Off-Policy Deep Reinforcement Learning with Analogous Disentangled Exploration

Paper 1662

ABSTRACT

Reinforcement learning (RL) is concerned with learning a rewarding policy by executing another policy that gathers sufficient samples. While the former policy is rewarding but in-expressive (in most cases, deterministic), doing well in the latter task in contrast requires an expressive policy that offers guided and effective exploration. Contrary to most methods that make a trade-off between optimality and expressiveness, disentangled frameworks explicitly decouple the two objectives, which are dealt with by two separate policies. Although being able to freely design and optimize both policies with respect to their own objectives, naively disentangling the two 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 critics and explicitly bind their corresponding actors. We empirically evaluate ADAC on 14 continuous-control tasks 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 tasks.

KEYWORDS

Reinforcement Leaning; Deep Reinforcement Learning; Exploration

1 INTRODUCTION

Reinforcement learning (RL) studies the control problem where an agent tries to navigate through an unknown environment [31]. The agent attempts to maximize its cumulative rewards through an iterative trial-and-error learning process [1]. Recently, we have seen many successes of applying RL to challenging simulation [18, 22] and real-world problems [17, 30, 35]. 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 [14, 21].

In order to explore novel states, it is crucial to incorporate randomness when interacting with the environment. Thanks to its simplicity, injecting noise into action space [11, 19] or parameter space [9, 25] 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) [14] and Soft-Q Learning (SQL) [32] learn to capture potentially rewarding actions through its energy objective. A second line of work considers using external guidance to aid exploration. In a nutshell, they formulate some intuitive tendencies in exploration as an additional reward function called intrinsic reward [2, 16]. 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 fully *exploit* the collected samples and turn them into the highest performing policy, as their learned policy often executes sub-optimal actions. To avoid this undesirable trade-off between *exploration* and *exploitation*, several attempts have been made to separately design two policies (i.e. disentangle them), of which one aims to gather the most informative examples (and hence is commonly referred as the behavior policy) while the other keeps itself optimal with respect to the knowledge from the gathered samples (and hence is usually referred as the target policy) [4, 6]. To help each fulfill their respective aim, disentangled objectives are further 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 tend to differ substantially 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: (i) the behavior policy and target policy are represented by the same neural network, (ii) and their corresponding critics are both calculated with regards to the target policy, yet potentially with different reward functions.

The rest of the paper is organized as follows. Section 2 reviews and summarizes the related work. RL as well as key background concepts and notations are introduced in Section 3. Section 4 introduces our proposed method Analogous Disentangled Actor-Critic (ADAC). Experiment details of ADAC are illustrated in Section 5. Finally, conclusions are made in Section 6.

2 RELATED WORK

Learning to be aware of potentially rewarding actions is a promising strategy to conduct exploration, as it automatically omits less rewarding actions and concentrates exploration effort on those with high potential. To capture these actions, expressive learning models/objectives are widely used. Most noticeable recent work on this direction, for example Soft Actor-Critic [14], EntRL [27],

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 2020, Auckland, New Zealand

https://doi.org/doi

and Soft Q Learning [13], learns an expressive energy-based target policy according to the maximum entropy reinforcement learning objective [39]. However, the expressiveness of their policies in turn becomes a burden for their optimality, and in practice, trade-offs such as temperature controlling [15] and reward scaling [13] have to be made for better overall performance. As we shall show later, ADAC makes use of a similar but extended energy-based target, and mitigates its optimality problem using the disentangled framework.

Ad-hoc exploration-oriented learning targets that are designed to better explore the state space are also promising. Some recent research efforts on this line include count-based exploration [2, 38] and intrinsic motivation [10, 16] approaches. The outcome of these methods is usually a reward function termed the *intrinsic reward*, which is extremely useful when the environment-defined reward is sparsely available. However, as we shall illustrate in Section 5.3, intrinsic reward potentially biases the task-defined learning objective, leading to catastrophic failure in some tasks. Again, with the disentangled nature of ADAC, we give a principled solution to solve this problem with theoretical guarantees (Section 4.3).

Explicitly separating exploration from exploitation has been used to solve a common problem in the above approaches, which is sacrificing the target policy's optimality for better exploration. By separately designing exploration and exploitation components, both objectives can be better pursued simultaneously. Specifically, GEP-PG [6] uses a Goal Exploration Process (GEP) [8] to generate samples and feed them to the replay buffer of DDPG or its variants. MULEX [4] proposes to use a series of intrinsic rewards to optimize different policies in parallel, which in turn generates abundant samples for training the target policy. Despite their intriguing conceptual idea, they consider the two processes independently without considering their relations. On the other hand, not using the disentanglement idea, Batch-Constrained deep Q-learning (BCQ) [12] proposes to restrict the action space to force the agent towards acting closely around the behavior policy, which stabilizes the training process as well as improves the performance. However, BCQ assumes the behavior policy is unknown, and adapts the target policy to it, which non-negligibly biases the update process of the target policy.

3 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.

3.1 RL with Continuous Control

In a standard *reinforcement learning* (RL) setup, an agent interacts with an unknown environment at discrete time steps and aims to maximize the "reward" signal [31]. The environment is commonly formalized as a *Markov Decision Process* (MDP), which can be succinctly defined as a 5-tuple $\mathcal{M} =< \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma >$. At time step t, the agent in state $s_t \in \mathcal{S}$ takes action $a_t \in \mathcal{A}$ according to *policy* π, a conditional distribution of a given s, leading to the next state s_{t+1} according to the transition probability $\mathcal{P}(s_{t+1} \mid s_t, a_t)$. Meanwhile, the agent observes reward $r_t \sim \mathcal{R}(s_t, a_t)$ emitted from the environment. In all the environments considered in this paper, actions are assumed to be continuous.

The agent strives to learn the *optimal policy* that maximizes the expected return $J(\pi) = \mathbb{E}_{s_0\sim\rho_0, a_t\sim\pi, s_{t+1}\sim\mathcal{P}, r_t\sim\mathcal{R}}\left[\sum_{t=0}^{\infty}\gamma^t r_t\right]$, where ρ_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 policy π (also known as the actor in the actor-critic framework) parameterized by θ can be updated by taking the gradient $\nabla_\theta J(\pi)$. According to the deterministic policy gradient theorem [29], $\nabla_\theta J(\pi) = \mathbb{E}_{(s,a)\sim\rho_\pi}\left[\nabla_a Q^\pi_\mathcal{R}(s,a)\nabla_\theta\pi(s)\right]$, where ρ_π denotes the state-action marginals of the trajectory distribution induced by π, and $Q^\pi_\mathcal{R}$ denotes the state-action value function (also know 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 π afterwards. Intuitively, it measures how preferable executing action a is at state s with respect to the policy π and reward function $\mathcal{R}$. Following [3], we additionally introduce the Bellman operator, which is commonly used to update the Q-function. The Bellman operator $\mathcal{T}^\pi_\mathcal{R}$ uses $\mathcal{R}$ and π to update an arbitrary value function Q, which is not necessarily defined with respect to the same π or $\mathcal{R}$. For example, the outcome of $\mathcal{T}^{\pi_1}_{\mathcal{R}_1} Q^{\pi_2}_{\mathcal{R}_2}(s_t, a_t)$ is defined as $\mathcal{R}_1(s_t, a_t) + \gamma\mathbb{E}_{s_{t+1}\sim\mathcal{P}, a_{t+1}\sim\pi_1}[Q^{\pi_2}_{\mathcal{R}_2}(s_{t+1}, a_{t+1})]$. By slightly abusing notations, we further define the outcome of $\mathcal{T}^{max}_{\mathcal{R}_1} Q^{\pi_2}_{\mathcal{R}_2}(s_t, a_t)$ as $\mathcal{R}_1(s_t, a_t) + \gamma \max_{a_{t+1}} \mathbb{E}_{s_{t+1}\sim\mathcal{P}}[Q^{\pi_2}_{\mathcal{R}_2}(s_{t+1}, a_{t+1})]$. Some also call $\mathcal{T}^{max}_\mathcal{R}$ the Bellman optimality operator.

3.2 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 actor-critic methods [36] allow us to make a clear separation between the *target policy*, which refers to the best policy currently learned by the agent, and the *behavior policy*, which the agent follows to interact with the environment. Note the discussion in the earlier subsection is largely around the target policy. Thus, starting from this point, to avoid confusion, π is reserved to only denote the target policy and notation μ is dedicated to denote the behavior policy. Due to the policy separation, the target policy is instead resorting to the estimates calculated with regards to samples collected by the behavior policy, that is, the deterministic policy gradient mentioned above is approximated as

$$\nabla_\theta J_\pi(\theta) \approx \mathbb{E}_{(s,a)\sim\rho_\mu}\left[\nabla_a Q^\pi_\mathcal{R}(s,a)\nabla_\theta\pi_\theta(s)\right]. \tag{1}$$

where $\mathcal{R}$ is the environment-defined reward. One of the most notable off-policy learning algorithms that capitalize on this idea is deep deterministic policy gradient (DDPG) [19]. To mitigate function approximation errors in DDPG, [11] proposes TD3. Given that DDPG and TD3 have demonstrated themselves to be competitive in many continuous control benchmarks, we choose to implement our *Analogous Disentangled Actor Critic* (ADAC) on top of their target policies. Yet, it is worth reiterating that ADAC is compatible with any existing off-policy learning algorithms. We defer a more detailed discussion of ADAC's compatibility until we start formally introducing our method in Section 4.1.

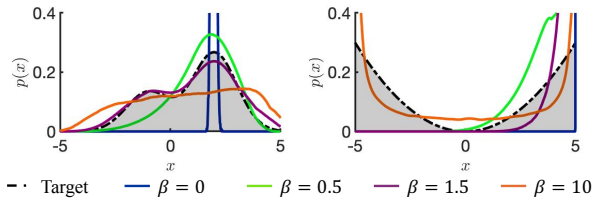

Figure 1: Evaluation of the SVGD learning algorithm (Eq 3) with different β under two different distributions.

3.3 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 μ to capture information about potentially rewarding actions. Energy-based representations have recently been increasingly chosen as the target form to construct an expressive behavior policy. Since its first introduction by [39] to achieve maximum-entropy reinforcement learning, several additional prior works keep improving upon this idea. Among them, the most notable ones include Soft Q-Learning (SQL) [13], EntRL [27], and Soft Actor-Critic (SAC) [15]. Collectively, they have achieved competitive results on many benchmark tasks. Formally, the energy-based behavior policy is defined as

$$\mu(a \mid s) \propto \exp\left(Q(s,a)\right), \tag{2}$$

where Q is commonly selected to be the target critic $Q^{\pi}_{\mathcal{R}}$ in prior work (i.e., [14, 15]). Various efficient samplers have been proposed to approximate the distribution specified in Eq 2 [13, 14]. Among them, [13]'s Stein Variational Gradient Descent (SVGD) [20] 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 target distribution. In the context of RL, following Amortized SVGD [7], 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 [7], f_{φ} is updated according to the following gradient:

$$\nabla_{\varphi} J_{\mu}(\varphi) \approx \mathbb{E}_{s,\xi\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\Big[\sum_{j=1}^{K}\Big[\underbrace{\mathcal{K}(a,a'_j)\nabla_{a'_j}Q(s,a'_j)}_{\text{term 1}} + \beta\cdot\underbrace{\nabla_{a'_j}\mathcal{K}(a,a'_j)}_{\text{term 2}}\Big]\Big|_{a=f_{\varphi}(s,\xi)}\frac{\partial f_{\varphi}(s,\xi)}{\partial\varphi}\Big]/K, \tag{3}$$

where $\mathcal{K}$ is a positive definite kernel[1], and β is an additional hyperparameter proposed to make optimality-expressiveness tradeoff.

[1]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.

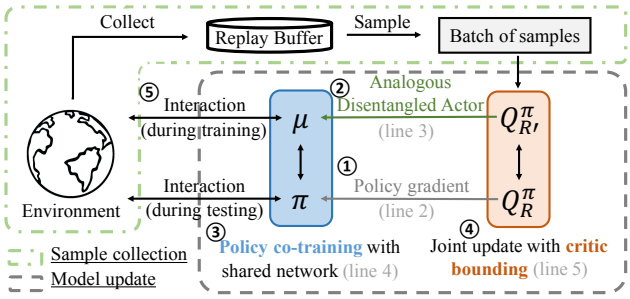

Figure 2: Block diagram of ADAC, which consists of the sample collection phase (green box with dotted line) and the model update phase (dashed gray box). Model updates are performed sequentially from ① to ④. Each update step's corresponding line in Algorithm 1 is shown in brackets.

Algorithm 1 Training procedure of the Analogous Disentangled Actor-Critic. Correspondence with Figure 2 is given after "//".

1: **Input:** A minibatch of samples $\mathcal{B}$, actor model f_{φ} (represents the target policy f^{π}_{φ} as well as the behavior policy f^{μ}_{φ}), critic models $Q^{\pi}_{\mathcal{R}}$ and $Q^{\pi}_{\mathcal{R}'}$.
2: $\nabla_{\varphi} f^{\pi}_{\varphi} \leftarrow$ the deterministic policy gradient of $Q^{\pi}_{\mathcal{R}}$ with respect to π (Eq 1). // **target policy update**
3: $\nabla_{\varphi} f^{\mu}_{\varphi} \leftarrow$ gradient of $Q^{\pi}_{\mathcal{R}'}$ with respect to the behavior policy μ (Eq 3, Section 3.3) // **behavior policy learning**
4: Update f with $\nabla_{\varphi} f^{\pi}_{\varphi}$ and $\nabla_{\varphi} f^{\mu}_{\varphi}$ // **policy co-training**
5: Update Q^{π}_{ϕ} and Q^{μ}_{ψ} to minimize the mean squared error on $\mathcal{B}$ with respect to the target $\mathcal{T}^{\pi}Q^{\pi}_{\mathcal{R}}$ and $\mathcal{T}^{\pi}Q^{\pi}_{\mathcal{R}'}$, respectively. // **value update with critic bounding**

The intrinsic connection between Eq 3 and the deterministic policy gradient (Eq 1) is introduced in [13] and [7]: 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.

To aid a better understand of this relation, we illustrate the distribution approximated by SVGD using different β in a toy example as shown in Figure 1. The dashed line is the approximation target. When β is small, the entropy of the learned distribution is restricted and the overall policy leans towards the optimal. On the other hand, larger β leads to more expressive approximation.

4 ANALOGOUS DISENTANGLED ACTOR CRITIC

This section introduces our proposed method *Analogous Disentangled Actor-Critic* (ADAC). We start with providing an overview of it, followed by elaborating the specific choices we make to design our actors and critics.

4.1 Algorithm Overview

Figure 2 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 $\langle \mu, Q^{\pi}_{\mathcal{R}'} \rangle$ and $\langle \pi, Q^{\pi}_{\mathcal{R}} \rangle$. Mainly two places manifest the analogous property of our method. First, both actors/policies (μ and π) are represented by the same neural network f, where $\mu(s) := f_\varphi(s, \xi)|_{\xi \sim \mathcal{N}(\mathbf{0}, \mathbf{I})}$ and $\pi(s) := f_\varphi(s, \xi)|_{\xi = [0, 0, \ldots, 0]^T}$. Here we highlight the key reasons for this particular design to give intuitions, while its details are deferred in Section 4.2. The choice of ξ acts as a restriction of the "distance" between π and μ, which stabilizes the policy learning process. Furthermore, the gradients of π and μ have a key connection that motivates this design and improves the expressiveness of μ under certain conditions.

The second exhibit of our method's analogous nature lies on our designed critics $Q^{\pi}_{\mathcal{R}}$ and $Q^{\pi}_{\mathcal{R}'}$, which are based on reward function $\mathcal{R}$ and $\mathcal{R}'$ respectively yet are both computed with regard to the target policy π. As a standard approach, $Q^{\pi}_{\mathcal{R}}$ approximates the task-defined objective that the algorithm aims to maximize. On the other hand, $Q^{\pi}_{\mathcal{R}'}$ is a behavior critic that can be shown to be both *explorative* and *stable* theoretically (Section 4.3) and empirically (Section 5.3). 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. π and μ 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 2). As defined above, π 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 $Q^{\pi}_{\mathcal{R}}$ is used as the referred critic (① in Figure 2). On the other hand, for the sake of expressiveness, the energy-based objective (Eq 2) is a good fit for μ. To further encourage exploration, we use $Q^{\pi}_{\mathcal{R}'}$ in the objective, which gives $\mu(a \mid s) \propto \exp(Q^{\pi}_{\mathcal{R}'}(s, a))$ (② in Figure 2). Since both policies share the same network f, critic optimization is done by maximizing

$$J_\pi(\varphi) + J_\mu(\varphi), \tag{4}$$

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

In the sample collection phase, μ interacts with the environment and the gathered samples are stored in a replay buffer [21] for later use in the model update phase. Given state s, actions are sampled from μ with a two-step procedure: (i) sample $\xi \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and then plug the sampled ξ 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 π (which is also their target policy) and critic $Q^{\pi}_{\mathcal{R}}$. To be more specific, ADAC merely adds a new actor μ to interact with the environment and a new critic $Q^{\pi}_{\mathcal{R}'}$ that guides μ's updates on top of the base model, along with the constraints/connections enforced between the inherented 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 [19] and TD3 [11]) are adopted.

4.2 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 π's learning process: being expressive also means that μ is often significantly different from π, leading to collect substantially biased samples, which in turn render the learning process of $Q^{\pi}_{\mathcal{R}}$ unstable and vulnerable to catastrophic failure [26, 33, 37]. To be more specific, since the difference between π and an expressive μ is more than some zero-mean random noise, the state marginal distribution ρ_μ defined with respect to μ can potentially diverge greatly from the state marginal distribution ρ_π defined with respect to π. Since ρ_π is not directly accessible, as shown in Equation 1, the gradients of π are approximated using samples from ρ_μ. When the approximated gradients constantly deviate significantly from the true values (i.e. the approximated gradients are biased), the updates to π 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 μ and π, which naturally reduces the KL-divergence between distribution ρ_μ and ρ_π. One straightforward approach to reduce the distance between the two policies is to reduce the randomness of μ, for example by lowering β. However, this inevitably sacrifices μ's expressiveness, which in turn would also harm ADAC's competitiveness. Alternatively, we propose *policy co-training* to best maintain the expressiveness of μ while also stabilizing it by restricting it with regards to π, which is motivated by the intrinsic connection between Eqs 1 and 3 (Section 3.3). As described in Section 4.1, 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 ξ. During training, f is updated to maximize Eq 4. The method to sample actions from μ is described in the $4th$ paragraph of Section 4.1.

We further justify the above choice by demonstrating that the imposed restrictions on μ and π only has minor influence on π's optimality and μ's expressiveness. To argue for this point, we need to revisit Equation 3 for one more time: π can be viewed as being updated with $\beta = 0$, whereas μ is updated with $\beta > 0$. Intuitively, this makes policy π optimal since its action is not affected by the entropy term (the second term). μ is still expressive since only when the input random variable ξ is close to the zero vector, it will be significantly restricted by π. In Section 5.1, we will empirically demonstrate policy co-training indeed reduces the distance between μ and π during training, fulfilling its mission.

Additionally, *policy co-training* enforces the underlying relations between π and μ. Specifically, policy co-training forces π to be contained in μ since $[0, 0, \ldots, 0]^T$ is the highest-density point of

$\mathcal{N}(\mathbf{0}, \mathbf{I})$, and sampling ξ from $\mathcal{N}(\mathbf{0}, \mathbf{I})$ is likely to generate actions close to that from π. This matches the intuition that π and μ should share similarities: actions proposed by π is rewarding (with respect to $\mathcal{R}$) and thus should be frequently executed by μ.

4.3 Incorporating Intrinsic Reward in Behavior Critic via Critic Bounding

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

As introduced in Section 4.1, $\mathcal{R}$ 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 $Q^{\pi}_{\mathcal{R}}$ and behavior critic $Q^{\pi}_{\mathcal{R}'}$ are defined with regard to the same policy but updated differently according to the following

$$Q^{\pi}_{\mathcal{R}} \leftarrow \mathcal{T}^{\pi}_{\mathcal{R}} Q^{\pi}_{\mathcal{R}}; \quad Q^{\pi}_{\mathcal{R}'} \leftarrow \mathcal{T}^{\pi}_{\mathcal{R}'} Q^{\pi}_{\mathcal{R}'}, \tag{5}$$

Updates are performed through minibatches in practice. Note that when no intrinsic reward is used, Eq 5 becomes trivial and the two critics ($Q^{\pi}_{\mathcal{R}}$ and $Q^{\pi}_{\mathcal{R}'}$) 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 μ is updated following the guidance (i.e., the energy-based objective) of $Q^{\pi}_{\mathcal{R}'}$, we do so to prevent μ from diverging disastrously from π. The following theorem provides rigorous justifications.

Theorem 4.1. *Let π be a greedy policy w.r.t. $Q^{\pi}_{\mathcal{R}}$ and μ be a greedy policy w.r.t. $Q^{\pi}_{\mathcal{R}'}$. Assume $Q^{\pi}_{\mathcal{R}'}$ is optimal w.r.t. $\mathcal{T}^{\pi}_{\mathcal{R}'}$ and $\mathcal{R}' \geq \mathcal{R}$ in all states, We have the following results. First, $\mathbb{E}_{\rho_\pi}[\mathcal{T}^{max}_{\mathcal{R}} Q^{\pi}_{\mathcal{R}} - Q^{\pi}_{\mathcal{R}}]$, a proxy of training stability, is lower bounded by $\mathbb{E}_{\rho_\mu}[\mathcal{T}^{max}_{\mathcal{R}} Q^{\pi}_{\mathcal{R}} - Q^{\pi}_{\mathcal{R}}] + \mathbb{E}_{\rho_\pi}[\mathcal{R}] - \mathbb{E}_{\rho_\mu}[\mathcal{R}]$. Second, $\mathbb{E}_{\rho_\mu}[\mathcal{T}^{max}_{\mathcal{R}} Q^{\pi}_{\mathcal{R}} - Q^{\pi}_{\mathcal{R}}]$, a proxy of training effectiveness, is lower bounded by $\mathbb{E}_{\rho_\pi}[\mathcal{T}^{max}_{\mathcal{R}} Q^{\pi}_{\mathcal{R}} - Q^{\pi}_{\mathcal{R}}] + \mathbb{E}_{\rho_\pi}[\mathcal{R} - \mathcal{R}']$.*

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 4.1. We first examine the assumptions made by the theorem. While other assumptions are generally satisfiable and are commonly made in the RL literature [23], 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., [10, 16]), 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 3.1), $\mathcal{T}^{max}_{\mathcal{R}} Q^{\pi}_{\mathcal{R}} - Q^{\pi}_{\mathcal{R}}$ quantifies the improvement on Q after performing one value iteration [3] 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, $\mathbb{E}_{\rho}[\mathcal{T}^{max}_{\mathcal{R}} Q^{\pi}_{\mathcal{R}} - Q^{\pi}_{\mathcal{R}}]$ is the expected policy improvement under state-action distribution ρ in the near future.

We formalize training stability as the lower bound of the expected policy improvement under ρ_π (i.e., $\mathcal{T}^{max}_{\mathcal{R}} Q^{\pi}_{\mathcal{R}} - Q^{\pi}_{\mathcal{R}}$). Theorem 4.1 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 π is optimized to maximize the cumulative reward of $\mathcal{R}$ while μ is not. On the other hand, the first term can be viewed as the improvement of $Q^{\pi}_{\mathcal{R}}$ during training since ρ_μ is the training sample distribution of ADAC. Therefore, the improvement of $Q^{\pi}_{\mathcal{R}}$ during training lower bounds the expected policy improvement under ρ_π, which represents stability.

Conversely, the lower bound on $\mathbb{E}_{\rho_\pi}[\mathcal{T}^{max}_{\mathcal{R}} Q^{\pi}_{\mathcal{R}} - Q^{\pi}_{\mathcal{R}}]$ 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 π would visit states that are frequently visited by μ, 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 ρ_π and ρ_μ are not identical, as long as μ allows substantial visitations of high probability states in ρ_π to make $\mathbb{E}_{\rho_\pi}[\mathcal{R}^{in}]$ sufficiently small, improvement when trained on the samples ρ_μ would be almost as large as the training improvement on the target distribution ρ_π.

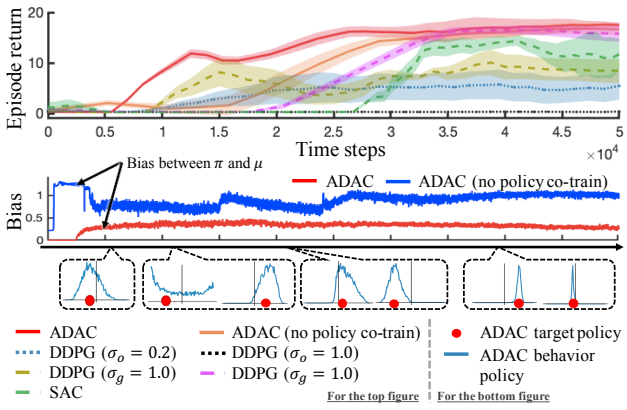

Figure 3: Learning curves of ADAC (with base model DDPG) and baselines on the modified CartPole environment. In addition, ADAC's target (red dots) and behavior policies (solid blue curves) at different timesteps are plotted below the learning curves.

5 EXPERIMENTS

In this section, we take gradual steps to analyze and illustrate our proposed method ADAC. Specifically, We first investigate the behavior of our analogous disentangled behavior policy (Section 5.1). Next, we perform an empirical evaluation of ADAC without intrinsic rewards on 14 standard continuous-control benchmarks (Section 5.2). 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 5.3). Throughout this paper, we highlight two benefits from the analogous disentangled nature of ADAC: (i) avoiding unnecessary trade-offs between current optimality and exploration (i.e. a more expressive and effective behavior policy); (ii) natural compatibility with intrinsic rewards without altering environment-defined optimality. In this context, the first two subsections are devoted to demonstrating the first benefit and the last subsection is dedicated for the second.

Table 1: Specifications of our action and reward designs for the modified CartPole task. The original task consists of two discrete actions *left* and *right*, each pushing the cart towards its corresponding direction. We converted them into a single-dimension continuous action.

Action ($a \in [-1, 1]$)	Reward ($r \in \mathbb{R}$)
$a = \begin{cases} \text{left} & a < -0.5 \\ p(\text{left}) = p(\text{right}) = 0.5 & a \in [0.5, 0.5] \\ \text{right} & a > 0.5 \end{cases}$	$r = -0.1\lvert a\rvert - 0.05a^2 + \begin{cases} -1.0 & \text{episode ended} \\ 0.1 & \text{otherwise} \end{cases}$

Table 2: Continuous-control performance in 14 benchmark environments. Average episode return (± standard deviation) over 20 trials are reported. Bold indicates the best average episode return. † 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±373	838.1*±97.1	**2903**†±666	450.0±27.9	**2726**±652	1280±71
RoboschoolHopper	**2299**†±333	766.5*±10	**2302**†±537	543.8±307	2089±657	1229±345
RoboschoolHalfCheetah	1578†±166	**1711***±95	607.2±246.2	441.6±120.4	807.0±252.6	1225±184.2
RoboschoolAtlasForwardWalk	**234.6**†±55.7	186.7*±37.9	190.6±50.1	52.63±26.2	126.0±47.1	107.6±29.4
RoboschoolWalker2d	**1769**†±452	**1564***±651	995.1±146.3	208.7±137.1	1021±263	578.9±231.3
Ant	3353±847	1226*±18	4034†±517	370.5±223	**4291**±1498	1401±168
Hopper	**3598**†± 374	374.5*±36.5	2845±609	38.93±0.88	**3307**±825	1555±458
HalfCheetah	9392±199	2238*±40	10526†±2367	1009±49	**11541**±2989	881.7±10.1
Walker2d	**5122**†±1314	1291*±42	4630†±778	186.2±33.3	4067±1211	1146±368
InvertedPendulum	**1000**†±0	**1000***±0	**1000**†±0	**1000***±0	**1000**±0	98.90±2.08
InvertedDoublePendulum	**9359**†±0.17	9334*±1.39	7665±566	27.20±2.61	9353±2896	98.90±5.88
BipedalWalker	**309.8**†±15.6	-52.77*±1.94	288.4†±51.25	-123.90±11.17	**307.2**±57.92	266.9±28.52
BipedalWalkerHardcore	**-10.76**†±27.70	-98.52±3.21	-57.97±21.08	-50.05*±10.27	-127.4±45.2	-105.3±22.2
LunarLanderContinuous	**290.0**†±50.9	85.67*±23.42	**289.7**†±54.1	-65.89±96.48	283.3±69.29	59.32±68.44

5.1 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 restrains 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 restrains, we examine the policy network's stability (i.e., does the *policy co-training* lowers the bias between two policies and stabilize the π's learning process). Finally, we focus on the effectiveness of our behavior policy by measuring the overall performance of ADAC (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 environment, namely CartPole, as our demonstration bed. The goal in this environment 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 as specified in Table 1. 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 (σ_g) and Ornstein-Uhlenbeck process noise (σ_o) [34], 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 To minimize distraction, our discussion starts with closely examining ADAC's behavior and target alone. First, see the cells at the bottom of Figure 3, 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 (μ) 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 (μ) 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 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*[2]. The effect of critic bounding is measured by the *bias* between π and μ, which is shown in the middle of Figure 3. 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 μ, which is suggested by the behavior policies rendered below in Figure 3.

Finally we move our attention to the learning curves in Figure 3: 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, adversely affecting its policy improvement process. In other words, different from ADAC, SAC cannot fully exploits its learned knowledge of the environment (i.e. its value functions) to construct its target policy, leading to a performance inferior to ADAC's.

5.2 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 achieve state-of-the-art performance 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 [5] On top of the two baselines adopted earlier (i.e. DDPG and SAC), we further include TD3 [11], which improves upon DDPG by addressing some of its function approximation errors, PPO [28], which is regarded as one of the most stable and efficient on-policy policy gradient algorithm, and GEP-PG [6], which combines Goal Exploration Process [24] with policy gradient to perform curious exploration as well as stable learning. Though not exhaustive, this baseline suite still embodies many of the latest advancements and can be indeed deemed as the existing state-of-the-art. However, we compare with GEP-PG only in tasks adopted in their original experiments. Additionally, since the GEP part of the algorithm needs hand-crafted exploration goals, we are not able to run their model on new experiments since it is nontrivial to generalize their experiments in other tasks. 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.[3]

Empirical Insights Table 2 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. Though not as remarkable as DDPG-based ADAC, TD-based ADAC also manages to achieve statically better or comparable performance over its base model on more than 78%(11/14) of the tasks; see the † 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.

We would like to specially 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 do not compare with GEP-PG [6] across 14 benchmarks and hence GEP-PG is not included in Table 2 because the Goal Exploration Process (GEP) in GEP-PG requires manually defining a goal space to explore, which is task dependent and critically influences the algorithm performance. Therefore, we only compare our results on the two experiments that they have run, of which only one overlaps with our task suit, namely HalfCheetah. In HalfCheetah, GEP-PG achieves 6118 cumulative reward, while ADAC (TD3) achieves 9392, showing superiority over GEP-PG. Furthermore 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.

5.3 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. In

[2]Two separate neural networks are used to store π and μ when policy co-training is not used.

[3]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.

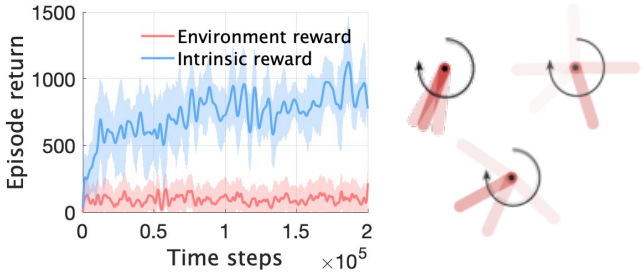

Figure 4: Illustration of how intrinsic reward contaminates the environment-defined optimality in PendulumSparse. Fooled into collecting more intrinsic rewards rather than environment rewards (see the learning curves on the left), the agent constantly alternates between spinning the pendulum and barely moving it (see the snapshots of the target policy on the right), making no real progress.

this regard, we are particularly interested in investigating whether the disentangled nature of ADAC helps mitigate intrinsic rewards' undesirable bias effect on the environment-defined optimality.

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 countinuous), to construct a decently sized testing suite, we further hand-craft new tasks, namely PendulumSparse and CartPoleSwingUpSparse by sparsifying the rewards in the existing environments. Sparsifying is achieved mainly through suppressing the original rewards until reaching some predefined threshold. Due to their dependency on environment-provided rewards as feedback signals, most model-free RL algorithms suffer significant performance degradation in these sparse-reward tasks. 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) [16] as our internal reward generator for its consistent good performance on a wide variety of exploration-challenging tasks. 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 4, 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 5, 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,

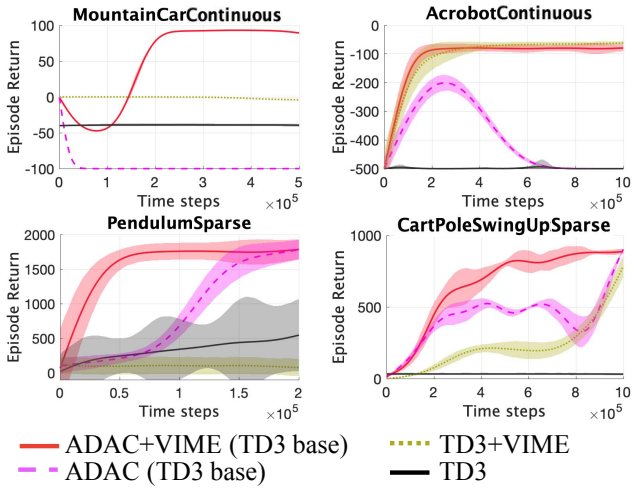

Figure 5: Learning curves for four sparse-reward tasks. Lines denote the average over 20 trials and the shaded areas represent the range of one standard deviation.

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 tasks. As suggested by the the complete Figure 5, VIME-augmented ADAC consistently surpasses all reported alternatives by a large margin in terms of both convergence speed and final performance.

6 CONCLUSION

We present Analogous Disentangled Actor-Critic (ADAC), an off-policy reinforcement learning framework that makes explicit disentanglement between the behavior and target policy. Compared to prior work, to stabilize model updates, we restrain our behavior policy and its corresponding critic to be analogous of their target counterparts. Thanks to its disentangled and analogous nature, ADAC achieves the state-of-the-art results in 10 out of 14 continuous control benchmarks. Moreover, ADAC is naturally compatible with intrinsic rewards, outperforming alternatives in exploration-challenging tasks.

APPENDIX

A Hyper-parameters

We made great efforts to make sure we have a fair comparison with baselines. To best retain their performance, we always adopted hyper-parameters reported in the respective papers and used their open-source code when available; if not, we used the most-stared third-party implementations in Github. Since ADAC is based on DDPG and TD3, we fixed the original hyper-parameters such as replay buffer size, and tuned parameters introduced by ADAC only, i.e., K, β, and learning rate. Specifically, β was annealed from 2.0 to 1.0 during training, and K was 32 for all tasks except Hopper and RoboschoolAtlasForwardWalk, where it was 8. ξ was set to be 16-dimensional. In ADAC (TD3), learning rate for the target policy and behavior policy is $1e-3$ and $3e-4$, respectively. For ADAC (DDPG), learning rate of both policies was set to $1e-4$.

REFERENCES

[1] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. 2017. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine* 34, 6 (2017), 26–38.

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

[3] Richard Bellman. 1966. Dynamic programming. *Science* 153, 3731 (1966), 34–37.

[4] Lucas Beyer, Damien Vincent, Olivier Teboul, Sylvain Gelly, Matthieu Geist, and Olivier Pietquin. 2019. MULEX: Disentangling Exploitation from Exploration in Deep RL. *arXiv preprint arXiv:1907.00868* (2019).

[5] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI Gym. (2016). arXiv:arXiv:1606.01540

[6] Cédric Colas, Olivier Sigaud, and Pierre-Yves Oudeyer. 2018. Gep-pg: Decoupling exploration and exploitation in deep reinforcement learning algorithms. *arXiv preprint arXiv:1802.05054* (2018).

[7] Yihao Feng, Dilin Wang, and Qiang Liu. 2017. Learning to draw samples with amortized stein variational gradient descent. *arXiv preprint arXiv:1707.06626* (2017).

[8] Sébastien Forestier, Yoan Mollard, and Pierre-Yves Oudeyer. 2017. Intrinsically motivated goal exploration processes with automatic curriculum learning. *arXiv preprint arXiv:1708.02190* (2017).

[9] Meire Fortunato, Mohammad Gheshlaghi Azar, Bilal Piot, Jacob Menick, Ian Osband, Alex Graves, Vlad Mnih, Remi Munos, Demis Hassabis, Olivier Pietquin, et al. 2017. Noisy networks for exploration. *Internation Conference on Learning Representations (ICLR)* (2017).

[10] Justin Fu, John Co-Reyes, and Sergey Levine. 2017. Ex2: Exploration with exemplar models for deep reinforcement learning. In *Advances in Neural Information Processing Systems*. 2577–2587.

[11] Scott Fujimoto, Herke Hoof, and David Meger. 2018. Addressing Function Approximation Error in Actor-Critic Methods. In *International Conference on Machine Learning*.

[12] Scott Fujimoto, David Meger, and Doina Precup. 2018. Off-policy deep reinforcement learning without exploration. *arXiv preprint arXiv:1812.02900* (2018).

[13] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. 2017. Reinforcement learning with deep energy-based policies. In *Proceedings of the International Conference on Machine Learning-Volume 70*.

[14] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*.

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

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

[17] Joel Z. Leibo, Vinicius Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. 2017. Multi-agent Reinforcement Learning in Sequential Social Dilemmas. In *Proc. of AAMAS*.

[18] Yitao Liang, Marlos C. Machado, Erik Talvitie, and Michael Bowling. 2016. State of the Art Control of Atari Games Using Shallow Reinforcement Learning. In *Proc. of AAMAS*.

[19] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. In *Proceedings of the International Conference on Learning Representations*.

[20] Qiang Liu and Dilin Wang. 2016. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances In Neural Information Processing Systems*. 2378–2386.

[21] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).

[22] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level Control through Deep Reinforcement Learning. *Nature* 518, 7540 (26 02 2015), 529–533.

[23] Rémi Munos. 2007. Performance bounds in l_p-norm for approximate value iteration. *SIAM journal on control and optimization* 46, 2 (2007), 541–561.

[24] Alexandre Péré, Sébastien Forestier, Olivier Sigaud, and Pierre-Yves Oudeyer. 2018. Unsupervised learning of goal spaces for intrinsically motivated goal exploration. *arXiv preprint arXiv:1803.00781* (2018).

[25] Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. 2018. Parameter space noise for exploration. In *Proceedings of the International Conference on Learning Representations*.

[26] Matthew Schlegel, Wesley Chung, Daniel Graves, Jian Qian, and Martha White. 2019. Importance Resampling for Off-policy Prediction. *arXiv preprint arXiv:1906.04328* (2019).

[27] John Schulman, Xi Chen, and Pieter Abbeel. 2017. Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440* (2017).

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

[29] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. 2014. Deterministic Policy Gradient Algorithms. In *International Conference on International Conference on Machine Learning*.

[30] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. 2017. Mastering the game of Go without human knowledge. *Nature* 550 (18 10 2017), 354 EP –.

[31] Richard S. Sutton and Andrew G. Barto. 1998. *Reinforcement Learning: An Introduction*. MIT Press.

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

[33] Richard S Sutton, Csaba Szepesvári, and Hamid Reza Maei. 2008. A convergent O (n) algorithm for off-policy temporal-difference learning with linear function approximation. *Advances in neural information processing systems* 21, 21 (2008), 1609–1616.

[34] G. E. Uhlenbeck and L. S. Ornstein. 1930. On the Theory of the Brownian Motion. *Phys. Rev.* 36 (1930), 823–841. Issue 5.

[35] Yue Wang and Fumin Zhang. 2017. *Trends in Control and Decision-Making for Human-Robot Collaboration Systems* (1st ed.). Springer Publishing Company, Incorporated.

[36] Christopher J. C. H. Watkins and Peter Dayan. 1992. Technical Note: Q-Learning. *Machine Learning* 8, 3-4 (May 1992).

[37] Tengyang Xie, Yifei Ma, and Yu-Xiang Wang. 2019. Optimal Off-Policy Evaluation for Reinforcement Learning with Marginalized Importance Sampling. *arXiv preprint arXiv:1906.03393* (2019).

[38] Zhi-Xiong Xu, Xi-Liang Chen, Lei Cao, and Chen-Xi Li. 2017. A study of count-based exploration and bonus for reinforcement learning. In *Proceedings of the IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*. IEEE, 425–429.

[39] Brian D Ziebart. 2010. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Ph.D. Dissertation. figshare.

Proof of Themrem 4.1

We define Q_*^π as the optimal value function with respect to policy π and reward $\mathcal{R}$, i.e., $Q_*^\pi = \mathcal{T}_\mathcal{R}^\pi Q_*^\pi$. We further define $Q_{\mathcal{R}'}^\mu$ as the optimal value function with respect to μ and $\mathcal{R}'$. Before delving into the detailed derivation, we make the following clarifications. First, although π is greedy policy w.r.t. $Q_\mathcal{R}^\pi$, $Q_\mathcal{R}^\pi$ is not the optimal value function w.r.t. π and $\mathcal{R}$. In other words, we have $Q_\mathcal{R}^\pi \neq \mathcal{T}_\mathcal{R}^\pi Q_\mathcal{R}^\pi$ and $\mathcal{T}_\mathcal{R}^{max} Q_\mathcal{R}^\pi = \mathcal{T}_\mathcal{R}^\pi Q_\mathcal{R}^\pi$. Second, in both the theorem and the proof, we omit the state-action notation (e.g., $Q(s,a)$) for notation simplicity.

We begin from the difference between the respective optimal value function with regard to $\mathcal{T}_{\mathcal{R}'}^\pi$ and $\mathcal{T}_\mathcal{R}^\pi$:

$$
\begin{aligned}
Q_{\mathcal{R}'}^\mu - Q_*^\pi &= \mathcal{T}_{\mathcal{R}'}^\mu Q_{\mathcal{R}'}^\mu - \mathcal{T}_{\mathcal{R}'}^\mu Q_\mathcal{R}^\pi + \mathcal{T}_\mathcal{R}^{max} Q_\mathcal{R}^\pi - \mathcal{T}_\mathcal{R}^\pi Q_*^\pi - (\mathcal{T}_{\mathcal{R}'}^\mu Q_{\mathcal{R}'}^\mu - Q_{\mathcal{R}'}^\mu + \mathcal{T}_\mathcal{R}^{max} Q_\mathcal{R}^\pi - \mathcal{T}_{\mathcal{R}'}^\mu Q_\mathcal{R}^\pi) && (6)\\
&= \gamma \mathcal{P}^\mu (Q_{\mathcal{R}'}^\mu - Q_*^\pi + Q_*^\pi - Q_\mathcal{R}^\pi) + \gamma \mathcal{P}^\pi (Q_\mathcal{R}^\pi - Q_*^\pi) - (\mathcal{T}_\mathcal{R}^{max} Q_\mathcal{R}^\pi - \mathcal{T}_{\mathcal{R}'}^\mu Q_\mathcal{R}^\pi) && (7)\\
&= \gamma \mathcal{P}^\mu (Q_{\mathcal{R}'}^\mu - Q_*^\pi + Q_*^\pi - Q_\mathcal{R}^\pi) + \gamma \mathcal{P}^\pi (Q_\mathcal{R}^\pi - Q_*^\pi) - (\mathcal{T}_\mathcal{R}^{max} Q_\mathcal{R}^\pi - \mathcal{T}_{\mathcal{R}'}^\mu Q_\mathcal{R}^\pi) && (8)
\end{aligned}
$$

where $\mathcal{P}^\pi$ is the state probability transition operator with respect to the environment dynamics and policy π. Eq 6 uses the equality $Q_*^\pi = \mathcal{T}_\mathcal{R}^\pi Q_*^\pi$; Eq. 7 adopts the fact that $\mathcal{T}_\mathcal{R}^{max} Q_\mathcal{R}^\pi = \mathcal{T}_\mathcal{R}^\pi Q_\mathcal{R}^\pi$. Combining the terms $Q_{\mathcal{R}'}^\mu - Q_*^\pi$ and $Q_*^\pi - Q_\mathcal{R}^\pi$ gives us

$$
(\mathcal{I} - \gamma \mathcal{P}^\mu)(Q_{\mathcal{R}'}^\mu - Q_*^\pi) = (\gamma \mathcal{P}^\mu - \gamma \mathcal{P}^\pi)(Q_*^\pi - Q_\mathcal{R}^\pi) - (\mathcal{T}_\mathcal{R}^{max} Q_\mathcal{R}^\pi - \mathcal{T}_{\mathcal{R}'}^\mu Q_\mathcal{R}^\pi) \tag{9}
$$

where $\mathcal{I}$ is the identity operator, i.e., $\mathcal{I} Q = Q$. We define $(\mathcal{I} - \gamma \mathcal{P})^{-1} \overset{def}{=} \mathcal{I} + \sum_{t=1}^\infty \gamma^t \mathcal{P}^t$. By definition, giving initial state-action distribution β, $(\mathcal{I} - \gamma \mathcal{P}^\pi)^{-1} \beta$ is the state-action marginal distribution with respect to β and policy π. We can easily verify that $(\mathcal{I} - \gamma \mathcal{P})^{-1}(\mathcal{I} - \gamma \mathcal{P}) = \mathcal{I}$ and $(\mathcal{I} - \gamma \mathcal{P})(\mathcal{I} - \gamma \mathcal{P})^{-1} = \mathcal{I}$ since by definition, $\gamma < 1$.

Next, we derive the connection between $Q_*^\pi - Q_\mathcal{R}^\pi$ and $\mathcal{T}^\mathcal{R} Q_\mathcal{R}^\pi - Q_\mathcal{R}^\pi$, which is closely related to the result given by Munos et al. [23]:

$$
\begin{aligned}
(\mathcal{I} - \gamma \mathcal{P}^\pi)(Q_*^\pi - Q_\mathcal{R}^\pi) &= Q_*^\pi - Q_\mathcal{R}^\pi - \gamma \mathcal{P}^\pi Q_*^\pi + \gamma \mathcal{P}^\pi Q_\mathcal{R}^\pi \\
&= \mathcal{R} + \gamma \mathcal{P}^\pi Q_\mathcal{R}^\pi - (\mathcal{R} + \gamma \mathcal{P}^\pi Q_*^\pi) + Q_*^\pi - Q_\mathcal{R}^\pi \\
&= \mathcal{T}_\mathcal{R}^\pi Q_\mathcal{R}^\pi - \mathcal{T}_\mathcal{R}^\pi Q_*^\pi + Q_*^\pi - Q_\mathcal{R}^\pi \\
&= \mathcal{T}_\mathcal{R}^{max} Q_\mathcal{R}^\pi - Q_\mathcal{R}^\pi,
\end{aligned}
$$

where the result $\mathcal{T}_\mathcal{R}^\pi Q_*^\pi = Q_*^\pi$ and $\mathcal{T}_\mathcal{R}^\pi Q_\mathcal{R}^\pi = \mathcal{T}_\mathcal{R}^{max} Q_\mathcal{R}^\pi$ are used. Plug in Eq. 9, we have

$$
(\mathcal{I} - \gamma \mathcal{P}^\mu)(Q_{\mathcal{R}'}^\mu - Q_*^\pi) = (\gamma \mathcal{P}^\mu - \gamma \mathcal{P}^\pi)(\mathcal{I} - \gamma \mathcal{P}^\pi)^{-1}(\mathcal{T}_\mathcal{R}^{max} Q_\mathcal{R}^\pi - Q_\mathcal{R}^\pi) - (\mathcal{T}_\mathcal{R}^{max} Q_\mathcal{R}^\pi - \mathcal{T}_{\mathcal{R}'}^\mu Q_\mathcal{R}^\pi),
$$

Combining the above equation with Eq. 8, we get

$$
\begin{aligned}
Q_{\mathcal{R}'}^\mu - Q_*^\pi &= (\mathcal{I} - \gamma \mathcal{P}^\mu)^{-1}(\gamma \mathcal{P}^\mu - \gamma \mathcal{P}^\pi)(\mathcal{I} - \gamma \mathcal{P}^\pi)^{-1}(\mathcal{T}_\mathcal{R}^{max} Q_\mathcal{R}^\pi - Q_\mathcal{R}^\pi) - (\mathcal{I} - \gamma \mathcal{P}^\mu)^{-1}(\mathcal{T}_\mathcal{R}^{max} Q_\mathcal{R}^\pi - \mathcal{T}_{\mathcal{R}'}^\mu Q_\mathcal{R}^\pi) \\
&= \left[(\mathcal{I} - \gamma \mathcal{P}^\mu)^{-1} - (\mathcal{I} - \gamma \mathcal{P}^\pi)^{-1}\right](\mathcal{T}_\mathcal{R}^{max} Q_\mathcal{R}^\pi - Q_\mathcal{R}^\pi) - (\mathcal{I} - \gamma \mathcal{P}^\mu)^{-1}(\mathcal{T}_\mathcal{R}^{max} Q_\mathcal{R}^\pi - \mathcal{T}_{\mathcal{R}'}^\mu Q_\mathcal{R}^\pi). \qquad (10)
\end{aligned}
$$

Observe that $Q_{\mathcal{R}'}^\mu$ and Q_*^π are the optimal value function with respect to $\langle \mu, \mathcal{R}' \rangle$ and $\langle \pi, \mathcal{R} \rangle$, respectively. By definition, we have $Q_{\mathcal{R}'}^\mu = (\mathcal{I} - \gamma \mathcal{P}^\pi)^{-1} \mathcal{R}'$ and $Q_*^\pi = (\mathcal{I} - \gamma \mathcal{P}^\pi)^{-1} \mathcal{R}$ (since $(\mathcal{I} - \gamma \mathcal{P}^\pi)^{-1} \mathcal{R} = \sum_{t=0}^\infty \gamma^t \mathcal{P}^t \mathcal{R} = Q_*^\pi$).

We are now ready to prove the second result stated in the theorem (bound on training effectiveness). Since $\mathcal{T}_{\mathcal{R}'}^\mu Q_\mathcal{R}^\pi \geq \mathcal{T}_\mathcal{R}^{max} Q_\mathcal{R}^\pi$, we have $(\mathcal{T}_\mathcal{R}^{max} Q_\mathcal{R}^\pi - \mathcal{T}_{\mathcal{R}'}^\mu Q_\mathcal{R}^\pi) \leq 0$. Plug in Eq. 10 and use the equality

$$
Q_{\mathcal{R}'}^\mu - Q_*^\pi = (\mathcal{I} - \gamma \mathcal{P}^\pi)^{-1}(\mathcal{R}' - \mathcal{R}),
$$

we have

$$
(\mathcal{I} - \gamma \mathcal{P}^\mu)^{-1}(\mathcal{T}_\mathcal{R}^{max} Q_\mathcal{R}^\pi - Q_\mathcal{R}^\pi) \geq (\mathcal{I} - \gamma \mathcal{P}^\pi)^{-1}(\mathcal{T}_\mathcal{R}^{max} Q_\mathcal{R}^\pi - Q_\mathcal{R}^\pi) + (\mathcal{I} - \gamma \mathcal{P}^\pi)^{-1}(\mathcal{R} - \mathcal{R}'),
$$

which is equivalent to the second result stated in the theorem (bound on training effectiveness).

To prove the first result stated in the theorem (bound on stability), we start from rearranging Eq. 10:

$$
\begin{aligned}
\left[(\mathcal{I}-\gamma\mathcal{P}^{\pi})^{-1}-(\mathcal{I}-\gamma\mathcal{P}^{\mu})^{-1}\right](\mathcal{T}_{\mathcal{R}}^{max}Q_{\mathcal{R}}^{\pi}-Q_{\mathcal{R}}^{\pi}) &= -(\mathcal{I}-\gamma\mathcal{P}^{\mu})^{-1}((\mathcal{I}-\gamma\mathcal{P}^{\mu})(Q_{\mathcal{R}'}^{\mu}-Q_{*}^{\pi})+\mathcal{T}_{\mathcal{R}}^{max}Q_{\mathcal{R}}^{\pi}-\mathcal{T}_{\mathcal{R}'}^{\mu}Q_{\mathcal{R}}^{\pi}) \\
&= -(\mathcal{I}-\gamma\mathcal{P}^{\mu})^{-1}(\mathcal{R}'+\gamma\mathcal{P}^{\mu}Q_{*}^{\pi}-Q_{*}^{\pi}+\mathcal{T}_{\mathcal{R}}^{max}Q_{\mathcal{R}}^{\pi}-\mathcal{T}_{\mathcal{R}'}^{\mu}Q_{\mathcal{R}}^{\pi}) \\
&= -(\mathcal{I}-\gamma\mathcal{P}^{\mu})^{-1}(\mathcal{R}+\gamma\mathcal{P}^{\mu}Q_{*}^{\pi}-Q_{*}^{\pi}+\gamma\mathcal{P}^{\pi}Q_{\mathcal{R}}^{\pi}-\gamma\mathcal{P}^{\mu}Q_{\mathcal{R}}^{\pi}) \\
&\geq -(\mathcal{I}-\gamma\mathcal{P}^{\mu})^{-1}(\gamma\mathcal{P}^{\pi}Q_{\mathcal{R}}^{\pi}-\gamma\mathcal{P}^{\mu}Q_{\mathcal{R}}^{\pi}) \qquad (11) \\
&\geq -\gamma(\mathcal{I}-\gamma\mathcal{P}^{\mu})^{-1}(\mathcal{P}^{\pi}-\mathcal{P}^{\mu})(\mathcal{I}-\gamma\mathcal{P}^{\pi})^{-1}\mathcal{R} \qquad (12) \\
&= \left[(\mathcal{I}-\gamma\mathcal{P}^{\pi})^{-1}-(\mathcal{I}-\gamma\mathcal{P}^{\mu})^{-1}\right]\mathcal{R}, \qquad (13)
\end{aligned}
$$

where Eq. 11 uses the inequality $\mathcal{P}^{\mu}Q_{*}^{\pi} \leq \mathcal{P}^{\pi}Q_{*}^{\pi}$, and Eq. 12 follows from $Q_{\mathcal{R}}^{\pi} \leq Q_{*}^{\pi} = (\mathcal{I}-\gamma\mathcal{P}^{\pi})^{-1}\mathcal{R}$. Rewriting Eq. 13 gives us the first result stated in the theorem:

$$
(\mathcal{I}-\gamma\mathcal{P}^{\pi})^{-1}(\mathcal{T}_{\mathcal{R}}^{max}Q_{\mathcal{R}}^{\pi}-Q_{\mathcal{R}}^{\pi}) \geq (\mathcal{I}-\gamma\mathcal{P}^{\mu})^{-1}(\mathcal{T}_{\mathcal{R}}^{max}Q_{\mathcal{R}}^{\pi}-Q_{\mathcal{R}}^{\pi})+\left[(\mathcal{I}-\gamma\mathcal{P}^{\pi})^{-1}-(\mathcal{I}-\gamma\mathcal{P}^{\mu})^{-1}\right]\mathcal{R}.
$$