Exploring Pessimism and Optimism Dynamics in Deep Reinforcement Learning

Bahareh Tasdighi **Nicklas Werge** **Yi-Shan Wu** **Melih Kandemir**
Department of Mathematics and Computer Science
University of Southern Denmark
tasdighi@imada.sdu.dk werge@sdu.dk {yswu,kandemir}@imada.sdu.dk

Abstract

Off-policy actor-critic algorithms have shown promise in deep reinforcement learning for continuous control tasks. Their success largely stems from leveraging pessimistic state-action value function updates, which effectively address function approximation errors and improve performance. However, such pessimism can lead to under-exploration, constraining the agent's ability to explore/refine its policies. Conversely, optimism can counteract under-exploration, but it also carries the risk of excessive risk-taking and poor convergence if not properly balanced. Based on these insights, we introduce Utility Soft Actor-Critic (USAC), a novel framework within the actor-critic paradigm that enables independent control over the degree of pessimism/optimism for both the actor and the critic via interpretable parameters. USAC adapts its exploration strategy based on the uncertainty of critics through a utility function that allows us to balance between pessimism and optimism separately. By going beyond binary choices of optimism and pessimism, USAC represents a significant step towards achieving balance within off-policy actor-critic algorithms. Our experiments across various continuous control problems show that the degree of pessimism or optimism depends on the nature of the task. Furthermore, we demonstrate that USAC can outperform state-of-the-art algorithms for appropriately configured pessimism/optimism parameters.

1 Introduction

Deep reinforcement learning (RL) faces significant challenges when navigating the complex landscape of high-dimensional state spaces and non-linear state-action functions. The deadly triad of function approximation, off-policy learning, and bootstrapping compounds these challenges, leading to instability and poor sample efficiency [19, 27, 33]. One particularly critical consequence is the *overestimation bias* [29], where the estimated state-action values (Q-values) exceed their true counterparts [31, 32]. The overestimation bias is typically addressed through a pessimistic approach that use the minimum between two critics' estimations [10, 12]. While effective in reducing overestimation, this approach can lead to *pessimistic under-exploration* [7], which limits the agent's ability to explore and discover new policies. A natural remedy for pessimistic under-exploration could involve integrating optimism into actors' exploration; however, the balance between pessimism and optimism is not straightforward, as this balance varies across tasks and evolves during the training process [20].

The core focus of this paper is to explore the dynamics between pessimism and optimism in deep RL, especially within actor critics. To address this dynamic, we introduce the novel framework termed Utility Soft Actor-Critic (USAC). Unlike binary approaches, USAC considers pessimism and optimism as points along a spectrum. Our aim is to address the overestimation bias while avoiding pessimistic under-exploration. In the next section, we explore the current literature on this topic and elaborate on our main contributions.

17th European Workshop on Reinforcement Learning (EWRL 2024).

2 Exploring pessimism and optimism in deep reinforcement learning

The primary focus of this work is to understand the interplay between pessimism and optimism and their impact on overestimation bias and pessimistic under-exploration. In this section, we delve into related work addressing these challenges, exploring various strategies to overcome them. Moreover, we conclude this section by outlining our main contributions.

Addressing overestimation bias with pessimism A pessimistic approach has proven effective in reducing the overestimation bias[1] – as demonstrated in Twin Delayed DDPG[2] (TD3) [10] – it also introduces challenges related to pessimistic under-exploration [7]. Soft Actor-Critic (SAC) [12] addresses this challenge by integrating an entropy term into the actor's policy decision-making process, thus making the actor stochastic. By incorporating the entropy, SAC encourages exploration and diversifies the actor's actions, aiming to alleviate the adverse effects of pessimistic under-exploration. However, despite its effectiveness, entropy regularization alone do not resolve the underlying issues.

Exploring optimism to counter pessimistic under-exploration An intuitive approach to tackle pessimistic under-exploration is based on a general form of the *optimism in the face of uncertainty* principle. Ciosek et al. [7] introduced the Optimistic Actor-Critic (OAC) algorithm, which adopts an optimistic approximate upper confidence bound on the Q-value function to enhance policy exploration. However, it's crucial to note that aggressive exploration, as encouraged by optimistic algorithms like OAC, can be risky, particularly in environments where the size of overestimation is not well-understood [20].

Balancing between pessimism and optimism The balance of pessimism and optimism is nuanced and it varies across different tasks and develops throughout the training process [20]. The Tactical Optimism and Pessimism (TOP) algorithm, introduced by Moskovitz et al. [20], addresses this variability by dynamically balancing pessimism and optimism using multi-armed bandits [6, 16]. However, unlike algorithms such as OAC, which lean towards optimism, TOP learns to adjust the degree of pessimism or optimism based on the prevailing uncertainty. However, it uses the same level of pessimism/optimism for both critic and actor training, which limits its versatility.

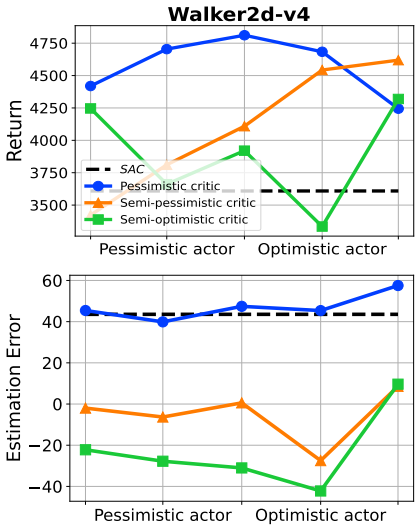

Figure 1: Impact of pessimism/optimism levels on critic and actor. Top: Final returns. Bottom: Estimation error (i.e., true - estimation); positive values means underestimation, whereas negative values means overestimation.

USAC and our main contributions Inspired by these insights, we see pessimism and optimism as points along a spectrum rather than binary opposites. While TOP is rooted in the TD3 algorithm, our approach stems from SAC. We introduce a novel framework called USAC, which quantifies the inherent uncertainty in learning through a utility function. Unlike binary approaches, USAC adapts its exploration strategy based on the critics' distribution through uncertainty estimates of their Q-values. Moreover, USAC permits the fine-tuning of pessimism or optimism levels during critic or actor training. Through this methodology, USAC strives for a balanced pessimism-optimism trade-off that overcomes both overestimation bias and pessimistic under-exploration. Figure 1 illustrates the impact of varying levels of pessimism and optimism on both the critic and actor components. This indicates that optimal levels of pessimism and optimism can effectively address both overestimation (or underestimation) and pessimistic under-exploration, thereby enhancing the overall return. Through empirical evaluations and comparisons with state-of-the-art algorithms, we demonstrate the versatility and robustness of USAC across diverse continuous control tasks, paving the way for more adaptive and efficient learning algorithms in the field of deep RL.

[1]The overestimation bias stems from inherent noise and approximation errors introduced by the function approximations and is further amplified by bootstrapping [29, 31, 32].

[2]The Deep Deterministic Policy Gradient (DDPG) algorithm was proposed by Lillicrap et al. [17].

3 Preliminary

Throughout this paper, we denote $\mathcal{P}(\Omega)$ as the set of all probability distributions on Ω and let $\mathcal{B}(\Omega)$ be the set of bounded functions on Ω.

Markov decision processes (MDPs) An MDP can be represented by the tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, p, p_0, r, \gamma \rangle$ [23]. Here, $\mathcal{S}$ and $\mathcal{A}$ denote the continuous state and action spaces. The function $p(s'|s,a)$ for $(s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ represents the unknown transition probability from the current state s and action a to the next state s'. This function satisfies the condition $\int_{s'} p(s'|s,a)ds' = 1$. The initial state distribution is denoted by $p_0 \in \mathcal{P}(\mathcal{S})$, while $r : \mathcal{S} \times \mathcal{A} \to [0, B_r]$ represents the bounded reward function with $B_r > 0$, and $\gamma \in [0,1]$ stands for the discount factor.

Policies Let $\Pi = \{\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})\}$ be the set of policies. The interaction between the agent and the MDP $\mathcal{M}$ under some policy $\pi \in \Pi$ progresses iteratively: At each time step $t \in \mathbb{N}$, the agent observes state $s_t \in \mathcal{S}$, chooses action $a_t \in \mathcal{A}$ based on the policy $a_t \sim \pi(\cdot|s_t)$, receives a (bounded) reward $r_t := r(s_t, a_t)$, and transitions to the next state $s_{t+1} \sim p(\cdot|s_t, a_t)$. Additionally, we define $p^\pi(s', a'|s, a) = p(s'|s,a)\pi(a'|s')$ as the one-step transition probability from (s,a) to (s',a').

Maximum entropy RL The standard goal of RL is to find a policy π that maximizes the expected sum of discounted rewards $J(\pi) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_t]$ with initial state $s_0 \sim p_0$ [3, 27, 28]. To improve exploration, we will consider the more general maximum entropy RL framework [11, 12, 25, 36]. In this framework, one aims not only to maximize the expected (discounted) return but also to maximize the (discounted) entropy of the actions suggested by the stochastic policies. Formally, this is achieved by incorporating an entropy term, tempered by a parameter $\alpha > 0$, which governs the relative importance of the return compared to the policy entropy. The maximum entropy objective function is defined as $J_\alpha(\pi) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t (r_t + \alpha \mathcal{H}(\pi(\cdot|s_t)))\right]$, where the policy entropy term $\mathcal{H}(\pi(\cdot|s)) = -\mathbb{E}_{a\sim\pi(\cdot|s)}[\log \pi(a|s)]$.

A policy is typically assessed by a state-action value function $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ [35]. Specifically, starting from initial state $s_0 = s$ and action $a_0 = a$, the state-action value function Q^π obtained by following policy π satisfies $Q^\pi(s,a) = J_\alpha(\pi)$. In addition, this is the unique fixed-point solution to the soft Bellman operator $T^\pi : \mathcal{B}(\mathcal{S} \times \mathcal{A}) \to \mathcal{B}(\mathcal{S} \times \mathcal{A})$ [3]:

$$T^\pi Q(s,a) = r(s,a) + \gamma \mathbb{E}_{(s',a')\sim p^\pi(\cdot|s,a)}[Q(s',a') - \alpha \log \pi(a'|s')], \tag{1}$$

i.e., $T^\pi Q^\pi(s,a) = Q^\pi(s,a)$ for any $(s,a) \in \mathcal{S} \times \mathcal{A}$.

Actor-critic algorithms One approach to learning the optimal policy is through the framework of actor-critic algorithms. In these algorithms, an agent simultaneously learns a policy (the actor), responsible for selecting actions that maximize expected return, and an estimation of the state-action value function (the critic), responsible for evaluating the policy's quality through iterative updates. Formally, the *critic* Q estimates the state-action value function Q^π (derived by following policy π) by

$$\arg\min_Q \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}[(Q(s,a) - y(s,a,r,s'))^2], \tag{2}$$

where (s,a,r,s') is sampled from the replay buffer $\mathcal{D}$, and the critic's *target value* is defined as

$$y(s,a,r,s') = r + \gamma[\bar{Q}(s',a') - \alpha \log \pi(a'|s')], \quad a' \sim \pi(\cdot|s'), \tag{3}$$

which is built using a *target critic* $\bar{Q}$ that may differ from the critic Q [18, 19]. The *actor* learns its policy by solving the following:

$$\arg\max_{\pi\in\Pi} \mathbb{E}_{a\sim\pi(\cdot|s)}[\tilde{Q}(s,a) - \alpha \log \pi(a|s)], \tag{4}$$

where $\tilde{Q}$ is (again) an estimate of the state-action value function Q^π, which can differ from both the critic Q and the target critic $\bar{Q}$. In scenarios where the agent do a full Bellman backup without approximation errors, the critic Q, the target critic $\bar{Q}$, and the critic used to update the actor $\tilde{Q}$ are all identical to Q^π [12, 27]. However, variations in Q, $\bar{Q}$, and $\tilde{Q}$ are often introduced to address the challenges we outlined in Section 2; we will explore some of these variations in more details below.

Exploring pessimism and optimism in actor-critic algorithms A common strategy to address overestimation involves estimating the state-action value function using twin critics, along with associated target critics that are delayed versions of these critics [10, 17–19]. Specifically, let $Q_1, Q_2 : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ be the two critics and $\bar{Q}_1, \bar{Q}_2 : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ the two target critics that are delayed versions of the critics. The pessimistic strategy, employed by TD3 [10], SAC [12] and OAC [7], is to use the minimum of the two target critics $\{\bar{Q}_k\}_{k=1,2}$ to compute the target value of the critics in (3):

$$\bar{Q}(s,a) = \min_{k=1,2} \bar{Q}_k(s,a). \tag{5}$$

This target is used to train both critics Q_1 and Q_2. Similarly, the same approach is employed to update the actor (in TD3 and SAC). Here, the critic to update the actor, $\tilde{Q}$ in (4), is determined by the minimum of the two critics $\{Q_k\}_{k=1,2}$:

$$\tilde{Q}(s,a) = \min_{k=1,2} Q_k(s,a). \tag{6}$$

Ciosek et al. [7] noted that the pessimistic approach, described in (5) and (6), can lead to pessimistic under-exploration. To prevent this, they proposed embracing the *optimism in the face of uncertainty* principle in their OAC algorithm. However, this approach can potentially make exploration too aggressive, thereby negatively impacting overall performance [20]. Instead, the balance between pessimism and optimism should be tailored to the problem's nature. Moskovitz et al. [20] suggested using the following target critic to achieve this balance:

$$\bar{Q}(s,a) = \bar{\mu}(s,a) + \beta\bar{\sigma}(s,a), \quad \beta \in \mathbb{R}, \tag{7}$$

where the mean $\bar{\mu}$ and (unbiased) standard deviation $\bar{\sigma}$ are constructed using the two target critics $\bar{Q}_1, \bar{Q}_2$; $\bar{\mu}(s,a) = \frac{1}{2}(\bar{Q}_1(s,a) + \bar{Q}_2(s,a))$ and $\bar{\sigma}(s,a) = \sqrt{\sum_{k=1,2}(\bar{Q}_k(s,a) - \bar{\mu}(s,a))^2}$. The critic to update the actor, $\tilde{Q}$, can be defined in a similar manner from the two critics $\{Q_k\}_{k=1,2}$. Moskovitz et al. [20] classify (7) as *optimistic* when $\beta \geq 0$ and *pessimistic* when $\beta < 0$. In the special case when $\beta = -1/\sqrt{2}$, (7) simplifies to taking the minimum of the two critics – as in (5) and (6). The TOP algorithm tries to adjust the balance between pessimism and optimism by using a multi-armed bandit algorithm to select β from the set $\{-1, 0\}$; here $\beta = -1$ corresponds to a pessimistic estimate and $\beta = 0$ to an optimistic one. Although β can vary over time using a bandit algorithm, TOP uses the same β for both $\bar{Q}$ and $\tilde{Q}$.

4 Utility Soft Actor-Critic (USAC): framework and algorithms

Unlike the aforementioned algorithms, our framework decouples the levels of optimism and pessimism for the actor and critic. This novel approach allows us to be pessimistic in learning the critic while being optimistic in actor training. By doing so, we address the overestimation bias observed during critic training and overcome the problem of pessimistic under-exploration faced by the actor. Consequently, our approach promotes both a stable critic and an explorative actor simultaneously.

In this section, we establish the theoretical groundwork for exploring the balance between optimism and pessimism. We introduce the Utility Soft Actor-Critic (USAC) framework, which seamlessly integrates the uncertainty of critics into the actor learning process. Our framework guarantees policy improvement, as detailed in Section 4.1, while offering flexibility in navigating the optimism-pessimism trade-off. Moving on to Sections 4.2 and 4.3, we present the USAC algorithm and demonstrate how specific choices within our framework can emulate the optimism and pessimism strategies of algorithms like TD3 [10], SAC [12], OAC [7], and TOP [20]; in Section 5, we will explore these varying levels of optimism and pessimism across five MuJoCo environments. The pseudo-code of USAC is provided in Algorithm 1.

Before opening the entire discussion, we first introduce how to quantify the uncertainty of the Q-functions.

Quantifying the uncertainty of the Q-value distribution Let $\mathcal{Q} \in \mathcal{P}(\mathcal{B}(\mathcal{S} \times \mathcal{A}))$ be a distribution over Q-functions, serving as the agent's estimate of the Q-functions. The dispersion of the distribution reflects the uncertainty of the learned Q-functions. To characterize such a distribution, we use the utility function of $\mathcal{Q}$, defined by

$$U_\lambda^{\mathcal{Q}}(s,a) = \frac{1}{\lambda} \log \mathbb{E}_{Q\sim\mathcal{Q}}[\exp(\lambda Q(s,a))], \quad \lambda \in \mathbb{R}, \tag{8}$$

where $U_\lambda^{\mathcal{Q}}(s,a) = \mathbb{E}_{Q\sim\mathcal{Q}}[Q(s,a)]$ when $\lambda = 0$. This utility function is commonly used for measuring risk in finance, economics, and decision-making under uncertainty [9, 34].[3] It also shares similarities with the utility functions employed in risk-sensitive RL [14, 22, 26] and distributional RL [1, 2, 24]. It's important to note that we apply this utility function to the $\mathcal{Q}$ distribution rather than the return distribution, as is typically done in risk-sensitive and distributional RL.

In this work, we use (8) as a measure of pessimism or optimism of the $\mathcal{Q}$ distribution: a positive value of λ implies an *optimistic* view, while a negative value indicates a *pessimistic* view. This interpretation is clear from the Taylor expansion: $U_\lambda^{\mathcal{Q}} = \mathbb{E}_{Q\sim\mathcal{Q}}[Q] + \frac{\lambda}{2}\mathbb{V}_{Q\sim\mathcal{Q}}[Q] + \mathcal{O}(\lambda^2)$. Consequently, the utility function $U_\lambda^{\mathcal{Q}}$ spans the entire spectrum of the $\mathcal{Q}$ distribution by varying λ. Another important property of the utility is that when $\mathcal{Q}$ is a Dirac delta distribution centered at Q^π, the Q-function of a given policy $\pi \in \Pi$, the expected utility function $U_\lambda^{\mathcal{Q}}$ is equivalent to Q^π for any finite λ.

4.1 USAC framework: policy evaluation, improvement and guarantees

In this section, we will explain the policy evaluation and policy improvement steps, along with the corresponding guarantees for our USAC framework.

For any given policy $\pi \in \Pi$, the utility $U_\lambda^{\mathcal{Q}} \in \mathcal{B}(\mathcal{S}\times\mathcal{A})$ will reduce to Q^π for any $|\lambda| < \infty$, when $\mathcal{Q}$ is a Dirac delta distribution centered at Q^π. This implies that the deviation of $U_\lambda^{\mathcal{Q}}$ from Q^π is due to the uncertainty in the learned Q-function. In fact, we observe that the Bellman operator in (1), $T^\pi : \mathcal{B}(\mathcal{S}\times\mathcal{A}) \to \mathcal{B}(\mathcal{S}\times\mathcal{A})$, should also bring the utility $U_\lambda^{\mathcal{Q}}$ to its fixed point Q^π [3]. Therefore, ultimately, the Bellman error

$$T^\pi U_\lambda^{\mathcal{Q}}(s,a) - U_\lambda^{\mathcal{Q}}(s,a) = r(s,a) + \gamma\mathbb{E}_{(s',a')\sim p^\pi(\cdot|s,a)}\left[U_\lambda^{\mathcal{Q}}(s',a') - \alpha\log(\pi(a'|s'))\right] - U_\lambda^{\mathcal{Q}}(s,a)$$

should be diminished. Nevertheless, it fosters exploration during learning by maintaining a distribution over the Q-functions, which can guide exploration.

On the other hand, given a $\mathcal{Q}$ distribution associated with the current policy π, denoted as $\mathcal{Q}^\pi$, we update the policy, similar to (4), through the following maximization problem:

$$\arg\max_{\pi\in\Pi} \mathbb{E}_{a\sim\pi(\cdot|s)}[U_\lambda^{\mathcal{Q}^\pi}(s,a) - \alpha\log\pi(a|s)].$$

Extending the policy iteration process from SAC to utility functions ensures that the guarantees of policy improvement still hold [11, 12].

4.2 USAC algorithm

The USAC learning process follows the general actor-critic algorithm. In particular, we estimate the desired quantities by samples from a replay buffer $(s,a,r,s') \sim \mathcal{D}$. As the operator T^π should bring the utility $U_\lambda^{\mathcal{Q}}$ to its fixed-point Q^π, the USAC framework learns the critic Q (an estimate of Q^π) as in (2) but with the critic's *target value* defined by

$$y(s,a,r,s') = r + \gamma[U_{\lambda_{\text{critic}}}^{\bar{\mathcal{Q}}}(s',a') - \alpha\log\pi(a'|s')], \quad a' \sim \pi(\cdot|s'), \tag{9}$$

which is built using a *target critic distribution* $\bar{\mathcal{Q}}$ with some utility parameter λ_{critic} chosen specifically for the critic. Similar to (4), the *actor* learns its policy by

$$\arg\max_{\pi\in\Pi} \mathbb{E}_{a\sim\pi(\cdot|s)}[U_{\lambda_{\text{actor}}}^{\tilde{\mathcal{Q}}}(s,a) - \alpha\log\pi(a|s)], \tag{10}$$

using $\tilde{\mathcal{Q}}$ as the estimate for $\mathcal{Q}$ and λ_{actor} as the utility parameter for the actor.

Note that although Moskovitz et al. [20] consider an optimism/pessimism parameter that can switch between two values over time, they use the same parameter for both actor and critic training. In contrast, our work decouples the optimism/pessimism levels of the actor and critic. This novel approach allows us to select a pessimistic parameter λ_{critic} for learning the critic while using an optimistic parameter λ_{actor} for training the actor. This strategy effectively addresses the overestimation bias observed during critic training and mitigates the issue of pessimistic under-exploration

[3] The utility function in (8) is also referred to as *expected utility*, *exponential utility*, *exponential risk measure*, *generalized mean*, or *entropic risk measure* according to the context [8, 9].

encountered by the actor. Consequently, our approach promotes a stable critic and an explorative actor simultaneously.

Later in the experiments presented in Section 5, we will explore various combinations of pessimism and optimism levels. Before proceeding, we will ground the algorithm by considering a specific distribution. It's worth noting that while we focus on the Laplace distribution in this paper, the framework described in this section is applicable to other distributions as well. The specific choice of Laplace is simply because it makes it more tangible to fine-tune the level of pessimism/optimism in the critic and actor, and to compare with other actor-critic algorithms.

4.3 USAC algorithm: Laplace distribution

The utility formulation in (8) enables us to compute the utility for any distribution with a moment-generating function. Let's consider a specific case where $\mathcal{Q}$ follows a Laplace distribution. The algorithm overview is provided in Algorithm 1. Here, we denote the utility parameter associated with this distribution as κ, reflecting its characteristics within the Laplace context.

Proposition 1. *Suppose $\mathcal{Q}$ is a Laplace distribution. Then, for any $(s, a) \in \mathcal{S} \times \mathcal{A}$, the utility function of $\mathcal{Q}$ can be expressed as*

$$U_\kappa^{\mathcal{Q}}(s,a) = \mu_{\mathcal{Q}}(s,a) + g(\kappa)\sigma_{\mathcal{Q}}(s,a) \quad \textit{for} \quad \kappa \in (-1,1), \tag{11}$$

where $\mu_{\mathcal{Q}}$ and $\sigma_{\mathcal{Q}}$ represent the mean and standard deviation of $\mathcal{Q}$, respectively, and

$$g(\kappa) = \begin{cases} \log(1/(1-\kappa^2))/\sqrt{2}\kappa & \textit{for } \kappa \in (-1,1)\backslash\{0\}, \\ 0 & \textit{for } \kappa = 0. \end{cases} \tag{12}$$

The proof of Proposition 1 can be found in Appendix A. Additionally, we present a corresponding proposition for Gaussian-distributed Q-values in Appendix B. In Proposition 1, we observe that by varying κ in the small interval $(-1, 1)$, $U_\kappa^{\mathcal{Q}}$ spans the entire spectrum of the $\mathcal{Q}$ distribution. Specifically, as $\lim_{\kappa\to-1} g(\kappa) \to -\infty$, $\lim_{\kappa\to0} g(\kappa) \to 0$, and $\lim_{\kappa\to1} g(\kappa) \to \infty$.

Now suppose the uncertainty follows a Laplace distribution. Following the algorithmic steps outlined in Section 4.2, we can estimate the utility of the target critic distribution $\bar{\mathcal{Q}}$ by two target critics $\{\bar{Q}_k\}_{k=1,2}$, which are delayed versions of the critics $\{Q_k\}_{k=1,2}$. The utility function is given by $U_{\kappa_{\text{critic}}}^{\bar{\mathcal{Q}}} = \mu_{\bar{\mathcal{Q}}} + g(\kappa_{\text{critic}})\sigma_{\bar{\mathcal{Q}}}$ with utility parameter $\kappa_{\text{critic}} \in (-1, 1)$, where

$$\mu_{\bar{\mathcal{Q}}}(s,a) = \frac{1}{2}(\bar{Q}_1(s,a) + \bar{Q}_2(s,a)) \quad \text{and} \quad \sigma_{\bar{\mathcal{Q}}}(s,a) = \frac{1}{2}|\bar{Q}_1(s,a) - \bar{Q}_2(s,a)|. \tag{13}$$

Hence, the target value to train the critic becomes

$$y(s,a,r,s') = r + \gamma[\mu_{\bar{\mathcal{Q}}}(s',a') + g(\kappa_{\text{critic}})\sigma_{\bar{\mathcal{Q}}}(s',a') - \alpha\log\pi(a'|s')], \quad a' \sim \pi(\cdot|s'). \tag{14}$$

Similarly, the actor learns its policy by

$$\arg\max_{\pi\in\Pi} \mathbb{E}_{a\sim\pi(\cdot|s)}[\mu_{\tilde{\mathcal{Q}}}(s,a) + g(\kappa_{\text{actor}})\sigma_{\tilde{\mathcal{Q}}}(s,a) - \alpha\log\pi(a|s)], \quad \kappa_{\text{actor}} \in (-1,1), \tag{15}$$

where $\mu_{\tilde{\mathcal{Q}}}$ and $\sigma_{\tilde{\mathcal{Q}}}$ are estimated in similar manner as (13) on $\tilde{\mathcal{Q}}$ using the critics $\{Q_k\}_{k=1,2}$. The pseudo-code of this algorithm can be found in Algorithm 1.

We note that a smaller κ value indicates more pessimism, while a larger κ value indicates more optimism. By choosing κ such that $g(\kappa) = -1$ (i.e., $\kappa \approx -0.83$), the utility in (11) simply reduces to the minimum of the two critics. Thus, choosing $\kappa_{\text{critic}} = \kappa_{\text{actor}} \approx -0.83$, we obtain the pessimism choices of SAC [12] and TD3 [10]. On the other hand, by choosing $\kappa_{\text{critic}} \approx -0.83$ and a positive κ_{actor}, we have a pessimism critic and an optimistic actor as explored in OAC [7]. Lastly, by letting a multi-armed bandit algorithm choose $\kappa_{\text{critic}} = \kappa_{\text{actor}} \in \{-0.916563, 0\}$, we recover the pessimistic/optimistic design suggested by TOP [20].[4] We note that by choosing -0.916563, TOP considers a more pessimistic critic target than SAC, TD3, and OAC.

In Section 5, we will explore various $(\kappa_{\text{critic}}, \kappa_{\text{actor}})$ pairs and discuss whether the previous wisdom about the effectiveness of these strategies holds across different environments.

[4]To recover the pessimistic case of the TOP algorithm, we can choose $\kappa = -0.916563$, which corresponds to having the function g in (12) equal to $\sqrt{2}$.

Algorithm 1 USAC with Laplace distribution (Section 4.3)

Input: Averaging parameter $\tau \in (0,1)$, learning rates $\eta_\theta, \eta_\phi > 0$, mini-batch size $n \in \mathbb{N}$, pessimism/optimism coefficients $\kappa_{\text{actor}}, \kappa_{\text{critic}} \in (-1,1)$
Initialize: Replay buffer $\mathcal{D} = \varnothing$, initial state $s_0 \sim p_0$, critic $\{\theta_k\}_{k=1,2}$, target critic $\{\bar{\theta}_k\}_{k=1,2}$ and actor policy ϕ
for each time step **do**
  for each environment step **do**
    $a_t \sim \pi_\phi(\cdot|s_t)$ ▷ Sample action from the policy ϕ
    $s_{t+1} \sim p(\cdot|s_t, a_t)$ ▷ Sample transition from the environment
    $\mathcal{D} \leftarrow \mathcal{D} \cup (s_t, a_t, r_t, s_{t+1})$ ▷ Store transition in the replay pool
  end for
  for each training step **do**
    $\{(s_i, a_i, r_i, s'_i, a'_i) : (s_i, a_i, r_i, s'_i) \sim \mathcal{D}, a'_i \sim \pi_\phi(\cdot|s'_i)\}_{i=1}^n$ ▷ Sample mini-batch
    for $i \in \{1, \dots, n\}$ **do** ▷ Compute target critic distribution $\bar{\mathcal{Q}}$
      $\mu_{\bar{\mathcal{Q}}}(s_i, a_i) \leftarrow \frac{1}{2}(Q_{\bar{\theta}_1}(s_i, a_i) + Q_{\bar{\theta}_2}(s_i, a_i))$
      $\sigma_{\bar{\mathcal{Q}}}(s_i, a_i) \leftarrow \frac{1}{2}|Q_{\bar{\theta}_1}(s_i, a_i) - Q_{\bar{\theta}_2}(s_i, a_i)|$
    end for
    for each critic, $k \in \{1, 2\}$ **do**
      $\theta_k \leftarrow \theta_k - \eta_\theta \nabla_{\theta_k} \left\{ \frac{1}{n} \sum_{i=1}^n \left(Q_{\theta_k}(s_i, a_i) - y(s_i, a_i, r_i, s'_i)\right)^2 \right\}$ ▷ Update critic
      with $y(s_i, a_i, r_i, s'_i) = r_i + \gamma[\mu_{\bar{\mathcal{Q}}}(s'_i, a'_i) + g(\kappa_{\text{critic}})\sigma_{\bar{\mathcal{Q}}}(s'_i, a'_i) - \alpha \log \pi_\phi(a'_i|s'_i)]$
    end for
    for $i \in \{1, \dots, n\}$ **do** ▷ Compute actor distribution $\tilde{\mathcal{Q}}$
      $\mu_{\tilde{\mathcal{Q}}}(s_i, a_i) \leftarrow \frac{1}{2}(Q_{\theta_1}(s_i, a_i) + Q_{\theta_2}(s_i, a_i))$
      $\sigma_{\tilde{\mathcal{Q}}}(s_i, a_i) \leftarrow \frac{1}{2}|Q_{\theta_1}(s_i, a_i) - Q_{\theta_2}(s_i, a_i)|$
    end for ▷ Update actor
    $\phi \leftarrow \phi + \eta_\phi \nabla_\phi \left\{ \frac{1}{n} \sum_{i=1}^n \left(\mu_{\tilde{\mathcal{Q}}}(s_i, a_i) + g(\kappa_{\text{actor}})\sigma_{\tilde{\mathcal{Q}}}(s_i, a_i) - \alpha \log \pi_\phi(a_i|s_i)\right) \right\}$
    for each critic, $k \in \{1, 2\}$ **do**
      $\bar{\theta}_k \leftarrow \tau\theta_k + (1-\tau)\bar{\theta}_k$ ▷ Update target critic
    end for
  end for
end for

5 Experiments

The goal of our experiments is to explore the interplay between critic and actor pessimism/optimism, estimation error (i.e., overestimation/underestimation), and performance. We compare our USAC algorithm with Laplace distribution (Section 4.3) with prior off-policy actor-critic algorithms. Specifically, we consider SAC [12], TD3 [10], OAC [7] and TOP [20] as baselines, as they represent the most relevant algorithms that incorporate optimism and pessimism. For SAC and USAC, we use auto-tuned entropy temperature α [13], while for OAC and TOP, we adopt the best parameters reported in their respective papers.

We conduct our experiments on 5 continuous control environments in the MuJoCo physics engine [30]. Each experiment is repeated across 5 different seeds, each comprising one million time steps, and evaluated over 10 episodes. The reported results represent the average (along with the standard deviation) across these repetitions. Additional details about architectures, baselines, computational cost, environment properties, grid search, hyper-parameters, learning curves, area under learning curves, estimation error plots and their values can be found in Appendix C.

Insights on USAC performance In Table 1, we present two sets of results for our USAC algorithm: a default and a best pair of parameters $(\kappa_{\text{critic}}, \kappa_{\text{actor}})$; we will discuss how these parameters are selected below. These results demonstrate that our USAC algorithm can outperform (or match) all baseline algorithms across all environments. This success reaffirms our hypothesis, as outlined in Section 4, that the levels of pessimism and optimism should vary between the critic and actor, as well as across environments [20]. In particular, our findings reveals that there exist pairs of utility parameters, $(\kappa_{\text{critic}}, \kappa_{\text{actor}})$, capable of overcoming overestimation while facilitating exploration, thus enhancing overall performance. In addition, USAC smaller standard deviation indicates an improved

model stability during learning. The corresponding learning curves and estimation error curves for these results are presented in Figures 5 and 6 (Appendix C).

Table 1: Final return on MuJoCo environments trained with 1M time steps, averaged over 5 seeds. The best algorithms are highlighted in **bold**. $\pm$ corresponds to the standard deviation across repetitions. The default $(\kappa_{\text{critic}}, \kappa_{\text{actor}})$ are both -0.831559. The best $(\kappa_{\text{critic}}, \kappa_{\text{actor}})$ are listed in Table 2.

Environment	USAC (**ours**)		SAC	TD3	OAC	TOP
	Default $(\kappa_{\text{critic}}, \kappa_{\text{actor}})$	Best $(\kappa_{\text{critic}}, \kappa_{\text{actor}})$				
`Ant-v4`	$\mathbf{5139 \pm 978}$	$\mathbf{5158 \pm 1186}$	4756 ± 1411	4091 ± 1303	4177 ± 1392	4334 ± 1276
`HalfCheetah-v4`	$\mathbf{11024 \pm 849}$	$\mathbf{11736 \pm 317}$	10763 ± 895	10570 ± 1400	8684 ± 1678	7311 ± 3074
`Hopper-v4`	3194 ± 810	$\mathbf{3442 \pm 247}$	3185 ± 537	1986 ± 1154	3293 ± 113	$\mathbf{3367 \pm 154}$
`Humanoid-v4`	$\mathbf{5602 \pm 505}$	$5602 \pm 505^{\star}$	$\mathbf{5503 \pm 373}$	5149 ± 737	5390 ± 234	5332 ± 445
`Walker2d-v4`	$\mathbf{4525 \pm 534}$	$\mathbf{4530 \pm 767}$	3757 ± 1282	4369 ± 601	3467 ± 1200	4317 ± 631

Exploring the dynamics of pessimistic and optimistic with USAC To explore the dynamics of pessimism and optimism in USAC, we conducted a grid search across the 5 continuous control environments in MuJoCo. More details about this grid search can be found in Appendix C. We examined critics ranging from pessimistic to semi-optimistic and actors from pessimistic to optimistic. Specifically, we used $\kappa_{\text{critic}} \in \{-0.831559, -0.5, -0.33\}$[5], and $\kappa_{\text{actor}} \in \{-0.99, -0.5, 0, 0.5, 0.99\}$. The rationale for choosing more pessimistic (to semi-optimistic) κ_{critic} values is to counteract the overestimation bias. Given that USAC stems from the framework of SAC, SAC was included in our figures for comparison.

Table 2: Best κ_{critic} and κ_{actor} parameters for USAC.

Environment	κ_{critic}	κ_{actor}
`Ant-v4`	-0.831559	-0.99
`HalfCheetah-v4`	-0.33	-0.50
`Hopper-v4`	-0.831559	0.50
`Humanoid-v4`	-0.831559	$-0.831559^{\star}$
`Walker2d-v4`	-0.831559	0.0

By comparing the final returns (Figure 2) with the estimation error results (Figure 3) for the different grid combinations, we reaffirm previous findings [7, 10, 20]: i) overestimation (negative estimation error) should always be avoided as it reduces performance, and ii) underestimation (positive estimation error) does not necessarily harm performance. This indicates that a pessimistic approach should be adopted in critic training, whereas in actor training, either a pessimistic or optimistic approach can be suitable depending on the environment. The best combinations of κ_{critic} and κ_{actor}, with respect to final returns, are summarized in Table 2. Additionally, the area under the learning curve for the grid search parameters is illustrated in Figure 4 (Appendix C).

6 Discussion

We introduce Utility Soft Actor-Critic (USAC), a novel off-policy actor-critic algorithm. The key idea is to balance the trade-off between pessimism and optimism through incorporating a utility function that captures the uncertainty in the critic due to limited access to the environment. This framework is flexible and can be adapted to different distributions, with our experiments primarily focusing on the Laplace distribution. Unlike OAC [7] and TOP [20], which also explore optimism/pessimism in actor-critic training, USAC provides a policy improvement guarantee.

USAC allows independent control over the optimism and pessimism levels for the actor and critic. This decoupling enables a stable critic, necessary for accurate value function estimation, and an explorative actor, crucial for effective policy learning. We conducted a grid search experiment to explore these levels in training, revealing that being pessimistic in critic training while being either optimistic or pessimistic in actor training can be beneficial. However, it is also essential to consider the specific environment. Our work paves the way for further exploration in optimizing actor-critic algorithms with varying optimism and pessimism levels.

*For the `Humanoid-v4` environment, we found that the default parameter pair outperformed the best pair identified through grid search, e.g., see Figure 2. Therefore, we used the default pair for this environment in our results (Table 1). However, we conjecture that by expanding our (limited) grid search, a better pair may exist.

[5]Recall that $\kappa = -0.831559$ corresponds to $g(\kappa) = 1$, simplifying the utility function in (11) to the minimum of the two critics. Therefore, by setting $\kappa_{\text{critic}} \approx -0.83$, we align with the pessimistic choices of SAC [12], TD3 [10], and OAC [7].

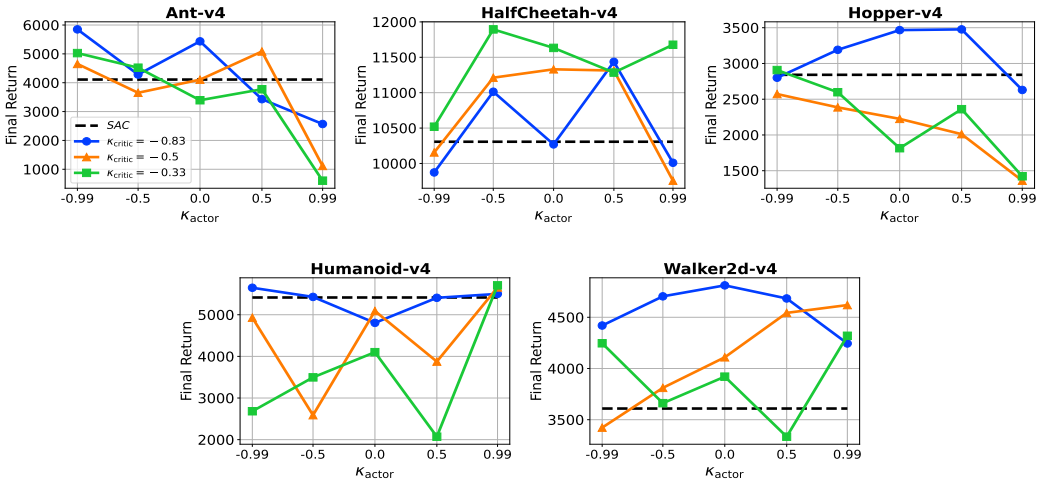

Figure 2: Grid search results for the final return of USAC algorithm with Laplace distribution, averaged over three seeds, each with three evaluation episodes. The grid search explore various κ_{critic} and κ_{actor} values, using SAC as the baseline.

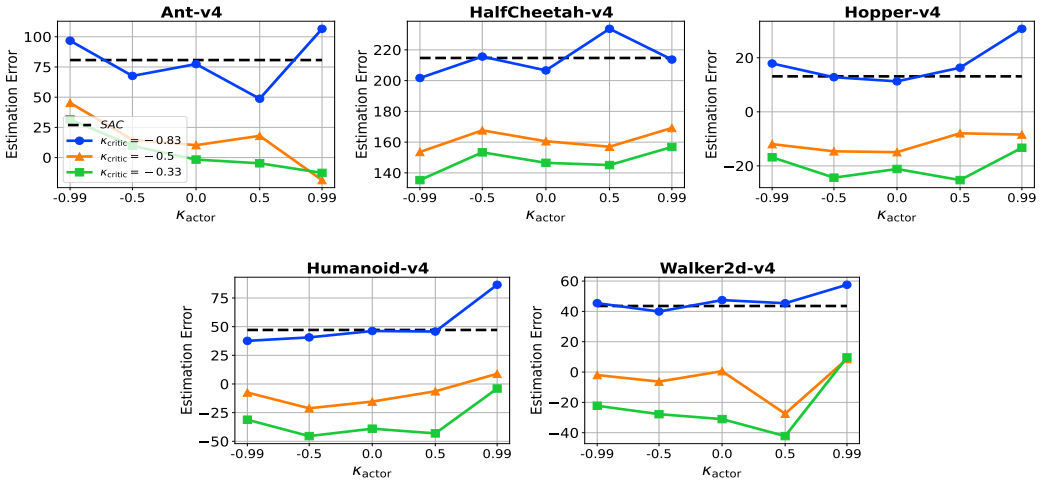

Figure 3: Grid search results for the estimation error (true - estimation) of USAC algorithm with Laplace distribution, averaged over three seeds, each with three evaluation episodes. The grid search explore various κ_{critic} and κ_{actor} values, using SAC as the baseline.

Limitations and future work USAC's primary limitation is its reliance on fixed optimism/pessimism parameters. An exciting direction for future research is to develop adaptive schemes where these parameters adjust dynamically throughout the learning process to better respond to the changing dynamics of the environment. There are several potential approaches to integrate automatic tuning. For example, these include: (i) adapting the automatic tuning scheme used for α in SAC [12, 13], (ii) employing bandit-based selection methods [20], or (iii) exploring continuous learning approaches using gradient descent [4].

Another promising direction for future work is to incorporate higher moments of the critic distributions. By considering not only the mean and variance but also skewness, kurtosis, and heavy-tailed distributions, we can better capture the nuances of uncertainty in the critic's value estimates. This approach could lead to more robust and flexible policy learning, accommodating a wider variety of distributional characteristics. Incorporating these higher-order moments may provide a richer framework for addressing the biases and variances in the value function estimation.

References

[1] M. G. Bellemare, W. Dabney, and R. Munos. A distributional perspective on reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

[2] M. G. Bellemare, W. Dabney, and M. Rowland. *Distributional reinforcement learning*. MIT Press, 2023.

[3] D. Bertsekas and J. N. Tsitsiklis. *Neuro-dynamic programming*. Athena Scientific, 1996.

[4] K. Bharadwaj and B. Ravindran. Continuous tactical optimism and pessimism. In *Third Conference on Deployable AI*, 2023.

[5] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. OpenAI Gym. *arXiv preprint arXiv:1606.01540*, 2016.

[6] S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 2012.

[7] K. Ciosek, Q. Vuong, R. Loftin, and K. Hofmann. Better exploration with optimistic actor critic. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[8] H. Föllmer and T. Knispel. Entropic risk measures: Coherence vs. convexity, model ambiguity and robust large deviations. *Stochastics and Dynamics*, 2011.

[9] H. Föllmer and A. Schied. *Stochastic finance: an introduction in discrete time*. Walter de Gruyter, 2011.

[10] S. Fujimoto, H. Hoof, and D. Meger. Addressing function approximation error in actor-critic methods. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

[11] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine. Reinforcement learning with deep energy-based policies. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

[12] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

[13] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

[14] R. A. Howard and J. E. Matheson. Risk-sensitive markov decision processes. *Management Science*, 1972.

[15] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[16] T. Lattimore and C. Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[17] T. Lillicrap, J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2016.

[18] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[19] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 2015.

[20] T. Moskovitz, J. Parker-Holder, A. Pacchiano, M. Arbel, and M. Jordan. Tactical optimism and pessimism for deep reinforcement learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[21] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[22] L. Prashanth, M. C. Fu, et al. Risk-sensitive reinforcement learning via policy gradient search. *Foundations and Trends® in Machine Learning*, 2022.

[23] M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[24] M. Rowland, R. Dadashi, S. Kumar, R. Munos, M. G. Bellemare, and W. Dabney. Statistics and samples in distributional reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2019.

[25] J. Schulman, X. Chen, and P. Abbeel. Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*, 2017.

[26] Y. Shen, M. J. Tobia, T. Sommer, and K. Obermayer. *Risk-sensitive reinforcement learning*. MIT Press, 2014.

[27] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[28] C. Szepesvári. *Algorithms for reinforcement learning*. Springer Nature, 2022.

[29] S. Thrun and A. Schwartz. Issues in using function approximation for reinforcement learning. In *Proceedings of the Fourth Connectionist Models Summer School*, 1993.

[30] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[31] H. Van Hasselt. Double q-learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2010.

[32] H. Van Hasselt, A. Guez, and D. Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016.

[33] H. Van Hasselt, Y. Doron, F. Strub, M. Hessel, N. Sonnerat, and J. Modayil. Deep reinforcement learning and the deadly triad. *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. Deep Reinforcement Learning Workshop.

[34] J. Von Neumann and O. Morgenstern. *Theory of games and economic behavior*. Princeton University Press, 1947.

[35] C. J. Watkins and P. Dayan. Q-learning. *Machine Learning*, 1992.

[36] B. D. Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University, 2010.

A Proofs

Proof of Proposition 1. Suppose $Q(s,a) \sim \text{Laplace}(\mu_{\mathcal{Q}}(s,a), b_{\mathcal{Q}}(s,a))$. Then, for any $(s,a) \in \mathcal{S} \times \mathcal{A}$, the expected utility function of $\mathcal{Q}$ on (s,a) can be expressed as

$$U_{\lambda}^{\mathcal{Q}}(s,a) = \mu_{\mathcal{Q}}(s,a) - \lambda^{-1} \log(1 - \lambda^2 b_{\mathcal{Q}}^2(s,a)),$$

with $|\lambda| < 1/b_{\mathcal{Q}}(s,a)$. Next, given that the variance of a Laplace distribution is $2b_{\mathcal{Q}}^2$, we can rewrite the expected utility function as

$$U_{\lambda}^{\mathcal{Q}}(s,a) = \mu_{\mathcal{Q}}(s,a) - \lambda^{-1} \log(1 - \lambda^2 \sigma_{\mathcal{Q}}^2(s,a)/2),$$

with $|\lambda| < \sqrt{2}/\sigma_{\mathcal{Q}}(s,a)$. Substituting $\lambda = \sqrt{2}\kappa/\sigma_{\mathcal{Q}}(s,a)$ for some $\kappa \in (-1,1)$, we have

$$U_{\kappa}^{\mathcal{Q}}(s,a) = \mu_{\mathcal{Q}}(s,a) - \sigma_{\mathcal{Q}}(s,a) \log(1 - \kappa^2)/\sqrt{2}\kappa.$$

At last, this expression can be further simplified using the definition of $g(\kappa)$ in (12). ■

B USAC under gaussianity

Proposition 2. *Suppose $\mathcal{Q}$ is Gaussian with mean $\mu_{\mathcal{Q}}$ and variance $\sigma_{\mathcal{Q}}^2$, i.e., $Q(s,a) \sim \mathcal{N}(\mu_{\mathcal{Q}}(s,a), \sigma_{\mathcal{Q}}^2(s,a))$. Then, for any $(s,a) \in \mathcal{S} \times \mathcal{A}$, the utility function of $\mathcal{Q}$ can be expressed as*

$$U_{\lambda}^{\mathcal{Q}}(s,a) = \mu_{\mathcal{Q}}(s,a) + \lambda \sigma_{\mathcal{Q}}^2(s,a)/2 \quad \textit{for} \quad \lambda \in (-\infty, \infty). \tag{16}$$

Proof of Proposition 2. The proof follows by similar steps as in the proof of Proposition 1. ■

Similarly, we can construct a USAC algorithm under Gaussian uncertainty, analogous to our approach under Laplace uncertainty as discussed in Section 4.3. By using the utility function defined in (16), we can incorporate specific λ_{critic} and λ_{actor} parameters for critic and actor training, respectively. This allows us to adjust the levels of optimism and pessimism tailored to the Gaussian-distributed $\mathcal{Q}$ values, thereby facilitating effective learning and exploration in our framework.

C Experiments details

Environments Our experiment is implemented in `PyTorch` [21, Version 2.1.0]. The experiments are conducted on 5 continuous control environments in the MuJoCo physics engine [5, 30]. More detailed information about these environments are provided in Table 3.

Table 3: Properties of MuJoCo environments

Environment	Action dimension	Observation dimension
`Ant-v4`	8	27
`HalfCheetah-v4`	6	17
`Hopper-v4`	3	11
`Humanoid-v4`	17	376
`Walker2d-v4`	6	17

Evaluation and seeds As mentioned in Table 4, we train our algorithms for one million time steps. Each experiment is repeated across 5 different seeds. We evaluate at every $10,000$ time step for ten evaluation episodes. For training, we use seeds 1 to 5, and for evaluation, we use seeds 101 to 105, which correspond to the training repetition number plus 100.

Table 4: Shared hyper-parameters

Hyper-parameter	Value
Evaluation episodes	10
Evaluation frequency	Maximum time steps / 100
Discount factor (γ)	0.99
n-step returns	1 step
Replay ratio	1
Replay buffer size	1,000,000
Maximum time steps	1,000,000
Mini-batch size (n)	256
Actor/critic optimizer	Adam [15]
Optimizer learning rates ($\eta_\phi, \eta_\theta, \eta_\alpha$)	3e-4
Averaging parameter (τ)	5e-3

Hyper-parameters The hyper-parameters and network configurations used in our experiments can be found in Table 4. For SAC and USAC, we learn the entropy temperature α during as outlined in Haarnoja et al. [13]. For OAC [7] and TOP [20], we adopt the best parameters reported in their respective papers.

Architectures and baselines The architectural specifications for all algorithms are presented in Table 5. Here, d_s and d_a represent the dimensions of the state and action spaces, respectively. The `SquashedGaussian` module features a Gaussian head, which uses the first d_a inputs for mean calculation and the subsequent d_a inputs for variance. This architecture is the standard one used in original implementations of SAC [12], TD3 [10], and OAC [7]. However, while TOP [20] states that it employs a network architecture with 2 hidden layers, its original implementation code actually considers 3 hidden layers.[6] This discrepancy prompted us to re-run TOP with a synchronized network such that it aligns with the other algorithms. In addition, the critic network in TOP follows the original implementation with a return shape that equals the number of quantiles (`N-Quantiles`), which is 50.

Table 5: Architecture details

Actor network		Critic network	
USAC SAC OAC	TD3 TOP	USAC SAC OAC TD3	TOP

`Linear(`d_s`, 256)`
`ReLU()`
`Linear(256, 256)`
`ReLU()`
		`Linear(`d_s `+` d_a`, 256)`
		`ReLU()`
		`Linear(256, 256)`
		`ReLU()`	

`Linear(256, 2×`d_a`, `θ`)` `SquashedGaussian(2×`d_a`, `d_a`)`	`Linear(256, `d_a`)` `Tanh()`	`Linear(256, 1)`	`Linear(256, N-Quantiles)`

Grid search of USAC In order to explore the dynamics of pessimism and optimism in USAC with Laplace distribution (Section 4.3), we conduct a grid search across five continuous control environments in MuJoCo (see e.g., environments specifications above). These grid search results are averaged over 3 seeds (seed 1 to 3), each with 3 evaluation episodes. This grid search investigated various values of κ_{critic} and κ_{actor}, using SAC as the baseline for comparison.

We examined critics ranging from pessimistic to semi-optimistic and actors from pessimistic to optimistic. Specifically, we used $\kappa_{\text{critic}} \in \{-0.831559, -0.5, -0.33\}$[7], and $\kappa_{\text{actor}} \in \{-0.99, -0.5, 0, 0.5, 0.99\}$. The rationale for choosing more pessimistic (to semi-optimistic) κ_{critic} values is to counteract the overestimation bias. Given that USAC stems from the framework of SAC, SAC was included in our figures for comparison.

In Figures 2, 3, and 4, we present the results of our grid search, showcasing the final returns, estimation error, and area under the learning curves for various parameter combinations. The estimation error calculated by subtracting the estimated Q-value from the true (discounted) reward. Thus, positive values indicate underestimating, while negative value indicate overestimation. These findings provide insights into the dynamics of pessimism and optimism in our USAC algorithm. As mentioned in Section 5, they reveal that: (i) overestimation (negative estimation error) should always be avoided as it reduces performance, and (ii) underestimation (positive estimation error) does not necessarily harm performance. This suggests that a pessimistic approach should be adopted in critic training, while either a pessimistic or optimistic approach can be suitable for actor training, depending on the environment. The best combinations of κ_{critic} and κ_{actor} in terms of final returns are summarized in Table 2. These optimal pairs highlight the importance of tailoring the levels of pessimism and optimism to the specific requirements of each environment to enhance overall performance.

Learning curve and estimation error curves The learning curve and estimation error curves of our USAC algorithm and the baseline algorithms can be seen in Figures 5 and 6, which correspond to the experiments summarized in Table 1. In addition, the estimation error values for these experiments are provided inTable 6. For USAC, the legend corresponds to the default case where $\kappa_{\text{critic}} = \kappa_{\text{actor}} =$

[6] We use the implementation provided by the authors for TOP, `https://github.com/tedmoskovitz/TOP`, where the `"number of random actions"` at the start of training updated to 10000.

[7] A $\kappa = -0.831559$ corresponds to $g(\kappa) = 1$, which means that the utility function in (11) simplifies into the minimum of the two critics. Hence, by setting $\kappa_{\text{critic}} \approx -0.83$, we align with the pessimistic choices of SAC [12], TD3 [10], and OAC [7].

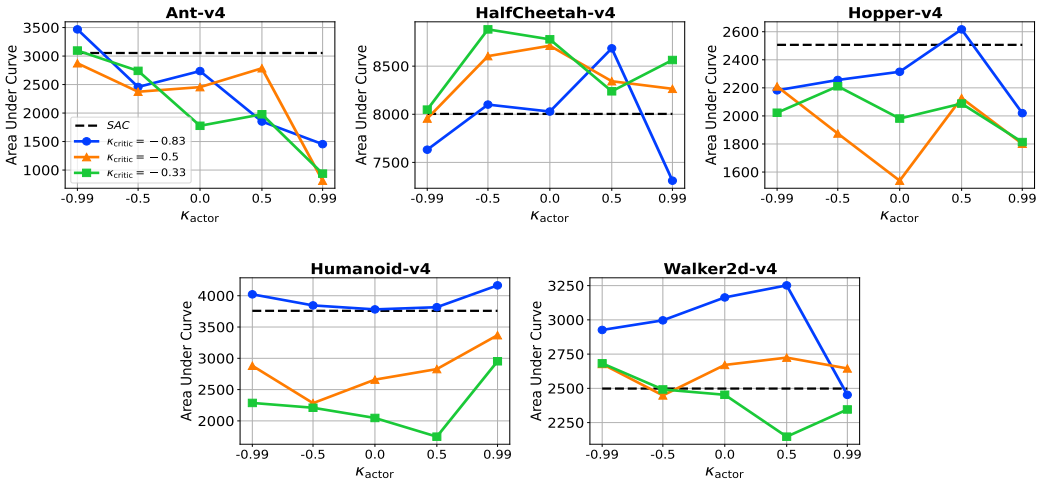

Figure 4: Grid search results for the area under the learning curve of USAC algorithm with Laplace distribution, averaged over three seeds, each with three evaluation episodes. The grid search explore various κ_{critic} and κ_{actor} values, using SAC as the baseline.

-0.831559. The USAC (best) legend uses the optimal $(\kappa_{\text{critic}}, \kappa_{\text{actor}})$ pairs (with respect to final return) as listed in Table 2.

Table 6: Estimation error of USAC and baselines.

Environment	USAC (**ours**)		SAC	TD3	OAC	TOP
	Default $(\kappa_{\text{critic}}, \kappa_{\text{actor}})$	Best $(\kappa_{\text{critic}}, \kappa_{\text{actor}})$				
`Ant-v4`	80.66	91.84	75.33	196.77	205.49	98.99
`HalfCheetah-v4`	221.21	146.07	222.93	216.54	190.28	81.97
`Hopper-v4`	12.46	18.51	12.73	24.82	25.05	11.73
`Humanoid-v4`	40.24	40.24*	42.78	31.89	52.66	24.02
`Walker2d-v4`	42.50	44.87	36.65	50.88	52.55	34.34

Computational cost Our USAC algorithm uses twin critics and a single actor network as for SAC, TD3, OAC, and TOP. This design choice eliminates the need for any additional computational steps and is no reliance on repetitive sampling, pre-training, or iterative approximation procedures. Consequently, it can be conclude that its computational characteristics closely match those of other widely used actor-critic algorithms and ours comparing baselines reported. The code for our algorithm implementation is also available alongside the paper.

*For the `Humanoid-v4` environment, we found that the default parameter pair outperformed the best pair identified through grid search, e.g., see Figure 2. Therefore, we used the default pair for this environment in our results (Table 1). However, we conjecture that by expanding our (limited) grid search, a better pair may exist.

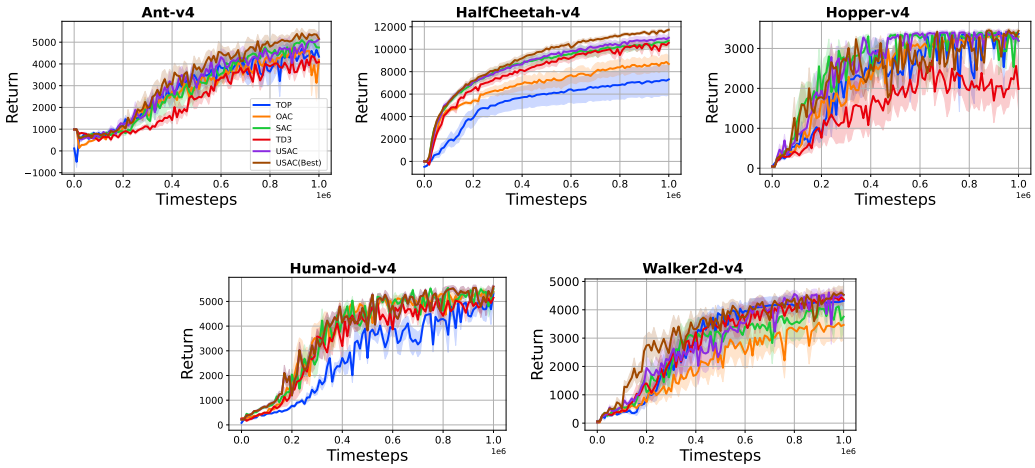

Figure 5: Learning curves of USAC and baselines. Solid curves depict the average return across evaluation episodes, while the shaded areas represent the standard deviation.

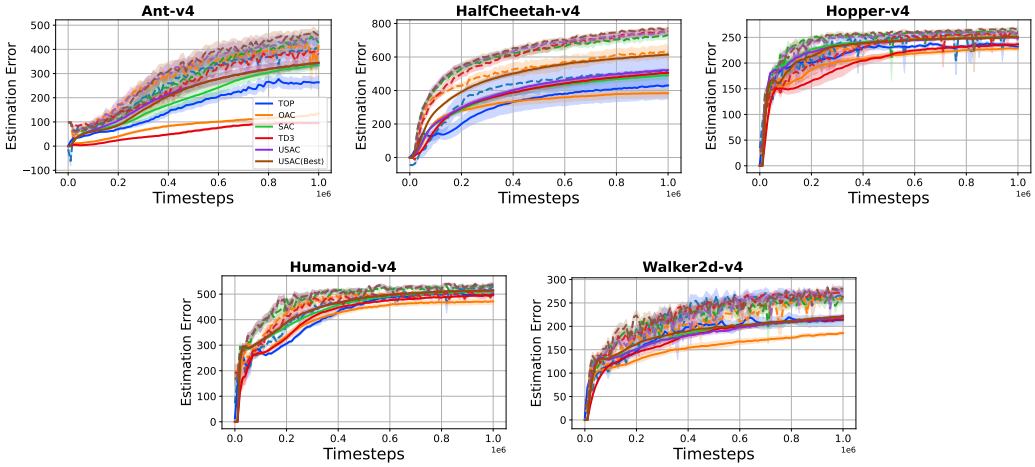

Figure 6: Estimation error curves of USAC and baselines. Solid lines indicate the estimated value function by the critic, while dashed lines show the averaged episodic discounted true return. The shaded areas around the solid and dashed lines represent the standard deviation from the average estimated value and the average true reward, respectively.