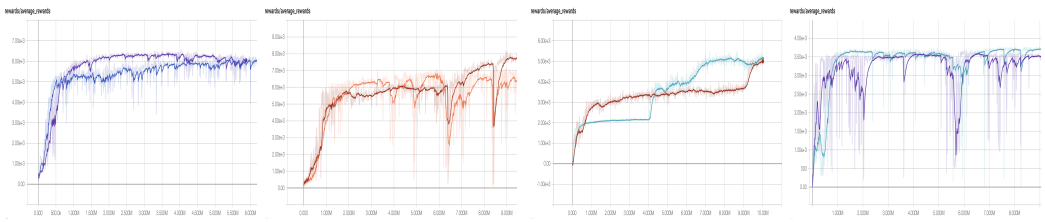
The above inequality is reduced to:

$$\mathbb{E}_{a_t \sim \pi^{\text{new}}_\phi}\Big[Q_{\pi_\phi}(\mathbf{s}_t, \mathbf{a}_t) - (\alpha+\beta) \log \pi^{\text{new}}_\phi(\mathbf{a}_t|\mathbf{s}_t) + \beta \log \pi_\theta(a_t|s_t)\Big] \geq V_{\pi_\phi}(\mathbf{s}_t) \tag{14}$$

Next, considering the Bellman equation with intrinsic reward signal for value functions given in 2 and 3 :

$$\begin{aligned}Q_{\pi_\phi}(s_t, a_t) &= z(\mathbf{s}_t, \mathbf{a}_t, \mathcal{D}) + \gamma \mathbb{E}_{s_{t+1} \sim \rho}\big[V_{\pi_\phi}(\mathbf{s}_{t+1})\big] \\ &\leq z(\mathbf{s}_t, \mathbf{a}_t, \mathcal{D}) + \gamma \mathbb{E}_{s_{t+1} \sim \rho}\Big[\mathbb{E}_{a_{t+1} \sim \pi^{\text{new}}_\phi}\Big[Q_{\pi_\phi}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}) \\ &\qquad - (\alpha+\beta) \log \pi^{\text{new}}_\phi(\mathbf{a}_{t+1}|\mathbf{s}_{t+1}) + \beta \log \pi_\theta(a_t|s_t)\Big]\Big] \\ &= z(\mathbf{s}_t, \mathbf{a}_t, \mathcal{D}) + \gamma \mathbb{E}_{s_{t+1} \sim \rho, a_{t+1} \sim \pi^{\text{new}}_\phi}\Big[\beta \log \pi_\theta(a_t|s_t) - (\alpha+\beta) \log \pi^{\text{new}}_\phi(\mathbf{a}_{t+1}|\mathbf{s}_{t+1})\Big] \\ &\qquad + \gamma \mathbb{E}_{s_{t+1} \sim \rho, a_{t+1} \sim \pi^{\text{new}}_\phi}\Big[Q_{\pi_\phi}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1})\Big] \\ &\quad \vdots \\ &\leq Q_{\pi^{\text{new}}_\phi}(\mathbf{s}_t, \mathbf{a}_t)\end{aligned}$$

by repeatedly expanding the expected Q values using the soft Bellmann equations and using the bound in equation 14. □



(a) Ant env. (b) Humanoid env. (c) HalfCheetah env. (d) Hopper env.

Figure 5: Learning curves produced by 2 different random seeds of CA-SAC for different continuous control benchmarks. For each of these experiments we used neural networks with 2 hidden layers with 256 dimensions and rectified linear units as non-linearities for both the Q functions and the policies. The reward function used 2 hidden layers with 16 dimensions and relu activations. We also set $\alpha = \beta = 0.2$ (i.e. we didn't perform any hyperparameter tuning for better performance).