

# Off-Policy Deep Reinforcement Learning with Analogous Disentangled Exploration

Anji Liu, Yitao Liang, Guy Van den Broeck

Department of Computer Science, University of California, Los Angeles  
anjiliu219@gmail.com, {yliang, guyvdb}@cs.ucla.edu

## Abstract

Though disentangling behavior policy from target policy offers freedom to optimize both policies with respect to their own objectives, naively doing so can lead to inefficient learning or stability issues. To mitigate this problem, our method *Analogous Disentangled Actor-Critic* (ADAC) proposes to design an analogous pair of policies/actors and bind their corresponding critics. We empirically evaluate ADAC on 14 continuous-control games and report the state-of-the-art on 10 of them. We further demonstrate ADAC, when paired with intrinsic rewards, outperform alternatives in exploration-challenging games.

## 1 Introduction

Reinforcement learning (RL) studies the control problem where an agent tries to navigate through an unknown environment (Sutton and Barto 1998). The agent attempts to maximize its cumulative rewards through an iterative trial-and-error learning process (Arulkumaran et al. 2017). Recently, we have seen many successes of applying RL to challenging simulation (Mnih et al. 2015) and real-world problems (Silver et al. 2017), of which many are games. Inherently, RL consists of two distinct but closely related objectives: learn the best possible policy from the gathered samples (*exploitation*) and collect new samples effectively (*exploration*). While the *exploitation* step shares certain similarities with tasks such as supervised learning, *exploration* is unique, essential, and is often viewed as the backbone of many successful RL algorithms (Haarnoja et al. 2018a).

In order to explore and find novel states, it is crucial to incorporate randomness when interacting with the environment. Thanks to its simplicity, injecting noise into action space (Lillicrap et al. 2015; Fujimoto, Hoof, and Meger 2018) or parameter space (Plappert et al. 2018) is widely used to implicitly construct behavior policies from target policies. In most prior work, the injected noise has a mean of zero, such that the updates to the target policy have no bias. The stability of noise-based exploration, which is obtained from its non-biased nature, makes it a safe exploration strategy. However, the noise-based approach is generally less effective

since it is neither aware of potentially rewarding actions nor guided by the exploration-oriented target.

To tackle the above problem, two orthogonal lines of approaches have been proposed. One of them considers extracting more information from the current knowledge (gathered samples). Specifically, energy-based reinforcement learning algorithms such as Soft Actor-Critic (SAC) (Haarnoja et al. 2018a) and Soft-Q Learning (SQL) (Sutton and Barto 2018) learn to capture potentially rewarding actions through its energy objective. A second line of work considers using external guidance to aid exploration. Essentially they formulate some intuitive tendencies in exploration as an additional reward function called intrinsic reward (Bellemare et al. 2016; Houthoof et al. 2016). Guided by these auxiliary tasks, RL algorithms tend to act curiously, substantially improving exploration of the state space.

Despite their promising exploration efficiency, both lines of work fail to best *exploit* the collected samples and turn them into the highest-performance policy, as their policy often executes sub-optimal actions. To avoid this undesirable tradeoff between *exploration* and *exploitation*, several attempts have been made to separately design two policies, of which one aims to gather the most informative examples (and hence is commonly referred as the behavior policy) while the other keeps its self optimal with respect to the knowledge from the gathered samples (and hence is usually referred as the target policy) (Colas, Sigaud, and Oudeyer 2018; Beyer et al. 2019). In this way, disentangled objectives can be designed and separately applied to the two policies.

However, naively disentangling behavior policy from the target policy would render the policy update process unstable. For example, when disentangled naively, the policy for exploration and that for exploitation tends to differ due to their different objectives. This difference between the two policies is known to potentially result in catastrophic learning failure. To mitigate this problem, we propose *analogous disentanglement*, which consists of two main improvements to the disentangled learning structure: the behavior policy and target policy are represented by the same neural network, and their corresponding critics are both calculated with regards to the target policy, yet for different reward functions.

## 2 Preliminaries

In this section, we introduce the reinforcement learning (RL) setting we address in this paper, as well as some background concepts that we utilize to build our method.

### 2.1 RL with Continuous Control

In a standard *reinforcement learning* (RL) setting, the environment is commonly formalized as a *Markov Decision Process* (MDP), which can be succinctly defined as a 5-tuple  $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma \rangle$ , where  $\mathcal{S}$  denotes the state set,  $\mathcal{A}$  action set,  $\mathcal{R}$  the environment-defined reward function,  $\mathcal{P}$  transition function, and  $\gamma$  the discount factor that balances the future rewards and the present ones. In all the games considered in this paper, actions are assumed to be continuous.

The agent strives to learn the *optimal policy* that maximizes the expected return  $J(\pi) = \mathbb{E}_{s_0 \sim \rho_0, a_t \sim \pi, s_{t+1} \sim \mathcal{P}, r_t \sim \mathcal{R}} [\sum_{t=0}^{\infty} \gamma^t r_t]$ , where  $\rho_0$  is the initial state distribution and  $\gamma \in [0, 1]$  is the discount factor balancing the priority of short and long-term rewards. For continuous control, the deterministic policy  $\pi$  (also known as the actor in the actor-critic framework) parameterized by  $\theta$  can be updated by taking the gradient  $\nabla_{\theta} J(\pi)$ . According to the deterministic policy gradient theorem (Silver et al. 2014),  $\nabla_{\theta} J(\pi) = \mathbb{E}_{(s,a) \sim \rho_{\pi}} [\nabla_a Q_{\mathcal{R}}^{\pi}(s, a) \nabla_{\theta} \pi(s)]$ , where  $\rho_{\pi}$  denotes the state-action marginals of the trajectory distribution induced by  $\pi$ , and  $Q_{\mathcal{R}}^{\pi}$  denotes the state-action value function (also known as the critic in the actor-critic framework), which represents the expected return under the reward function specified by  $\mathcal{R}$  when performing action  $a$  at state  $s$  and following policy  $\pi$  afterwards. Intuitively, it measures how preferable executing action  $a$  is at state  $s$  with respect to the policy  $\pi$  and reward function  $\mathcal{R}$ . Following (Bellman 1966), we additionally introduce the Bellman operator, which is commonly used to update the  $Q$ -function. The Bellman operator  $\mathcal{T}_{\mathcal{R}}^{\pi}$  uses  $\mathcal{R}$  and  $\pi$  to update an arbitrary value function  $Q$ , which is not necessarily defined with respect to the same  $\pi$  or  $\mathcal{R}$ . For example, the outcome of  $\mathcal{T}_{\mathcal{R}_1}^{\pi_1} Q_{\mathcal{R}_2}^{\pi_2}(s_t, a_t)$  is defined as  $\mathcal{R}_1(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1} \sim \mathcal{P}, a_{t+1} \sim \pi_1} [Q_{\mathcal{R}_2}^{\pi_2}(s_{t+1}, a_{t+1})]$ . By slightly abusing notations, we further define the outcome of  $\mathcal{T}_{\mathcal{R}_1}^{\max} Q_{\mathcal{R}_2}^{\pi_2}(s_t, a_t)$  as  $\mathcal{R}_1(s_t, a_t) + \gamma \max_{a_{t+1}} \mathbb{E}_{s_{t+1} \sim \mathcal{P}} [Q_{\mathcal{R}_2}^{\pi_2}(s_{t+1}, a_{t+1})]$ . Someone also call  $\mathcal{T}_{\mathcal{R}}^{\max}$  as the Bellman optimality operator.

### 2.2 Off-policy Learning and Behavior Policy

To aid exploration, it is a common practice to construct/store more than one policy for the agent. Off-policy learning (a.k.a. actor-critic methods) (Watkins and Dayan 1992) allows us to make a clear separation between the *target policy* and the *behavior policy*. Note the discussion in the earlier subsection is largely around the target policy. Thus, starting from this point, to avoid confusion,  $\pi$  is reserved to only denote the target policy and notation  $\mu$  is dedicated to the behavior policy. Due to the policy separation, the target policy is instead resorting on the estimates calculated with regards to samples collected by the behavior policy, that is, the deterministic policy gradient mentioned above is approximated as

$$\nabla_{\theta} J(\pi) \approx \mathbb{E}_{(s,a) \sim \rho_{\mu}} [\nabla_a Q_{\mathcal{R}}^{\pi}(s, a) \nabla_{\theta} \pi(s)]. \quad (1)$$

where  $\mathcal{R}$  is the environment-defined reward. DDPG (Lillicrap et al. 2015) is among the most notable off-policy algorithms that capitalize on this idea. TD3 (Fujimoto, Hoof, and Meger 2018) further improves upon DDPG by mitigating DDPG’s function approximation errors.

### 2.3 Expressive Behavior Policies through Energy-Based Representation

One promising way to design an exploration-oriented behavior policy without external guidance, which is usually in the form of intrinsic reward, is by increasing the expressiveness of  $\mu$  to capture information about potentially rewarding actions. Energy-based representations have recently been more and more chosen as the target form to construct an expressive behavior policy. Since its first introduction by (Ziebart 2010) to achieve maximum-entropy reinforcement learning, several additional prior work keeps improving upon this idea. Among them, the most notable ones include Soft Q-Learning (SQL) (Haarnoja et al. 2017), EntRL (Schulman, Chen, and Abbeel 2017), and Soft Actor-Critic (SAC) (Haarnoja et al. 2018b). Collectively, they have achieved competitive results in many benchmark games. Formally, the energy-based behavior policy is defined as

$$\mu(a | s) \propto \exp(Q(s, a)), \quad (2)$$

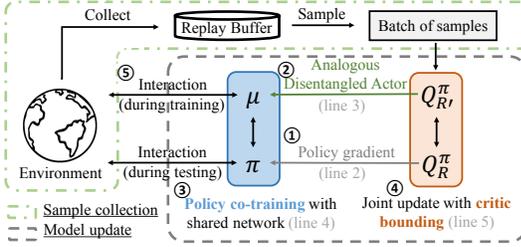
where  $Q$  is commonly selected to be the target critic  $Q_{\mathcal{R}}^{\pi}$  in prior work (i.e., (Haarnoja et al. 2018b; Haarnoja et al. 2018a)). Various efficient samplers have been proposed to approximate the distribution specified in Eq. 2 (Haarnoja et al. 2018a; Haarnoja et al. 2017). Among them, (Haarnoja et al. 2017)’s Stein Variational Gradient Descent (SVGD) (Liu and Wang 2016) based sampler is especially worth noting as it has the potential to approximate any arbitrarily complex and multi-model behavior policy. Given this, we also choose it to sample the behavior policy in our proposed ADAC.

Additionally, we want to highlight an intriguing property of SVGD that is critical for understanding why we can perform *analogous disentangled exploration* effectively. Intuitively, SVGD transforms a set of particles to match a distribution. In the context of RL, following Amortized SVGD (Feng, Wang, and Liu 2017), we use a neural network sampler  $f_{\varphi}(s, \xi)$  ( $\xi \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ) to approximate Eq. 2, which is done by minimizing the KL divergence between two distributions. According to (Feng, Wang, and Liu 2017),  $f_{\varphi}$  is updated according to the following gradient:

$$\begin{aligned} \nabla_{\varphi} \mathcal{L}(\varphi) \approx & \mathbb{E}_{s, \xi} \left[ \sum_{j=1}^K \underbrace{[\mathcal{K}(a, a'_j) \nabla_{a'_j} Q(s, a'_j)]}_{\text{term 1}} \right. \\ & \left. + \beta \cdot \underbrace{[\nabla_{a'_j} \mathcal{K}(a, a'_j)]}_{\text{term 2}} \Big|_{a=f_{\varphi}(s, \xi)} \frac{\partial f_{\varphi}(s, \xi)}{\partial \varphi} \right] / K, \end{aligned} \quad (3)$$

where  $\mathcal{K}$  is a positive definite kernel<sup>1</sup>, and  $\beta$  is a hyper-parameter to make optimality-expressiveness tradeoff. The

<sup>1</sup>Formally, in ADAC, we define the kernel as  $\mathcal{K}(a, \hat{a}_i) = \frac{1}{\sqrt{2\pi(d/K)}} \exp\left(-\frac{\|a - \hat{a}_i\|^2}{2(d/K)^2}\right)$ , where  $d$  is the number of dimensions of the action space.



**Figure 1:** Block diagram of ADAC, which consists of the sample collection phase and the model update phase. Model updates are performed sequentially from ① to ④. Each update step’s corresponding line in Algorithm 1 is shown in brackets.

**Algorithm 1** Training procedure of the Analogous Disentangled Actor-Critic. Correspondence with Figure 1 is given after “//”.

- 1: **Input:** A minibatch of samples  $\mathcal{B}$ , actor model  $f_\varphi$  (represents the target policy  $f_\varphi^\pi$  as well as the behavior policy  $f_\varphi^\mu$ ), critic model  $Q_\phi^\pi$  and  $Q_\psi^\mu$  ( $\varphi, \phi$ , and  $\psi$  are neural network parameters).
- 2:  $\nabla_\varphi f_\varphi^\pi \leftarrow$  the deterministic policy gradient of  $Q_\phi^\pi$  with respect to  $\pi$  (Eq. 1). // **target policy update**
- 3:  $\nabla_\varphi f_\varphi^\mu \leftarrow$  gradient of  $Q_\psi^\mu$  with respect to the behavior policy  $\mu$  (Eq. 3, Section 3.1) // **behavior policy learning**
- 4: Update  $f$  according to  $\nabla_\varphi f_\varphi^\pi$  and  $\nabla_\varphi f_\varphi^\mu$  // **policy co-training**
- 5: Update  $Q_\phi^\pi$  and  $Q_\psi^\mu$  to minimize the mean squared error on  $\mathcal{B}$  with respect to the target  $\mathcal{T}^\pi Q_\phi^\pi$  and  $\mathcal{T}^\pi Q_\psi^\mu$ , respectively. // **value update with critic bounding**

intrinsic connection between Eq. 3 and the deterministic policy gradient (Eq. 1) is introduced in (Haarnoja et al. 2017) and (Feng, Wang, and Liu 2017): the first term of the gradient represents a combination of deterministic policy gradients weighted by  $\mathcal{K}$ , while the second term of the gradient represents an entropy maximization objective.

### 3 Analogous Disentangled Actor Critic

This section introduces our proposed method *Analogous Disentangled Actor-Critic* (ADAC).

#### 3.1 Algorithm Overview

Figure 1 provides a diagram overview of ADAC. Same with prior off-policy algorithms (e.g., DDPG), during training ADAC alternates between two main procedures, namely *sample collection* (green box with dotted line) and *model updates* (dashed gray box).  $\mathcal{R}$  is the environment-defined reward function and  $\mathcal{R}'$  may contain additional intrinsic rewards  $\mathcal{R}^{in}$  on top of the rewards provided by the environment, that is  $\mathcal{R}' := \mathcal{R} + \mathcal{R}^{in}$ .

To achieve disentanglement, we maintain two pairs of actor-critic  $\langle \mu, Q_{\mathcal{R}'}^\pi \rangle$  and  $\langle \pi, Q_{\mathcal{R}}^\pi \rangle$ . Mainly two places manifest the analogous property of our method. First, both actors/policies ( $\mu$  and  $\pi$ ) are represented by the same neural network  $f$ , where  $\mu(s) := f_\varphi(s, \xi)|_{\xi \sim \mathcal{N}(\mathbf{0}, \mathbf{I})}$  and  $\pi(s) := f_\varphi(s, \xi)|_{\xi = [0, 0, \dots, 0]^T}$ . Here we highlight the key reasons for this particular design to give intuitions, while its details are deferred in Section 3.2. The choice of  $\xi$  acts as a restriction of the “distance” between  $\pi$  and  $\mu$ , which stabilizes the policy

learning process. Furthermore, the gradients of  $\pi$  and  $\mu$  have a key connection that motivates this design and improves the expressiveness of  $\mu$  under certain conditions.

The second exhibit of our method’s analogous nature lies on our designed critics  $Q_{\mathcal{R}}^\pi$  and  $Q_{\mathcal{R}'}^\pi$ , which are based on reward function  $\mathcal{R}$  and  $\mathcal{R}'$  respectively yet are both computed with regard to the target policy  $\pi$ . As a standard approach,  $Q_{\mathcal{R}}^\pi$  approximates the task-defined objective that the algorithm aims to maximize. On the other hand,  $Q_{\mathcal{R}'}^\pi$  is a behavior critic that can be shown to be both *explorative* and *stable* theoretically (Section 3.3) and empirically (Section 4.3). Note that when not using intrinsic reward (i.e.,  $\mathcal{R} = \mathcal{R}'$ ), the two critics are degraded to be identical to one another and in practice when that happens we only store one of them.  $\pi$  and  $\mu$  interact with the environment in the training and evaluation phase, respectively.

To better appreciate our method, it is not enough to only gain an overview about our actors and critics in isolation. Given this, we then formalize the connections between the actors and the critics as well as the objectives that are optimized during the model update phase (Figure 1). As defined above,  $\pi$  is the exploitation policy that aims to maintain optimality throughout the learning process, which is best optimized using the deterministic policy gradient (Eq 1), where  $Q_{\mathcal{R}}^\pi$  is used as the referred critic (① in Figure 1). On the other hand, for the sake of expressiveness, the energy-based objective (Eq 2) is a good fit for  $\mu$ . To further encourage exploration, we use  $Q_{\mathcal{R}'}^\pi$  in the objective, which gives  $\mu(a | s) \propto \exp(Q_{\mathcal{R}'}^\pi(s, a))$  (② in Figure 1). Since both policies share the same network  $f$ , critic optimization is done by maximizing

$$J_\pi(\varphi) + J_\mu(\varphi), \quad (4)$$

where the gradients of both terms are defined by Eqs 1 and 3, respectively. In particular, we set  $\pi(s) := f_\varphi(s, \xi)|_{\xi = [0, 0, \dots, 0]^T}$  in Eq 1 and  $Q := Q_{\mathcal{R}'}^\pi$  in Eq 3. As illustrated in Algorithm 1 (line 5), we update  $Q_{\mathcal{R}}^\pi$  and  $Q_{\mathcal{R}'}^\pi$  with the target  $\mathcal{T}^\pi Q_{\mathcal{R}}^\pi$  and  $\mathcal{T}^\pi Q_{\mathcal{R}'}^\pi$ , on the collected samples using the mean squared error loss, respectively.

In the sample collection phase,  $\mu$  interacts with the environment and the gathered samples are stored in a replay buffer (Mnih et al. 2013) for later use in the model update phase. Given state  $s$ , actions are sampled from  $\mu$  with a two-step procedure: (i) sample  $\xi \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ , and then plug the sampled  $\xi$  in  $f_\varphi(s, \xi)$  to get its output  $\hat{a}$ , and (ii) regard  $\hat{a}$  as the center of kernel  $\mathcal{K}(\cdot, \hat{a})^1$  and sample an action  $a$  from it.

On the implementation side, ADAC is compatible with any existing off-policy actor-critic model for continuous control: it directly builds upon them by inheriting their actor  $\pi$  (which is also their target policy) and critic  $Q_{\mathcal{R}}^\pi$ . To be more specific, ADAC merely adds a new actor  $\mu$  to interact with the environment and a new critic  $Q_{\mathcal{R}'}^\pi$ , that guides  $\mu$ ’s updates on top of the base model, along with the constraints/connections enforced between the inherited and the new actor and between the inherent and the new critic (i.e., policy co-training and critic bounding). In other words, modifications made by ADAC would not conflict with the originally proposed improvements on the base model. In our experiments, two base models (i.e., DDPG (Lillicrap et al. 2015) and TD3

(Fujimoto, Hoof, and Meger 2018)) are adopted.

### 3.2 Stabilizing Policy Updates by Policy Co-training

Although the energy-based behavior policy defined by Eq 2 is sufficiently expressive to capture potentially rewarding actions, it may still not be helpful for  $\pi$ 's learning process: being expressive also means that  $\mu$  is often significantly different from  $\pi$ , leading to collect substantially biased samples, which in turn render the learning process of  $Q_{\mathcal{R}}^{\pi}$  unstable and vulnerable to catastrophic failure (Sutton, Szepesvári, and Maei 2008). To be more specific, since the difference between  $\pi$  and an expressive  $\mu$  is more than some zero-mean random noise, the state marginal distribution  $\rho_{\mu}$  defined with respect to  $\mu$  can potentially diverge greatly from the state marginal distribution  $\rho_{\pi}$  defined with respect to  $\pi$ . Since  $\rho_{\pi}$  is not directly accessible, as shown in Equation 1, the gradients of  $\pi$  are approximated using samples from  $\rho_{\mu}$ . When the approximated gradients constantly deviate significantly from the true values (i.e. the approximated gradients are biased), the updates to  $\pi$  essentially become inaccurate and hence ineffective. This suggests that a brutal act of disentangling behavior policy from target policy alone is not a guarantee of improved training efficiency or final performance.

Therefore, to mitigate the aforementioned problem, we would like to reduce the distance between  $\mu$  and  $\pi$ , which naturally reduces the KL-divergence between distribution  $\rho_{\mu}$  and  $\rho_{\pi}$ . One straightforward approach to reduce the distance between the two policies is to reduce the randomness of  $\mu$ , for example by lowering  $\beta$ . However, this inevitably sacrifices  $\mu$ 's expressiveness, which in turn would also harm ADAC's competitiveness. Alternatively, we propose *policy co-training* to best maintain the expressiveness of  $\mu$  while also stabilizing it by restricting it with regards to  $\pi$ , which is motivated by the intrinsic connection between Eqs 1 and 3 (Section 2.3). As described in Section 3.1, we reiterate here that in a nutshell, both policies are modeled by the same network  $f$  and are distinguished only with their different input to  $\xi$ . During training,  $f$  is updated to maximize Eq 4. The method to sample actions from  $\mu$  is described in the 4th paragraph of Section 3.1.

We further justify the above choice by demonstrating that the imposed restrictions on  $\mu$  and  $\pi$  only has minor influence on  $\pi$ 's optimality and  $\mu$ 's expressiveness. To argue for this point, we need to revisit Equation 3 for one more time:  $\pi$  can be viewed as being updated with  $\beta = 0$ , whereas  $\mu$  is updated with  $\beta > 0$ . Intuitively, this makes policy  $\pi$  optimal since its action is not affected by the entropy term (the second term).  $\mu$  is still expressive since only when the input random variable  $\xi$  is close to the zero vector, it will be significantly restricted by  $\pi$ . In Section 4.1, we will empirically demonstrate policy co-training indeed reduces the distance between  $\mu$  and  $\pi$  during training, fulfilling its mission.

Additionally, *policy co-training* enforces the underlying relations between  $\pi$  and  $\mu$ . Specifically, policy co-training forces  $\pi$  to be contained in  $\mu$  since  $[0, 0, \dots, 0]^T$  is the highest-density point of  $\mathcal{N}(\mathbf{0}, \mathbf{I})$ , and sampling  $\xi$  from  $\mathcal{N}(\mathbf{0}, \mathbf{I})$  is likely to generate actions close to that from  $\pi$ . This matches the intuition that  $\pi$  and  $\mu$  should share similar-

ties: actions proposed by  $\pi$  is rewarding (with respect to  $\mathcal{R}$ ) and thus should be frequently executed by  $\mu$ .

### 3.3 Incorporating Intrinsic Reward in Behavior Critic via Critic Bounding

With the help of disentanglement as well as *policy co-training*, which makes  $\mu$  and  $\pi$  analogous, we manage to design an expressive behavior policy that not only explores effectively but also helps stabilize  $\pi$ 's learning process. In this section, we aim to achieve the same objective – stability and expressiveness – on a different subject, the behavior critic  $Q_{\mathcal{R}'}^{\pi}$ .

As introduced in Section 3.1,  $\mathcal{R}$  is the environment-defined reward function, while  $\mathcal{R}'$  consists of an additional exploration-oriented reward term  $\mathcal{R}^{in}$ . As hinted by the notations, ADAC's target critic  $Q_{\mathcal{R}}^{\pi}$  and behavior critic  $Q_{\mathcal{R}'}^{\pi}$  are defined with regard to the same policy but updated differently according to the following

$$Q_{\mathcal{R}}^{\pi} \leftarrow \mathcal{T}_{\mathcal{R}}^{\pi} Q_{\mathcal{R}}^{\pi}; \quad Q_{\mathcal{R}'}^{\pi} \leftarrow \mathcal{T}_{\mathcal{R}'}^{\pi} Q_{\mathcal{R}'}^{\pi}, \quad (5)$$

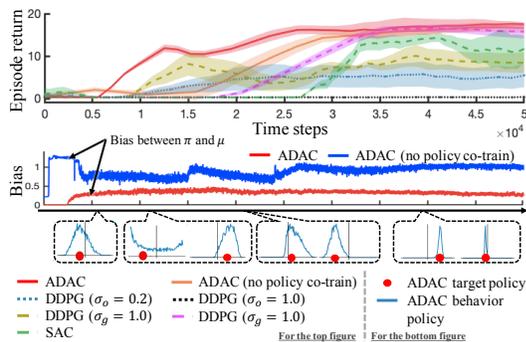
Updates are performed through minibatches in practice. Note that when no intrinsic reward is used, Eq 5 becomes trivial and the two critics ( $Q_{\mathcal{R}}^{\pi}$  and  $Q_{\mathcal{R}'}^{\pi}$ ) are identical. Therefore, we only consider the case where intrinsic reward exists in the following discussion.

While it is natural that the target critic is updated using the target policy, it may seem counterintuitive that the behavior critic is also updated using the target policy. Given that  $\mu$  is updated following the guidance (i.e., the energy-based objective) of  $Q_{\mathcal{R}'}^{\pi}$ , we do so to prevent  $\mu$  from diverging disastrously from  $\pi$ . The following theorem provides rigorous justifications.

**Theorem 1.** *Let  $\pi$  be a greedy policy w.r.t.  $Q_{\mathcal{R}}^{\pi}$  and  $\mu$  be a greedy policy w.r.t.  $Q_{\mathcal{R}'}^{\pi}$ . Assume  $Q_{\mathcal{R}'}^{\pi}$  is optimal w.r.t.  $\mathcal{T}_{\mathcal{R}'}^{\pi}$  and  $\mathcal{R}' \geq \mathcal{R}$  in all states. We have the following results. First,  $\mathbb{E}_{\rho_{\pi}}[\mathcal{T}_{\mathcal{R}}^{max} Q_{\mathcal{R}}^{\pi} - Q_{\mathcal{R}}^{\pi}]$ , a proxy of training stability, is lower bounded by  $\mathbb{E}_{\rho_{\mu}}[\mathcal{T}_{\mathcal{R}}^{max} Q_{\mathcal{R}}^{\pi} - Q_{\mathcal{R}}^{\pi}] + \mathbb{E}_{\rho_{\pi}}[\mathcal{R}] - \mathbb{E}_{\rho_{\mu}}[\mathcal{R}]$ . Second,  $\mathbb{E}_{\rho_{\mu}}[\mathcal{T}_{\mathcal{R}}^{max} Q_{\mathcal{R}}^{\pi} - Q_{\mathcal{R}}^{\pi}]$ , a proxy of training effectiveness, is lower bounded by  $\mathbb{E}_{\rho_{\pi}}[\mathcal{T}_{\mathcal{R}}^{max} Q_{\mathcal{R}}^{\pi} - Q_{\mathcal{R}}^{\pi}] + \mathbb{E}_{\rho_{\pi}}[\mathcal{R} - \mathcal{R}']$ .*

Due to the space limit, we leave the proof to a longer version of this paper. Here, we only focus on the insights conveyed by Theorem 1. We first examine the assumptions made by the theorem. While other assumptions are generally satisfiable and are commonly made in the RL literature (Munos 2007), the assumption on the rewards ( $\mathcal{R}'(s) \geq \mathcal{R}(s)$  for all  $s$ ) seems restrictive. However, since most intrinsic rewards are strictly greater than zero (e.g., (Houthoofd et al. 2016; Fu, Co-Reyes, and Levine 2017)), the condition can be easily satisfied in practice.

To better understand the theorem, we first provide interpretations of the key components. According to the definition of the Bellman optimality operator (Section 2.1),  $\mathcal{T}_{\mathcal{R}}^{max} Q_{\mathcal{R}}^{\pi} - Q_{\mathcal{R}}^{\pi}$  quantifies the improvement on  $Q$  after performing one value iteration (Bellman 1966) step (w.r.t.  $\mathcal{R}$ , where all states receive a hard update), which is a proxy of the policy improvement in the near future. Therefore,  $\mathbb{E}_{\rho}[\mathcal{T}_{\mathcal{R}}^{max} Q_{\mathcal{R}}^{\pi} - Q_{\mathcal{R}}^{\pi}]$  is the expected policy improvement under state-action distribution  $\rho$  in the near future.



**Figure 2:** Learning curves of DDPG-based ADAC and baselines on the modified CartPole game. In addition, ADAC’s target (red dots) and behavior policies (solid blue curves) at different timesteps and the bias between them are plotted below the learning curves.

We formalize training stability as the lower bound of the expected policy improvement under  $\rho_\pi$  (i.e.,  $\mathcal{T}_{\mathcal{R}}^{max} Q_{\mathcal{R}}^\pi - Q_{\mathcal{R}}^\pi$ ). Theorem 1 provides a lower bound that consists of two parts. The second part,  $\mathbb{E}_{\rho_\pi}[\mathcal{R}] - \mathbb{E}_{\rho_\mu}[\mathcal{R}]$ , is greater than zero since  $\pi$  is optimized to maximize the cumulative reward of  $\mathcal{R}$  while  $\mu$  is not. On the other hand, the first term can be viewed as the improvement of  $Q_{\mathcal{R}}^\pi$  during training since  $\rho_\mu$  is the training sample distribution of ADAC. Therefore, the improvement of  $Q_{\mathcal{R}}^\pi$  during training lower bounds the expected policy improvement under  $\rho_\pi$ , which represents stability.

Conversely, the lower bound on  $\mathbb{E}_{\rho_\pi}[\mathcal{T}_{\mathcal{R}}^{max} Q_{\mathcal{R}}^\pi - Q_{\mathcal{R}}^\pi]$  states the effectiveness of the training procedure of ADAC. Note that most intrinsic rewards are designed to be small in states that are frequently visited. Therefore, when the state-action pairs  $\pi$  would visit states that are frequently visited by  $\mu$ , which is promised using the policy co-training approach,  $\mathbb{E}_{\rho_\pi}[\mathcal{R} - \mathcal{R}'] = -\mathbb{E}_{\rho_\pi}[\mathcal{R}^{in}]$  will be small. Therefore, even if  $\rho_\pi$  and  $\rho_\mu$  are not identical, as long as  $\mu$  allows substantial visitations of high probability states in  $\rho_\pi$  to make  $\mathbb{E}_{\rho_\pi}[\mathcal{R}^{in}]$  sufficiently small, improvement when trained on the samples  $\rho_\mu$  would be almost as large as the training improvement on the target distribution  $\rho_\pi$ .

## 4 Experiments

In this section, we first investigate the behavior of our analogous disentangled behavior policy (Section 4.1). Next, we perform an empirical evaluation of ADAC without intrinsic rewards on 14 standard continuous-control benchmarks (Section 4.2). Finally, encouraged by its promising performance and to further justify the critic bounding method, we examine ADAC with intrinsic rewards in 4 sparse-reward and hence exploration-heavy environments (Section 4.3).

### 4.1 Analysis of Analogous Disentangled Behavior Policy

Since we are largely motivated by the potential luxury of designing an expressive exploration strategy offered by the disentangled nature of our framework, it is natural we are first interested in investigating how well our behavior policy lives up to our expectation. Yet as discussed in Section 4.2,

in order to aid stable policy updates, we specifically put some restraints on our behavior policy, deliberately making it analogous of the target policy, which means our behavior policy may not be as expressive as otherwise. Given this, we start this set of empirical experiments with investigating whether our behavior policy is still expressive enough, which is measure by its coverage (i.e., does it explore a wide enough action/policy space outside the current target strategy). To further examine the influence of our added restraints, we examine the policy network’s stability (i.e., does the *policy co-training* lowers the bias between two policies and stabilize the  $\pi$ ’s learning process). Finally, we focus on the effectiveness of our behavior policy by measuring the overall performance (i.e., does ADAC’s exploration strategy efficiently lead the target policy to iteratively converge to a more desirable local optimum).

**Setup** For the sake of ease of illustration, we choose a straightforward game, namely CartPole, as our demonstration bed. The goal in this game is to balance the pole that is attached to a cart by applying left/right force. For the compatibility with continuous control and a better modeling of real-world implications, we modified CartPole’s original discrete action space and added effort penalty to the rewards. To demonstrate the advantages of our behavior policy, we choose DDPG with two commonly-used existing exploration strategies as the main baselines, i.e. Gaussian noise ( $\sigma_g$ ) and Ornstein-Uhlenbeck process noise ( $\sigma_o$ ) (Uhlenbeck and Ornstein 1930), both with two variance levels 0.2 and 1.0. For the virtue of fair comparison, we only present DDPG-based ADAC here (or simply ADAC later in this subsection). To further demonstrate the benefits from disentanglement, we choose SAC as another baseline. As discussed earlier in related works, SAC similarly also utilizes energy-based policies, yet opposite to our approach its exploration is embedded into its target policy.

**Empirical Insights** First, see the cells at the bottom of Figure 2, which are snapshots of the behavior and target policy at different training stages. As suggested by the wide bell shape of the solid blue curves ( $\mu$ ) at the first cell, our behavior policy acts curiously when ignorant about the environment, extensively exploring all possible actions including those that are far away from the target policy (represented by the red dots). Yet having such a broad coverage alone is still not sufficient to overcome the beginning trap of getting stuck in the deceiving local-optimum of constantly exerting  $a \in [-0.5, 0.5]$ . As suggested by the bimodal shape of the solid blue curve ( $\mu$ ) in the second cell, after acquiring a preliminary understanding of the environment the agent starts to form preference for some actions when exploring. Almost at the same time, the target policy no longer stays close to 0.0 (represented by the vertical line), suggesting that the behavior policy is effective in leading the target policy towards a more desirable place. This can be further corroborated by what is suggested from the third and fourth cell. In the late stage, besides being able to balance the pole, our agent even manages to learn exerting actions with small absolute value from time to time to avoid the effort penalty.

Other than its expressiveness, stability critically influences ADAC’s overall performance, which is by design controlled

**Table 1:** Continuous-control performance in 14 benchmark environments. Average episode return ( $\pm$  standard deviation) over 20 trials are reported. Bold indicates the best average episode return.  $\dagger$  indicates the better performance between ADAC(TD3) and its base model TD3. Similarly,  $*$  indicates the better performance between ADAC (DDPG) and its base model DDPG. In all three cases, values that are statistically insignificantly different ( $>0.05$  in t-test) from the respective should-be indicated ones are denoted as well.

Environment	ADAC (TD3)	ADAC (DDPG)	TD3	DDPG	SAC	PPO
RoboschoolAnt	2219 $\pm$ 373	838.1* $\pm$ 97.1	<b>2903</b> $\dagger\pm$ 666	450.0 $\pm$ 27.9	<b>2726</b> $\pm$ 652	1280 $\pm$ 71
RoboschoolHopper	<b>2299</b> $\dagger\pm$ 333	766.5* $\pm$ 10	<b>2302</b> $\dagger\pm$ 537	543.8 $\pm$ 307	2089 $\pm$ 657	1229 $\pm$ 345
RoboschoolHalfCheetah	1578 $\dagger\pm$ 166	<b>1711</b> * $\pm$ 95	607.2 $\pm$ 246.2	441.6 $\pm$ 120.4	807.0 $\pm$ 252.6	1225 $\pm$ 184.2
RoboschoolAtlasForwardWalk	<b>234.6</b> $\dagger\pm$ 55.7	186.7* $\pm$ 37.9	190.6 $\pm$ 50.1	52.63 $\pm$ 26.2	126.0 $\pm$ 47.1	107.6 $\pm$ 29.4
RoboschoolWalker2d	<b>1769</b> $\dagger\pm$ 452	<b>1564</b> * $\pm$ 651	995.1 $\pm$ 146.3	208.7 $\pm$ 137.1	1021 $\pm$ 263	578.9 $\pm$ 231.3
Ant	3353 $\pm$ 847	1226* $\pm$ 18	4034 $\dagger\pm$ 517	370.5 $\pm$ 223	<b>4291</b> $\pm$ 1498	1401 $\pm$ 168
Hopper	<b>3598</b> $\dagger\pm$ 374	374.5* $\pm$ 36.5	2845 $\pm$ 609	38.93 $\pm$ 0.88	<b>3307</b> $\pm$ 825	1555 $\pm$ 458
HalfCheetah	9392 $\pm$ 199	2238* $\pm$ 40	10526 $\dagger\pm$ 2367	1009 $\pm$ 49	<b>11541</b> $\pm$ 2989	881.7 $\pm$ 10.1
Walker2d	<b>5122</b> $\dagger\pm$ 1314	1291* $\pm$ 42	4630 $\dagger\pm$ 778	186.2 $\pm$ 33.3	4067 $\pm$ 1211	1146 $\pm$ 368
InvertedPendulum	<b>1000</b> $\dagger\pm$ 0	<b>1000</b> * $\pm$ 0	<b>1000</b> $\dagger\pm$ 0	<b>1000</b> * $\pm$ 0	<b>1000</b> $\pm$ 0	98.90 $\pm$ 2.08
InvertedDoublePendulum	<b>9359</b> $\dagger\pm$ 0.17	9334* $\pm$ 1.39	7665 $\pm$ 566	27.20 $\pm$ 2.61	9353 $\pm$ 2896	98.90 $\pm$ 5.88
BipedalWalker	<b>309.8</b> $\dagger\pm$ 15.6	-52.77* $\pm$ 1.94	288.4 $\dagger\pm$ 51.25	-123.90 $\pm$ 11.17	<b>307.2</b> $\pm$ 57.92	266.9 $\pm$ 28.52
BipedalWalkerHardcore	<b>-10.76</b> $\dagger\pm$ 27.70	-98.52 $\pm$ 3.21	-57.97 $\pm$ 21.08	-50.05* $\pm$ 10.27	-127.4 $\pm$ 45.2	-105.3 $\pm$ 22.2
LunarLanderContinuous	<b>290.0</b> $\dagger\pm$ 50.9	85.67* $\pm$ 23.42	<b>289.7</b> $\dagger\pm$ 54.1	-65.89 $\pm$ 96.48	283.3 $\pm$ 69.29	59.32 $\pm$ 68.44

by the proposed *policy co-training* approach. To examine its effect, we perform an ablation study about it. To be more specific, we compare ADAC with against ADAC without *policy co-training*<sup>2</sup>. The effect of critic bounding is measured by the *bias* between  $\pi$  and  $\mu$ , which is shown in the middle of Figure 2. We can see that the bias of ADAC is much lower than its variant without policy co-training. Additionally, the constraint added by policy co-training does not affect the expressiveness of  $\mu$ , which is suggested by the behavior policies rendered below in Figure 2.

Finally we pay attention to the learning curves in Figure 2: ADAC exceeds baselines in both learning efficiency (i.e. being the first to consistently accumulate positive rewards) and final performance. Unlike our behavior policy, exploration through random noise is unguided, resulting in either wasted exploration on unpromising regions or insufficient exploration on rewarding areas. This largely explains the noticeable performance gap between DDPG with random noise and ADAC. On the other side, SAC bears an expressive policy similar to our behavior policy. However, suffering from no separate behavior policy, to aid exploration, SAC has to consistently take sub-optimal actions into account. In other words, different from ADAC, SAC cannot fully exploits its learned knowledge in its value functions) to construct its target policy, leading to a performance inferior to ADAC’s.

## 4.2 Comparison with the State of the Art

Though well-suited for illustration, CartPole alone is not challenging and generalized enough to fully manifest ADAC’s competitiveness. In this subsection, we present that ADAC can be the state-of-the-art in standard benchmarks.

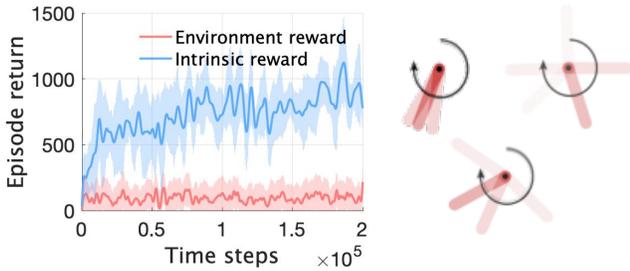
**Setup** To demonstrate the generality of our method, we construct a 14-task testbed suite composed of qualitatively diverse continuous-control environments from the OpenAI Gym toolkit (Brockman et al. 2016) On top of the two baselines adopted earlier (i.e. DDPG and SAC), we further include TD3 (Fujimoto, Hoof, and Meger 2018), which improves

upon DDPG by addressing some of its function approximation errors, PPO (Schulman et al. 2017), which is regarded as one of the most stable and efficient on-policy policy gradient algorithm, and GEP-PG (Colas, Sigaud, and Oudeyer 2018), which combines Goal Exploration Process (P er e et al. 2018) with policy gradient to perform curious exploration as well as stable learning. However, we compare with GEP-PG in only one game since it is only evaluated in two games in their original paper, of which only one overlaps with our game suit. Additionally, since the GEP part of the algorithm needs hand-crafted ex-ploration goals, we are not able to run their model on new experiments since it is nontrivial to generalize their experiments in other games To best reproduce the rest’s performance, we use their original open-source implementations if released; otherwise, we build our own versions after the most-starred third-party implementations in GitHub. Furthermore, we fine-tune their hyper-parameters around the values reported in the respective literature and only coarsely tune the hyper-parameters introduced by ADAC All experiments are run for 1 million time-steps, or until reaching performance convergence, whichever happens earlier.<sup>3</sup>

**Empirical Insights** Table 1 corroborates ADAC’s competitiveness stemmed from its disentangled nature over existing methods. More importantly, these results reveal two desirable properties of ADAC’s full compatibility with existing off-policy methods. First, ADAC consistently outperforms the method it is based on. As indicated by the  $*$  symbols, compared to its base model, DDPG-based ADAC achieves statistically better performance on more than 85%(12/14) of the benchmarks and obtains identical performance on one of the remaining two. TD3-based ADAC manages to achieves statically better or comparable performance over its base model on more than 78%(11/14) of the benchmarks; see the  $\dagger$  symbols. Second, ADAC retains the benefits of improvements developed by the base model themselves. This is best illustrated by TD3-based ADAC’s performance superiority over DDPG-based ADAC.

<sup>3</sup>For fairness concern, we do not use intrinsic reward throughout this section, since most baseline approaches are not able to naturally incorporate them during learning.

<sup>2</sup>Two separate neural networks are used to store  $\pi$  and  $\mu$  when policy co-training is not used.



**Figure 3:** Illustration of how intrinsic reward contaminates the environment-defined optimality in PendulumSparse. Fooled into collecting more intrinsic rewards rather than environment rewards (see the learning curves on the left), the agent constantly alternates between spinning the pendulum and barely moving it (see the snapshots of the target policy on the right), making no real progress.

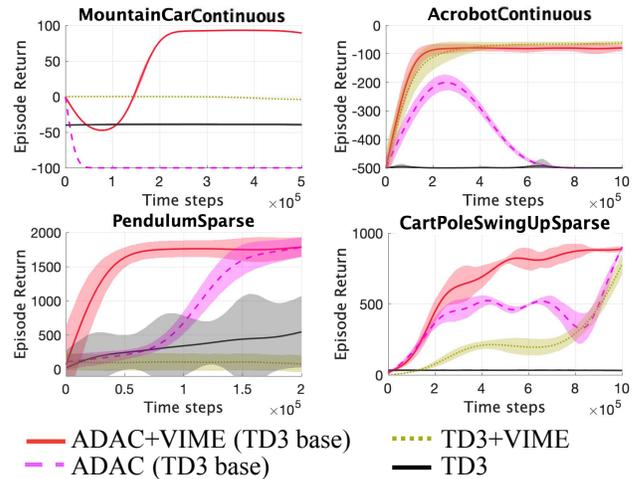
We would like to further call readers’ attention on our comparison of ADAC against SAC since they both use energy-based behavior policy. This comparison also reveals the benefit brought by the disentangled structure and the analogous actors and critics. ADAC (TD3) achieves better average performance over SAC on 71%(10/14) of the benchmarks, indicating the effectiveness of our proposed analogous disentangled structure. Despite also using the disentanglement idea, we can only compare with GEP-PG (Colas, Sigaud, and Oudeyer 2018) on HalfCheetah due to the reasons mentioned earlier. In HalfCheetah, GEP-PG achieves 6118 cumulative reward, while ADAC (TD3) achieves 9392, showing superiority over GEP-PG. As also acknowledged in its paper, GEP-PG lags behind SAC in performance, which suggests that simply disentangling behavior policy from target policy in a brutal way does not guarantee competitive performance. Rather, to design effective disentangled actor-critic, we should also pay attention to how to best restrict some components.

When considering all reported methods together, TD3-based ADAC obtains the most number of state-of-the-art results; as indicated in bold, it is the best performer (or statistically comparable with the best) on more than 71%(10/14) of the benchmarks.

### 4.3 Evaluation in Sparse-Reward Environments

Encouraged by the promising results observed on the benchmarks, in this subsection we evaluate ADAC under more challenging environments, in which rewards are barely provided. This set of experiments aim to test ADAC’s exploration capacity under extreme settings. Furthermore, we also see them fit as demonstration beds to present ADAC’s natural compatibility with intrinsic methods.

**Setup** To the surprise of many, sparse-reward environments turn out to be relatively unpopular in commonly-used RL toolkits. Besides including the classic MountainCarContinuous and Acrobot (after converting its action space to be continuous), to construct a decently sized testing suite, we further hand-craft new games, namely PendulumSparse and CartPoleSwingUpSparse by sparsifying the rewards in the existing environments. Sparsifying is achieved mainly through suppressing the original rewards until reaching some pre-defined threshold. Due to their dependency on environment-provided rewards as feedback signals, most model-free RL al-



**Figure 4:** Learning curves for four sparse-reward games. Lines denote the average over 20 trials and the shaded areas represent the range of one standard deviation.

gorithms suffer significant performance degradation in these sparse-reward games. In this situation, resorting to intrinsic methods (IM) for additional signals has been widely considered as the go-to solution. Among a wide variety of IM methods, we adopt Variational Information Maximization Exploration (VIME) (Houthoofd et al. 2016) as our internal reward generator for its consistent good performance on a wide variety of exploration-challenging games. Considering TD3-based ADAC’s superiority over DDPG-based ADAC, we only combine VIME into TD3 and TD3-based ADAC. Note when paired with ADAC, intrinsic rewards are only visible to the behavior policy.

**Empirical Insights** Among the four environments, PendulumSparse has the most vulnerable environment-defined optimality. The goal here is to swing the inverted pendulum up so it stays upright. As suggested by Figure 3, not knowing how to distinguish between intrinsic and environment rewards, VIME-augmented TD3 is completely fooled into chasing after the intrinsic rewards. In other words, the VIME-augmented TD3’s understanding of what is optimal is completely off from the true environment-defined optimality. Note as demonstrated in left-bottom part of Figure 4, VIME-augmented TD3’s performance even trails behind TD3’s, which is an indisputable evidence that the bias introduced by IM can be detrimental and should be addressed whenever possible. In contrast, thanks to its disentangled nature, VIME-augmented ADAC only perceives intrinsic rewards in its behavior policy, which means its target policy always remains optimal with regards to our current knowledge about environment rewards. Because of this, VIME-augmented manages to consistently solve this exploration-challenging task. ADAC’s natural compatibility with VIME is further corroborated by the results in the remaining 3 games. As suggested by the complete Figure 4, VIME-augmented ADAC consistently surpasses all baselines by a large margin in terms of both convergence speed and final performance.

## References

- [Arulkumaran et al. 2017] Arulkumaran, K.; Deisenroth, M. P.; Brundage, M.; and Bharath, A. A. 2017. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine* 34(6):26–38.
- [Bellemare et al. 2016] Bellemare, M.; Srinivasan, S.; Ostrovski, G.; Schaul, T.; Saxton, D.; and Munos, R. 2016. Unifying count-based exploration and intrinsic motivation. In *NeurIPS*, 1471–1479.
- [Bellman 1966] Bellman, R. 1966. Dynamic programming. *Science* 153(3731):34–37.
- [Beyer et al. 2019] Beyer, L.; Vincent, D.; Teboul, O.; Gelly, S.; Geist, M.; and Pietquin, O. 2019. Mulex: Disentangling exploitation from exploration in deep rl. *arXiv preprint arXiv:1907.00868*.
- [Brockman et al. 2016] Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. Openai gym.
- [Colas, Sigaud, and Oudeyer 2018] Colas, C.; Sigaud, O.; and Oudeyer, P.-Y. 2018. Gep-pg: Decoupling exploration and exploitation in deep reinforcement learning algorithms. *arXiv preprint arXiv:1802.05054*.
- [Feng, Wang, and Liu 2017] Feng, Y.; Wang, D.; and Liu, Q. 2017. Learning to draw samples with amortized stein variational gradient descent. *arXiv preprint arXiv:1707.06626*.
- [Fu, Co-Reyes, and Levine 2017] Fu, J.; Co-Reyes, J.; and Levine, S. 2017. Ex2: Exploration with exemplar models for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, 2577–2587.
- [Fujimoto, Hoof, and Meger 2018] Fujimoto, S.; Hoof, H.; and Meger, D. 2018. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*.
- [Haarnoja et al. 2017] Haarnoja, T.; Tang, H.; Abbeel, P.; and Levine, S. 2017. Reinforcement learning with deep energy-based policies. In *ICML*.
- [Haarnoja et al. 2018a] Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018a. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *ICML*.
- [Haarnoja et al. 2018b] Haarnoja, T.; Zhou, A.; Hartikainen, K.; Tucker, G.; Ha, S.; Tan, J.; Kumar, V.; Zhu, H.; Gupta, A.; Abbeel, P.; et al. 2018b. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*.
- [Houthoof et al. 2016] Houthoof, R.; Chen, X.; Duan, Y.; Schulman, J.; De Turck, F.; and Abbeel, P. 2016. Vime: Variational information maximizing exploration. In *NeurIPS*, 1109–1117.
- [Lillicrap et al. 2015] Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning. In *Proceedings of the International Conference on Learning Representations*.
- [Liu and Wang 2016] Liu, Q., and Wang, D. 2016. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *NeurIPS*, 2378–2386.
- [Mnih et al. 2013] Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.
- [Mnih et al. 2015] Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; Petersen, S.; Beattie, C.; Sadik, A.; Antonoglou, I.; King, H.; Kumaran, D.; Wierstra, D.; Legg, S.; and Hassabis, D. 2015. Human-level Control through Deep Reinforcement Learning. *Nature* 518(7540):529–533.
- [Munos 2007] Munos, R. 2007. Performance bounds in  $l_p$ -norm for approximate value iteration. *SIAM journal on control and optimization* 46(2):541–561.
- [Péré et al. 2018] Péré, A.; Forestier, S.; Sigaud, O.; and Oudeyer, P.-Y. 2018. Unsupervised learning of goal spaces for intrinsically motivated goal exploration. *arXiv preprint arXiv:1803.00781*.
- [Plappert et al. 2018] Plappert, M.; Houthoof, R.; Dhariwal, P.; Sidor, S.; Chen, R. Y.; Chen, X.; Asfour, T.; Abbeel, P.; and Andrychowicz, M. 2018. Parameter space noise for exploration. In *ICLR*.
- [Schulman et al. 2017] Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- [Schulman, Chen, and Abbeel 2017] Schulman, J.; Chen, X.; and Abbeel, P. 2017. Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*.
- [Silver et al. 2014] Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; and Riedmiller, M. 2014. Deterministic policy gradient algorithms. In *International Conference on Machine Learning*.
- [Silver et al. 2017] Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; Chen, Y.; Lillicrap, T.; Hui, F.; Sifre, L.; van den Driessche, G.; Graepel, T.; and Hassabis, D. 2017. Mastering the game of go without human knowledge. *Nature* 550:354 EP –.
- [Sutton and Barto 1998] Sutton, R. S., and Barto, A. G. 1998. *Reinforcement Learning: An Introduction*. MIT Press.
- [Sutton and Barto 2018] Sutton, R. S., and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.
- [Sutton, Szepesvári, and Maei 2008] Sutton, R. S.; Szepesvári, C.; and Maei, H. R. 2008. A convergent  $o(n)$  algorithm for off-policy temporal-difference learning with linear function approximation. *Advances in neural information processing systems* 21(21):1609–1616.
- [Uhlenbeck and Ornstein 1930] Uhlenbeck, G. E., and Ornstein, L. S. 1930. On the theory of the brownian motion. *Phys. Rev.* 36:823–841.
- [Watkins and Dayan 1992] Watkins, C. J. C. H., and Dayan, P. 1992. Technical Note: Q-Learning. *Machine Learning* 8(3-4).
- [Ziebart 2010] Ziebart, B. D. 2010. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Ph.D. Dissertation, figshare.