On the Theory of Risk-Aware Agents: Bridging Actor-Critic and Economics

Michal Nauman
Ideas NCBR & University of Warsaw
nauman.mic@gmail.com

Marek Cygan
Nomagic & University of Warsaw
ma.cygan@uw.edu.com

Abstract

Risk-aware Reinforcement Learning (RL) algorithms like SAC and TD3 were shown empirically to outperform their risk-neutral counterparts in a variety of continuous-action tasks. However, the theoretical basis for the pessimistic objectives these algorithms employ remains unestablished, raising questions about the specific class of policies they are implementing. In this work, we apply the expected utility hypothesis, a fundamental concept in economics, to illustrate that both risk-neutral and risk-aware RL goals can be interpreted through expected utility maximization using an exponential utility function. This approach reveals that risk-aware policies effectively maximize value certainty equivalent, aligning them with conventional decision theory principles. Furthermore, we propose Dual Actor-Critic (DAC). DAC is a risk-aware, model-free algorithm that features two distinct actor networks: a pessimistic actor for temporal-difference learning and an optimistic actor for exploration. Our evaluations of DAC across various locomotion and manipulation tasks demonstrate improvements in sample efficiency and final performance. Remarkably, DAC, while requiring significantly less computational resources, matches the performance of leading model-based methods in the complex dog and humanoid domains.

1 Introduction

Deep Reinforcement Learning (RL) is still in its infancy, with a variety of tasks unsolved (Sutton & Barto, 2018; Hafner et al., 2023) or solved within an unsatisfactory amount of environment interactions (Zawalski et al., 2022; Schwarzer et al., 2023). Whereas increasing the Replay Ratio (RR) (ie. the number of parameter updates per environment interactions step) is a promising general approach for increasing sample efficiency and final performance of RL agents (Janner et al., 2019; Chen et al., 2020; Nikishin et al., 2022), it is characterized by quickly diminishing gains (D'Oro et al., 2022) combined with linearly increasing computational cost (Rumelhart et al., 1986; Kingma & Ba, 2014). Moreover, the limitations of robot hardware and data acquisition frequency constrain the maximum achievable replay ratio (Smith et al., 2022). As such, it is worthwhile to pursue orthogonal techniques such as enhancing the properties of the underlying model-free agents. One continuously researched theme deals with finding risk attitudes that efficiently handle the *exploration-exploitation* dilemma (Ciosek et al., 2019; Moskovitz et al., 2021).

The attitudes towards risk of algorithmic agents have been researched in multiple contexts. For instance, a risk-seeking approach of optimism in the face of uncertainty has been identified as an effective exploration strategy, minimizing regret during the learning process (Wang et al., 2020b; Neu & Pike-Burke, 2020). Conversely, risk-averse, pessimistic Q-learning strategies have proven beneficial in counteracting value overestimation caused by temporal difference errors Hasselt (2010); Fujimoto et al. (2018). However, there is a disconnect between these risk-aware strategies and the foundational theories of RL, particularly in how they relate to the goal of value maximization. As a

38th Workshop on Aligning Reinforcement Learning Experimentalists and Theorists (ARLET 2024).

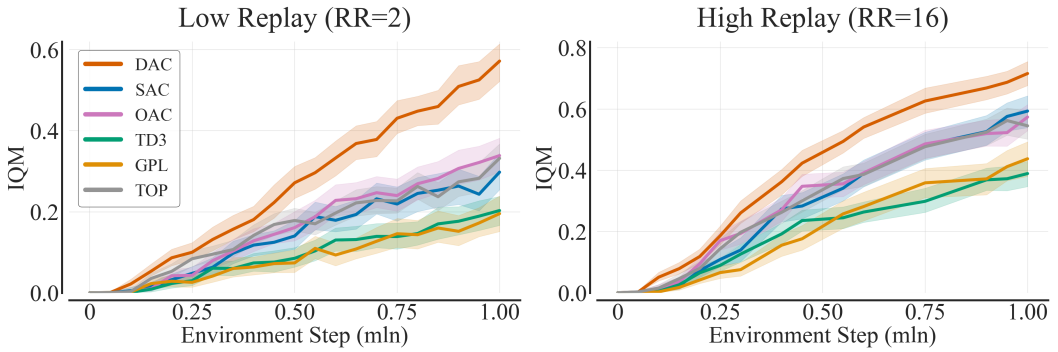


Figure 1: We test the proposed approach (DAC) against various risk-aware and risk-neutral actor-critic baselines in 30 tasks listed in Table 2. Due to a controlled experimental setup described in Section 5, the performance differences between algorithms stem solely from their risk-management. Y-axis reports IQM with 95% CI calculated using 10 seeds, with 1.0 representing the maximal score.

result, despite the empirical success of risk-aware agents like TD3 (Fujimoto et al., 2018) or SAC (Haarnoja et al., 2018a), the specific class of policies they implement remains ambiguous.

In this paper, we study the reinforcement learning objective aligned with the principles of decision theory. We show that, in contrast to pure value maximization, the decision theoretic perspective allows for derivation of both risk-neutral (e.g., DDPG (Silver et al., 2014)) and risk-aware approaches (e.g., TD3 (Fujimoto et al., 2018)). As such, we demonstrate that the pessimistic updates in state-of-the-art algorithms such as SAC (Haarnoja et al., 2018a), REDQ (Chen et al., 2020), or TOP Moskovitz et al. (2021) can be derived from expected utility maximization under an exponential utility function. Furthermore, we introduce Dual Actor-Critic (DAC), a risk-aware algorithm with two actors: optimistic and pessimistic. In DAC, each actor is trained independently using gradient backpropagation of a distinct objective: the optimistic actor aims to maximize an upper value bound for exploration, while the pessimistic actor focuses on a lower bound for temporal-difference (TD) targets (Fujimoto et al., 2018; Haarnoja et al., 2018a). DAC also features an automatic adjustment mechanism to balance the divergence between these actors, allowing for adaptability to various tasks without hyperparameter tuning. We evaluate DAC on a diverse set of locomotion and manipulation tasks and find that, despite its simplicity, DAC achieves significant improvements in sample efficiency and performance. Below, we outline our contributions:

1. We consider a generalized utility-based actor-critic objective, capable of formalizing both risk-neutral and risk-aware actor-critic algorithms. We demonstrate that the policies enacted by Clipped Double Q-Learning and generalizations thereof approximately align with the certainty equivalent of value under an exponential utility, and as such are optimal in the decision theoretic context.

2. We introduce Dual Actor-Critic (DAC), a risk-aware actor-critic framework featuring a dual actor network configuration. In DAC, each actor is trained through gradient backpropagation stemming from a specific objective that reflects different degrees of risk appetite. We establish the optimistic policy loss function and implement a system for online gradient-based adjustment of optimism hyperparameters. This feature enables DAC to effectively adapt to varying degrees of uncertainties, as well as different reward scales, without the need for manual hyperparameter tuning.

3. We show that DAC outperforms benchmark algorithms in terms of both sample efficiency and final performance. Notably, DAC solves the dog domain, reaching the performance of significantly more complex model-based methods. To facilitate further research, we perform ablations on various design and hyperparameter choices (over 5000 training runs). We release the implementation of DAC under the following URL: https://anonymous.4open.science/r/NeurIPS_DAC-D8B0/README.md.

2 Background

Reinforcement Learning We consider an infinite-horizon Markov Decision Process (MDP) (Puterman, 2014) which is described with a tuple (S, A, r, p, γ), where states S and actions A are continuous, $r_{s,a}$ is the transition reward, p is a deterministic transition mapping, with p_0 being the starting state distribution, and $\gamma \in (0, 1]$ is a discount factor. A policy $\pi(a|s)$ is a state-conditioned

action distribution with entropy denoted as $\mathcal{H}(\pi(s))$. Soft Value (Haarnoja et al., 2018a) is the sum of expected discounted return and state entropies from following the policy at a given state $V^\pi(s) = \mathrm{E}_{a\sim\pi}\left[r_{s,a} + \alpha\mathcal{H}(\pi(s)) + \gamma V^\pi(s')\right]$, with α denoting the entropy temperature parameter. Q-value is the expected discounted return from performing an action and following the policy thereafter $Q^\pi(s,a) = [r_{s,a} + \gamma V^\pi(s')]$. A policy is said to be optimal if it maximizes the expected value of the possible starting states s_0, such that $\pi^* = \arg\max_{\pi\in\Pi} \mathrm{E}_{s_0\sim p_0}\, V^\pi(s_0)$, with π^* denoting the optimal policy and Π denoting the considered set of policies (eg. Gaussian). Off-policy actor-critic algorithms perform gradient-based learning of both Q-values (ie. critic or value model) and the policy (ie. actor). The critic parameters θ are updated by minimizing SARSA temporal-difference loss $\mathcal{L}_\theta$ on transitions $T = (s, a, r, s')$ which are sampled from past experiences (Fujimoto et al., 2018; Haarnoja et al., 2018a) according to $\mathcal{L}_\theta = \mathrm{E}_{T\sim\mathcal{D}}\left(Q_\theta(s,a) - r_{s,a} - \gamma V^{\pi_\phi}(s')\right)^2$. Here, $Q_\theta(s,a)$ is the critic output, $V^\pi(s')$ is the bootstrap value derived from a target network, and $\mathcal{D}$ is the experience buffer (Mnih et al., 2015). The policy parameters ϕ are updated to maximize values approximated by the critic (Ciosek & Whiteson, 2020): $\mathcal{L}_\phi = \mathrm{E}_{s\sim\mathcal{D}}\, V^{\pi_\phi}(s)$. In this equation, π_ϕ represents the actor, and $V^{\pi_\phi}(s)$ is the state value under π_ϕ.

Risk-Awareness in Actor-Critic In standard actor-critic the critic represents the expected future discounted return and the actor is optimized with respect to such critic (Lillicrap et al., 2015). In contrast, popular algorithms like SAC or TD3 learn a critic that represents the lower bound of Q-values (Fujimoto et al., 2018; Haarnoja et al., 2018b). Employing the lower-bound has proven effective in mitigating value overestimation in Temporal Difference (TD) learning (Van Hasselt et al., 2016; Fujimoto et al., 2018). A popular approach is Clipped Double Q-Learning (CDQL), where the lower-bound is calculated by taking the minimum value of an ensemble of critics, most often two (Fujimoto et al., 2018; Haarnoja et al., 2018b; Ciosek et al., 2019; Hansen et al., 2022): $V^{\pi_\phi}(s) \approx \min(Q^1_\theta(s,a), Q^2_\theta(s,a)) - \log\pi_\phi(a|s)$, with $a \sim \pi_\phi(s)$. Here, $Q^1_\theta(s,a)$ and $Q^2_\theta(s,a)$ denote the first and the second critic in the ensemble, and $\log\pi_\phi$ is the state-action entropy. Furthermore, it was shown that using the minimum is equivalent to ensemble statistics (Ciosek et al., 2019):

$$\min\left(Q^1_\theta(s,a), Q^2_\theta(s,a)\right) = \underbrace{\frac{1}{2}\left(Q^1_\theta(s,a) + Q^2_\theta(s,a)\right)}_{\text{Mean}} - \underbrace{\frac{1}{2}\left|Q^1_\theta(s,a) - Q^2_\theta(s,a)\right|}_{\text{Standard Deviation}}. \tag{1}$$

This led to generalizations of the lower-bound as to include varying levels of pessimism (Ciosek et al., 2019; Moskovitz et al., 2021):

$$Q^\beta(s,a) = Q^\mu(s,a) + \beta Q^\sigma(s,a), \quad \text{and} \quad V^\beta(s) = V^\mu(s) + \beta V^\sigma(s) \tag{2}$$

Above, $Q^\beta(s,a)$ and $V^\beta(s)$ are the pessimistic Q-value and values respectively, β represents the degree of pessimism, and (μ, σ) denote the mean and standard deviation of the critic model ensemble. Furthermore, the pessimistic values and Q-values are related by $V^\beta(s) = \mathrm{E}_{a\sim\pi}(Q^\beta(s,a) - \log\pi_\theta(a|s))$. In this setup, the risk level of the lower bound is modulated by β. As shown in Equation 1, in case of $\beta = -1$ it is true that $Q^\beta = \min(Q^1_\theta(s,a), Q^2_\theta(s,a))$. In setup of pessimistic learning the value targets are affected by the uncertainty measured via the value model ensemble disagreement. A non-zero β indicates a departure from risk-neutrality: positive β yields a pessimistic lower bound, while negative β results in an optimistic upper bound. Thus, for given ϕ the difference between the risk-aware value (resulting from bootstrapping with Q^β), and risk-neutral values (resulting from bootstrapping with Q^μ) is proportional to the expected values of the discounted sum of critic disagreements (Fujimoto et al., 2018). As such, it is clear that iterating the pessimistic objective does not result in the optimal policy as long as the critic disagreement is not equal to zero for all state-actions (Kumar et al., 2020). Despite this, numerous effective off-policy algorithms adopt a risk-aware objective, leaning towards either pessimism or optimism (Fujimoto et al., 2018; Ciosek et al., 2019; Cetin & Celiktutan, 2023). This understanding of risk-awareness is slightly different to the risk-awareness considered in the distributional RL (Bellemare et al., 2017; Dabney et al., 2018b,a) or the CVaR series (La & Ghavamzadeh, 2013; Hiraoka et al., 2019; Lim & Malik, 2022). There, the distribution of returns is explicitly modelled via a distribution and the risk-awareness is calculated with respect to the aleatoric uncertainty (Dabney et al., 2018a; Lim & Malik, 2022). In contrast, CDQL and associated techniques use regular critics and consider the total estimation uncertainty associated with using point-estimate to model Q-values (Fujimoto et al., 2018; Hiraoka et al., 2021).

Expected Utility Theorem The Von Neumann–Morgenstern Theorem (Von Neumann & Morgenstern, 1947) posits that an agent whose preferences adhere to four axioms (completeness, transitivity, continuity, and independence), has a utility function $\mathcal{U}(x)$ that enables the comparison of the preferences. The expected utility hypothesis (Von Neumann & Morgenstern, 1947; Kahneman & Tversky, 1979) states that agents choose between risky options by comparing their expected utility. For example, given random variables X_i representing different propositions the agent ought to choose from, it follows that $X_1 \succeq X_2 \iff \mathrm{E}\, U(X_1) \geq \mathrm{E}\, U(X_2)$. In such setting, the goal of the agent is optimize the utility rather than its input (Stiglitz, 1997): $x^* = \arg\max_x \mathrm{E}_{x_i \sim X}\, \mathcal{U}(x_i)$. Here, risk stems from the potential decrease in utility due to uncertainty in its input space, making risk preference an attribute of the utility function under consideration. In particular, due to potential non-linearities in the utility there can be a discrepancy between $\arg\max \mathrm{E}\, U(x_i)$ and $\arg\max \mathrm{E}\, x_i$. Certainty equivalent of X, denoted as X_c, measures the impact of uncertainty on utility-optimal choices: $\mathcal{U}(X_c) = \mathrm{E}\, \mathcal{U}(X)$. Certainty equivalent presents a deterministic amount offering the same utility as a random event and varies based on risk preferences. Risk-averse utilities yield a certainty equivalent lower than the expected value, while risk-seeking leads to a higher certainty equivalent. The exponential utility, $\mathcal{U}(x, \beta) = e^{\beta x}$, is a simple model for risk-awareness, with β determining the risk preference. We show the basic risk-averse and risk-loving utilities in Figure 6.

3 Theory of Risk-Aware Actor-Critic

In RL, risks can be associated with any uncertainty present in the optimization of the learning objectives. In this manuscript we focus on risks associated with the uncertainty of value approximation. In particular, we focus on off-policy agents that perform updates according to the risk-aware pessimistic Q-values Q^β presented in Equation 2, such as SAC (Haarnoja et al., 2018a), TD3 (Fujimoto et al., 2018), OAC (Ciosek et al., 2019), TOP (Moskovitz et al., 2021) or GPL (Cetin & Celiktutan, 2023). Consequently, the policy derived from this approach deviates from the optimal policy stemming from pure value maximization, as indicated by the regret defined in the Background section. Notably, many state-of-the-art algorithms adopt non-risk-neutral strategies (Moskovitz et al., 2021; Hiraoka et al., 2021; D'Oro et al., 2022). However, the risk-aware correction, as outlined in Section 2, is not fully explicated by existing RL theory and does not emerge from pure value maximization problems. To this end, the efficacy of risk-aware agents is often attributed to two factors: risk-averse pessimism is justified by well-documented value overestimation in temporal learning (Hasselt, 2010; Fujimoto et al., 2018; Kumar et al., 2020), while risk-loving optimism is supported by lower regret guarantees for exploration Chen et al. (2017); Ciosek et al. (2019); Schrittwieser et al. (2020). In this section, we posit that both risk-aware and risk-neutral algorithms can be formalized as optimizing an expected utility objective. We consider a cycle of policy evaluation and improvement steps and analyze a single step thereof. In every step of iteration, the values are assumed to be samples from the distribution $\mathcal{V}(s)$, which is assumed to have finite moments and expected value denoted by V^μ, such that $V^\mu(s) = \mathrm{E}_{i \sim \mathcal{V}}\, V_i(s)$. Assuming that the expected value of $V_i(s)$ is an unbiased estimator of the on-policy value, the standard approach requires the policy to optimize for $V^\mu(s)$, such that $\pi^* = \arg\max_{\pi \in \Pi} \mathrm{E}_{s \sim \mathcal{D}} V^\mu(s)$. However, this is not the approach applied by the risk-aware algorithms, such as SAC or TD3. Given an invertible, increasing utility function $\mathcal{U}$, we define the *certainty equivalent value*, denoted as $V^c(s)$:

$$V^c(s) = \mathcal{I} \mathop{\mathrm{E}}_{i \sim \mathcal{V}} \mathcal{U}\, V_i(s) = V^\mu(s) + \Upsilon(s). \tag{3}$$

Above, the inverted utility function is denoted by $\mathcal{I} = \mathcal{U}^{-1}$ and $\Upsilon(s)$ denotes the risk premium. In this context, the certainty equivalent value represents the deterministic amount that amortizes the uncertainties associated with the value approximation stochasticity. As such, if the utility function implies risk-averse behaviour, then $\Upsilon(s) < 0$ resulting in certainty equivalent value that is smaller than $V^\mu(s)$. Building on the expected utility objective, we define the *certainty equivalence policy* that seeks to amortize the uncertainty associated with approximation of values:

$$\pi^c = \arg\max_{\pi \in \Pi} \mathop{\mathrm{E}}_{s \sim p} \mathop{\mathrm{E}}_{i \sim \mathcal{V}} \mathcal{U}\, V_i(s) = \arg\max_{\pi \in \Pi} \mathop{\mathrm{E}}_{s \sim p} V^c(s). \tag{4}$$

Above, π^c represents the policy that optimizes for the expected utility of values, which we term the *certainty equivalent policy*. As follows, the certainty equivalent policy is greedy with respect to the

certainty equivalent value and aligns with the expected utility hypothesis, wherein the agent seeks to maximize the expected utility of returns. This outcome contrasts with achieving the maximal value possible, as is typically sought in standard RL objectives. The certainty equivalent policy π^c optimizes for values that are not generally equal to the risk-neutral values. Specifically, the π^c and π^* are equivalent only iff $\Upsilon(s) = 0$, thus making $V^c(s) = V^\mu(s)$. This highlights the fact that the utility-based objective is a generalization of the traditional RL objective, with traditional RL objective being understood as utility maximization under a linear utility function. Hence, risk-neutral algorithms like DDPG can be interpreted as linear utility agents optimizing the certainty equivalent. Te explore scenarios where the utility is non-linear and diverges from the traditional objective, one has to evaluate the risk premium. Given that $\mathcal{U}$ is invertible, infinitely differentiable and its Taylor expansion is convergent, then the risk premium can be evaluated via Taylor series:

$$\Upsilon(s) \approx \mathcal{I} \mathop{\mathrm{E}}_{i\sim\mathcal{V}} \sum_{n=1}^{\infty} \frac{\mathcal{U}^n\big(V^\mu(s)\big)}{n!} \big(V_i(s) - V^\mu(s)\big)^n. \tag{5}$$

Above, we denote $\mathcal{U}^n(V^\mu(s))$ as the nth derivative of $\mathcal{U}$ calculated at $V^\mu(s)$. As such, the discrepancy between the certainty equivalent value and the risk-neutral value depends on moments of $\mathcal{V}(s)$. Now, we show that the risk-aware value $V^\beta(s)$ stems from an exponential utility.

Theorem 3.1. *Denote exponential utility $\mathcal{U}(V_i, \beta) = e^{2\beta V_i(s)}$, then the certainty equivalent value $V^c(s)$ is approximately equal to the pessimistic value $V^\beta(s)$:*

$$V^c(s) \approx \underbrace{\mathop{\mathrm{E}}_{i\sim\mathcal{V}} V_i(s)}_{\textit{Ensemble Mean}} + \beta \underbrace{\mathop{\mathrm{E}}_{i\sim\mathcal{V}} \big(V_i(s) - V^\mu(s)\big)^2}_{\textit{Ensemble Variance}}. \tag{6}$$

As detailed in Appendix A, we achieve this result by using convergent Taylor approximation, with the key step of approximating e^x with second-order series, which since $x = V_i(s) - V^\mu(s)$ is centered at 0. In consequence of the above Theorem, risk-aware agents like TD3 or SAC can be understood as optimizing for the utility objective, pursuing certainty equivalent rather than the expected values.

4 Dual Actor-Critic

In this section, we propose the Dual Actor-Critic (DAC). DAC is a risk-aware, off-policy algorithm that addresses the exploration and temporal-difference (TD) learning dichotomy in actor-critic algorithms. Usually, actor-critic algorithms use a single actor for both exploration (sampling actions for new transitions) and TD learning (calculating TD targets), which requires balancing between optimism for exploration (Wang et al., 2020b) and pessimism for avoiding value overestimation (Hasselt, 2010). DAC resolves this by employing optimistic and pessimistic actors, with each actor being updated to optimize the certainty equivalent value stemming from utilities with unique risk preferences. Following Soft Actor-Critic, DAC pursues the maximum entropy objective in a policy iteration performed on dataset of previous experiences Haarnoja et al. (2018a). Whereas Theorem 3.1 validates the use of risk-aware policies in terms of the optimality of the pursued solutions, we further develop DAC basing on the principles of SARSA, off-policy learning and Optimism in the Face of Uncertainty. Firstly, following SAC, we update the critic network according to a pessimistic value target sampled from the pessimistic actor. This design choice guarantees that the critic learns the pessimistic values under the pessimistic policy Van Seijen et al. (2009), while tackling critic value overestimation Fujimoto et al. (2018). Secondly, by performing off-policy value updates we allow for exploration via a different policy than the one used for value updates. In particular, we consider a policy that is optimistic and learns to perform actions that yield critic disagreement, thus tackling the issue of pessimistic underexploration Ciosek et al. (2019). In DAC, the optimistic actor is trained to maximize the upper Q-value bound and is solely used for exploration, while the pessimistic actor, guided by the lower Q-value bound, is used for TD learning and evaluation. This separation allows DAC to explore efficiently without the risk of value overestimation. DAC also addresses the shortcomings of Optimistic Actor-Critic (OAC) (Ciosek et al., 2019). By relaxing the first-order approximation and explicitly modeling the optimistic policy via a neural network DAC can approximate the maximum of arbitrary upper bound (Hornik et al., 1989). DAC adjusts the optimism

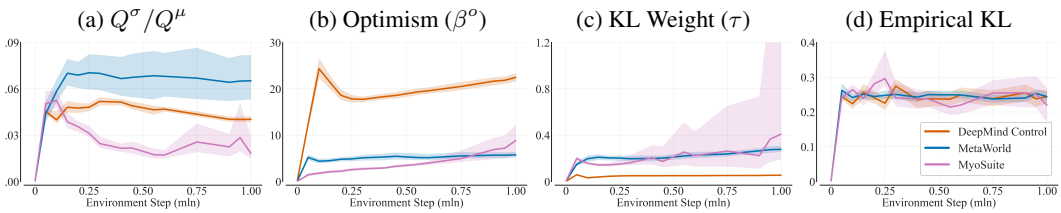


Figure 2: We report relative size of critic disagreement to critic means (2a), optimism (2b), KL penalty weight (2c), and the empirical KL between two actors (2d) during 1mln environment steps training on tasks listed in Table 2. DAC adjustment mechanism accommodates different scales of Q-values and allows for maintaining a predefined KL divergence between the two policies, despite significant differences between each task. 30 tasks, 10 seeds per task.

levels associated with the upper bound such that the two policies reach a predefined divergence target, thus alleviating the tandem problem induced by off-policy learning using two actors. DAC is structured around two components: a critic ensemble of k models (following the standard SAC/TD3 implementation we use $k = 2$ models) and the dual actor comprising pessimistic and optimistic actors denoted π^p_ϕ and π^o_η respectively. As shown in Theorem 3.1, each actor can be interpreted as optimizing exponential utility function with different levels of risk appetite, denoted as β^p and β^o for the pessimistic and optimistic actors.

Optimistic Actor The optimistic actor π^o_η, pursues an objective function that maximizes a utility expression that accounts for the divergence between the two actors:

$$\begin{aligned}\mathcal{L}_\eta &= -\mathop{\mathrm{E}}_{s\sim\mathcal{D}}\mathop{\mathrm{E}}_{i\sim\mathcal{V}} e^{\,2\beta^o(V_i(s)-\tau KL(\pi^p_\phi(s)|\bar{\pi}^o_\eta(s)))} \\ &\approx -\mathop{\mathrm{E}}_{s\sim\mathcal{D}}\big(Q^\mu_\theta(s,a)+\beta^o Q^\sigma_\theta(s,a)-\tau KL\big(\pi^p_\phi(s)|\bar{\pi}^o_\eta(s)\big)\big) \quad \text{with} \quad a\sim\pi^o_\eta(s).\end{aligned} \tag{7}$$

Above, KL represents the empirical Kullback-Leibler divergence between the actors, β^o is the optimism parameter, τ is the penalty weight associated with KL divergence, and $\bar{\pi}^o_\eta(s)$ is the transformed pessimistic policy. Incorporating two actors that are used for TD and exploration respectively, it is natural to assign distinct entropy levels to each policy. Our approach involves computing the KL divergence between the pessimistic policy π^p_ϕ and a modified optimistic policy, denoted as $\bar{\pi}^o_\eta$. This modified policy is characterized by a standard deviation that is m-times smaller than that of the optimistic policy π^o_η, with $m \in (0, \infty)$. We refer to m as exploration variance multiplier. We present ablations on different values of m in Figure 5, as well as tests regarding the importance of KL divergence in Table 1. We detail methodology for KL divergence calculation as well as technical details on implementing the optimistic actor via a neural network in Appendix C.

Pessimistic Actor and Critic The pessimistic actor is updated to maximize the utility:

$$\mathcal{L}_\phi = -\mathop{\mathrm{E}}_{s\sim\mathcal{D}}\mathop{\mathrm{E}}_{i\sim\mathcal{V}} e^{\,2\beta^p V_i(s)} \approx \mathop{\mathrm{E}}_{s\sim\mathcal{D}}\big(\alpha\log\pi^p_\phi(a|s)-Q^\mu_\theta(s,a)-\beta^p Q^\sigma_\theta(s,a)\big) \text{ with } a\sim\pi^p_\phi(s). \tag{8}$$

Above, $\mathcal{L}_\phi$ is the pessimistic actor loss, $\mathcal{D}$ is the experience buffer, α is the entropy temperature, β^p is the pessimism parameter, Q^μ_θ and Q^σ_θ denote the critic ensemble mean and standard deviation respectively. As such, the pessimistic actor convergences to the optimal policy under CDQL assumptions (Fujimoto et al., 2018). The critic parameters are updated via SARSA under the pessimistic policy:

$$\mathcal{L}_\theta = \mathop{\mathrm{E}}_{T\sim\mathcal{D}}\big(Q_\theta(s,a)-\big(r+\gamma Q^{\beta^p}_{\bar{\theta}}(s',a')-\alpha\log\pi^p_\phi(a'|s')\big)^2 \quad \text{with} \quad a'\sim\pi^p_\phi(s'). \tag{9}$$

Here, $\mathcal{L}_\theta$ is the critic loss function which is optimized off-policy on the dataset of previous experiences $\mathcal{D}$ with $T = (s, a, r, s')$, and $\bar{\theta}$ denotes the target network parameters which are updated via standard Polyak averaging (Fujimoto et al., 2018; Haarnoja et al., 2018a). This setup aligns the objectives of the critic and the pessimistic actor in DAC with the traditional pessimistic approaches like SAC.

Optimism and KL penalty adjustment The goal of KL regularization used by the optimistic actor is to achieve divergence small enough to ensure good coverage of the pessimistic policy in exploration data, yet large enough to maintain the optimistic policy's exploration effectiveness. In practice, the level of KL is influenced by three factors: the KL penalty weight τ, the difference in risk appetites between the actors $(\beta^o - \beta^p)$, and the critic ensemble disagreement $Q^\sigma_\theta(s,a)$, which varies based on reward scale and environment uncertainty. To achieve a balanced divergence between the two actors, we allow gradient-based adjustments in τ and $(\beta^o - \beta^p)$, while assuming a fixed β^p:

$$\mathcal{L}_{\beta^o} = \mathop{\mathrm{E}}_{s\sim\mathcal{D}}(\beta^o - \beta^p)D(s) \text{ for } \beta^o \in (\beta^p, \infty), \quad \text{and} \quad \mathcal{L}_\tau = \mathop{\mathrm{E}}_{s\sim\mathcal{D}} -\tau D(s) \text{ for } \tau \in (0, \infty). \tag{10}$$

Above, $\mathcal{L}_{\beta^o}$ and $\mathcal{L}_\tau$ represent the loss functions of the optimism and the KL penalty weight respectively. Where $D(s)$ represents the discrepancy between the recorded and target KL divergence:

$$D(s) = \frac{1}{|A|} KL\big(\pi^p_\phi(s) | \pi^o_\eta(s)\big) - \mathcal{KL}^*. \tag{11}$$

The mechanism adjusts β^o and τ based on the distance between the empirical and target KL divergences. When the empirical divergence exceeds the target, β^o decreases to a limit of β^p, and τ increases. Conversely, a smaller empirical divergence than the target leads to an increase in β^o and a decrease in τ. This dual adjustment allows DAC to regulate the divergence between two policy actors even when β^o reaches its lower bound. The optimization objectives for τ and β^o are thus formulated to adapt to varying conditions, ensuring efficient exploration and exploitation balance in different environments. We test the effectiveness of these adjustment mechanisms in Table 1, and depict the different levels of optimism achieved in various tasks in Figure 2. We expand our discussion of DAC in Appendix C, as well as present pseudocode in Figure 9.

5 Experiments

Our experimental framework is based on JaxRL (Kostrikov, 2021). We assess the performance of DAC across a diverse set of over 30 locomotion and manipulation tasks listed in Table 2, sourced from the DeepMind Control (DMC) (Tassa et al., 2018), MetaWorld (MW) (Yu et al., 2020) and MyoSuite (MYO) benchmarks. In DMC we report the returns, whereas in MW and MYO we report the success rates. We calculate robust evaluation statistics using RLiable (Agarwal et al., 2021).

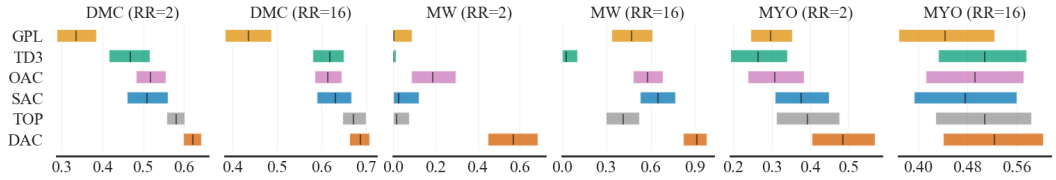


Figure 3: We report final IQM in 30 tasks (Table 2) in low ($RR = 2$) and high replay settings ($RR = 16$). 1.0 denotes the maximal possible score, 95% CI calculated with 10 random seeds.

Model-free benchmark We test DAC against an array of both risk-neutral and risk-aware baselines, including TD3 (Fujimoto et al., 2018), SAC (Haarnoja et al., 2018a), OAC (Ciosek et al., 2019), TOP (Moskovitz et al., 2021), and GPL (Cetin & Celiktutan, 2023). We align our experiments with the state-of-the-art SAC implementation, standardizing the common hyperparameters across all algorithms (D'Oro et al., 2022). We employ uniform network architectures and ensemble of two critics, as advocated in previous work (Fujimoto et al., 2018; Haarnoja et al., 2018a; Ciosek et al., 2019; Moskovitz et al., 2021; Cetin & Celiktutan, 2023). Importantly, by using uniform network architectures and hyperparameters, we ensure that the performance differences between algorithms stem solely from the different risk-preferences in tackling the exploration-exploitation dilemma. Each of 30 tasks is run for 1mln environment steps. We investigate two replay regimes: a compute-efficient regime, involving 2 gradient updates per environment step, and a sample-efficient regime, using 16 gradient updates per step with full-parameter resets every $160k$ environment steps D'Oro et al. (2022). As evidenced by Figure 1, DAC particularly excels in the earlier phases of the training, achieving

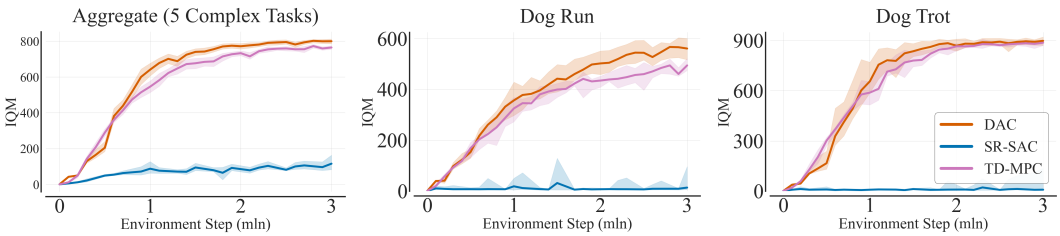


Figure 4: We compare model-free DAC, model-free SR-SAC to model-based TD-MPC. Y-axis reports IQM, and X-axis denotes environment steps. 3mln environment steps, 5 seeds.

final performance of the best baseline in only 60% of environment steps. Furthermore, as shown in Figure 3, DAC approach to balancing risk-averse exploitation with risk-loving exploration yields significant performance benefits, particularly visible in the low replay ratio setting.

Dog and Humanoid tasks Furthermore, we assess the performance of DAC when training for longer than 1mln environment steps. To this end, we choose 4 challenging locomotion tasks, two from the Dog domain (run, trot) and two from the Humanoid domain (run, walk), where we train for 3mln environment steps. The dog domain is particularly notable for its complexity, with only few methods showing non-random performance (Hansen et al., 2022; Xu et al., 2023). Here, we compare DAC against compute-heavy, sample-efficient baselines: SR-SAC with 32 gradient steps per environment step (D'Oro et al., 2022); and TD-MPC, a model-based approach that leverages a learned world model to perform trajectory optimization (Hansen et al., 2022). In particular, TD-MPC has been documented to achieve effective policies in both dog and humanoid domains (Hansen et al., 2022), however this performance comes at a significant compute cost, with TD-MPC using around 3 times more learnable parameters than DAC. To partially offset this disparity, we modified DAC critic to mimic the critic network used in TD-MPC, by adding a single hidden layer and layer normalization (Ball et al., 2023). As presented in Figure 4, despite being a model-free approach, DAC achieves competitive performance in the dog domain, marginally surpassing the model-based TD-MPC.

Design decisions We explore how DAC performance is influenced by variations in its design. We conduct evaluations of various DAC modifications in 15 tasks listed in Table 2, where we train for $500k$ environment steps and consider both replay ratios. We assess the performance of the following variations: ONLY π^o_η where we use the optimistic actor for both exploration and exploitation; NO KL REG where we do not use KL regularization for the optimistic actor; DETERMINISTIC π^o_η where we set the optimistic actor to be deterministic; DETERMINISTIC π^p_ϕ where we set the pessimistic actor to be deterministic; NO ADJUSTMENTS where we disable the DAC adjustment mechanisms; NO τ ADJUSTMENT where we disable τ adjustment; and NO β^o ADJUSTMENT where we disable β^o adjustment. As shown in Table 1, we find that all of these design choices bring significant gains to DAC performance. In particular, we observe that the dual policy setup allows for effective use of optimism, as evidenced by poor performance of the ONLY π^o_η agent. Similarly, we find that KL regularization yields significant improvements, most likely due to limiting the tandem problem.

Table 1: We evaluate the performance impact of DAC components. 15 tasks, $500k$ steps, 10 seeds.

	$RR = 2$	$RR = 16$
ONLY π^o_η	0.08	0.17
NO KL REG	0.42	0.77
DETERMINISTIC π^o_η	0.82	0.77
DETERMINISTIC π^p_ϕ	0.92	0.79
NO ADJUSTMENTS	0.91	0.87
NO τ ADJUSTMENT	0.95	0.93
NO β^o ADJUSTMENT	0.84	0.99
BASE DAC	1.00	1.00

Hyperparameter Sensitivity As discussed in Section 4, DAC introduces three new hyperparameters: KL divergence target, exploration multiplier, as well as the learning rates for the adjustment mechanism for β^o and τ. Here, we assess DAC sensitivity when changing the values of these hyperparameters. To this end, we train 12 DAC agents (each with a different hyperparameter setting) on $500k$ environment steps, in both replay ratio regimes, and on tasks listed in Table 2. We report these results In Figure 5, where we compare the final performance of these agents against SAC with

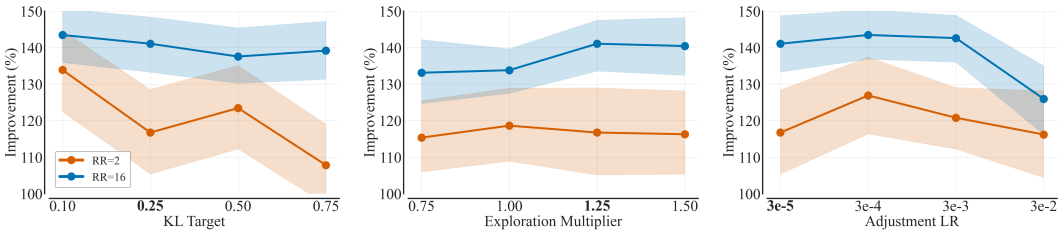


Figure 5: We evaluate DAC performance when changing the values of its hyperparameters (X-axis) for two replay ratio regimes. The bold value denotes the value used in the main experiment. Y-axis reports the percentage improvement over tuned SAC. 15 tasks, $500k$ steps, 10 seeds per task.

analogical replay ratio. As follows, we find that DAC performance is stable when changing the values of its hyperparameters, with all tested variations outperforming the tuned SAC baseline.

Additional Experiments In Appendix D, we detail supplementary experiments conducted to further evaluate DAC. Firstly, we assess the effectiveness of layer normalization when used with DAC. This regularization technique was shown to improve the performance of RL agents on a variety tasks (Hiraoka et al., 2021; Li et al., 2022; Lyle et al., 2023). We find that using layer normalization further improves DAC performance on locomotion tasks, and detail the results in Figure 11. Furthermore, we evaluate the compute costs associated with running a Dual Actor setup proposed by DAC. We find that DAC requires around 15% additional wallclock time to complete 1mln environment steps training as compared to SAC. However, as shown in Figure 10, DAC performance improvements offset the additional compute costs, leading to a beneficial trade-off between compute and performance. In Figure 12, we find that the dual architecture indeed mitigates the overestimation associated with an optimistic TD update. Finally, in Figures 13 & 14, we investigate the performance of distributional DAC that calculates upper-bound with respect to different types of uncertainties.

6 Limitations

Main limitation of DAC is its dual-actor framework, which incurs slightly higher memory and computation costs relative to the standard SAC. In our implementation, DAC wallclock is around 15% longer than SAC/TD3. Furthermore, using the extra actor network requires additional memory. In our implementation, the usage of optimistic actor translates to a 20% increased GPU memory dedicated to models as compared to SAC. Finally, DAC implementation requires setting three additional hyperparameters. Although tests showed DAC robust performance across a variety of hyperparameter values, it is uncertain whether this robustness translates to more complex environments.

7 Conclusions

In this work, we apply the expected utility theorem to reason about the empirical effectiveness of risk-aware actor-critic algorithms. In particular, we demonstrate that policies derived from the commonly used pessimistic objective are approximately utility-optimal when aligned with an exponential utility function. We think that the proposed framework is interesting as it revisits the foundational decision-theoretic perspective of RL — namely, optimization within the utility space rather than the space of utility inputs. Furthermore, we present DAC, a risk-aware actor-critic framework. DAC employs two actors with distinct risk appetites to optimize their respective expected utility functions. The pessimistic actor focuses on TD learning and evaluation, while the optimistic actor facilitates exploration. We evaluate DAC on various locomotion and manipulation tasks and compare it with over ten baseline algorithms. We find that DAC demonstrates significant performance improvements relative to other model-free methods and is competitive with leading model-based approaches. Finally, we investigate DAC performance robustness across diverse hyperparameter configurations and find that the experiments affirm its practical applicability.

Acknowledgments

We would like to thank Łukasz Kuciński for his help in developing the ideas presented in this paper. We also thank Piotr Miłoś and Gracjan Góral for their valuable help and discussions. We thank the Polish high-performance computing infrastructure PLGrid (HPC Center: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2023/016783. Marek Cygan was partially supported by an NCBiR grant POIR.01.01.01-00-0433/20.

References

Agarwal, R., Schwarzer, M., Castro, P. S., Courville, A. C., and Bellemare, M. Deep reinforcement learning at the edge of the statistical precipice. *Advances in neural information processing systems*, 34:29304–29320, 2021.

Auer, P., Cesa-Bianchi, N., and Fischer, P. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47:235–256, 2002.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Ball, P., Parker-Holder, J., Pacchiano, A., Choromanski, K., and Roberts, S. Ready policy one: World building through active learning. In *International Conference on Machine Learning*, pp. 591–601. PMLR, 2020.

Ball, P. J., Smith, L., Kostrikov, I., and Levine, S. Efficient online reinforcement learning with offline data. 2023.

Bellemare, M. G., Dabney, W., and Munos, R. A distributional perspective on reinforcement learning. In *International conference on machine learning*, pp. 449–458. PMLR, 2017.

Bhatt, A., Palenicek, D., Belousov, B., Argus, M., Amiranashvili, A., Brox, T., and Peters, J. Crossq: Batch normalization in deep reinforcement learning for greater sample efficiency and simplicity. *arXiv preprint arXiv: 1902.05605*, 2019.

Cetin, E. and Celiktutan, O. Learning pessimism for reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 6971–6979, 2023.

Chen, R. Y., Sidor, S., Abbeel, P., and Schulman, J. Ucb exploration via q-ensembles. *arXiv preprint arXiv:1706.01502*, 2017.

Chen, X., Wang, C., Zhou, Z., and Ross, K. W. Randomized ensembled double q-learning: Learning fast without a model. In *International Conference on Learning Representations*, 2020.

Chow, Y., Ghavamzadeh, M., Janson, L., and Pavone, M. Risk-constrained reinforcement learning with percentile risk criteria. *Journal of Machine Learning Research*, 18(167):1–51, 2018.

Chung, K.-J. and Sobel, M. J. Discounted mdp's: Distribution functions and exponential utility maximization. *SIAM journal on control and optimization*, 25(1):49–62, 1987.

Ciosek, K. and Whiteson, S. Expected policy gradients for reinforcement learning. *Journal of Machine Learning Research*, 21(2020), 2020.

Ciosek, K., Vuong, Q., Loftin, R., and Hofmann, K. Better exploration with optimistic actor critic. *Advances in Neural Information Processing Systems*, 32, 2019.

Dabney, W., Ostrovski, G., Silver, D., and Munos, R. Implicit quantile networks for distributional reinforcement learning. In *International conference on machine learning*, pp. 1096–1105. PMLR, 2018a.

Dabney, W., Rowland, M., Bellemare, M., and Munos, R. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018b.

D'Oro, P., Schwarzer, M., Nikishin, E., Bacon, P.-L., Bellemare, M. G., and Courville, A. Sample-efficient reinforcement learning by breaking the replay ratio barrier. In *The Eleventh International Conference on Learning Representations*, 2022.

Filippi, S., Cappé, O., and Garivier, A. Optimism in reinforcement learning and kullback-leibler divergence. In *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 115–122. IEEE, 2010.

Foster, D. J., Krishnamurthy, A., Simchi-Levi, D., and Xu, Y. Offline reinforcement learning: Fundamental barriers for value function approximation. In *Conference on Learning Theory*, pp. 3489–3489. PMLR, 2022.

Fujimoto, S., Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Garivier, A. and Moulines, E. On upper-confidence bound policies for switching bandit problems. In *International Conference on Algorithmic Learning Theory*, pp. 174–188. Springer, 2011.

Gogianu, F., Berariu, T., Rosca, M. C., Clopath, C., Busoniu, L., and Pascanu, R. Spectral normalisation for deep reinforcement learning: an optimisation perspective. In *International Conference on Machine Learning*, pp. 3734–3744. PMLR, 2021.

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

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

Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2019.

Hafner, D., Pasukonis, J., Ba, J., and Lillicrap, T. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Han, X., Chen, X., and Liu, L.-P. Gan ensemble for anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 4090–4097, 2021.

Hansen, N., Wang, X., and Su, H. Temporal difference learning for model predictive control. In *International Conference on Machine Learning, PMLR*, 2022.

Hasselt, H. Double q-learning. *Advances in neural information processing systems*, 23, 2010.

Hiraoka, T., Imagawa, T., Mori, T., Onishi, T., and Tsuruoka, Y. Learning robust options by conditional value at risk optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

Hiraoka, T., Imagawa, T., Hashimoto, T., Onishi, T., and Tsuruoka, Y. Dropout q-functions for doubly efficient reinforcement learning. In *International Conference on Learning Representations*, 2021.

Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Janner, M., Fu, J., Zhang, M., and Levine, S. When to trust your model: Model-based policy optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

Jaquette, S. C. A utility criterion for markov decision processes. *Management Science*, 23(1):43–49, 1976.

Kahneman, D. and Tversky, A. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–292, 1979.

Kaufmann, E., Cappé, O., and Garivier, A. On bayesian upper confidence bounds for bandit problems. In *Artificial intelligence and statistics*, pp. 592–600. PMLR, 2012.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kostrikov, I. JAXRL: Implementations of Reinforcement Learning algorithms in JAX, 10 2021. URL `https://github.com/ikostrikov/jaxrl`.

Kostrikov, I., Yarats, D., and Fergus, R. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020.

Krogh, A. and Hertz, J. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4, 1991.

Kumar, A., Gupta, A., and Levine, S. Discor: Corrective feedback in reinforcement learning via distribution correction. *Advances in Neural Information Processing Systems*, 33:18560–18572, 2020.

Kuznetsov, A., Shvechikov, P., Grishin, A., and Vetrov, D. Controlling overestimation bias with truncated mixture of continuous distributional quantile critics. In *International Conference on Machine Learning*, pp. 5556–5566. PMLR, 2020.

La, P. and Ghavamzadeh, M. Actor-critic algorithms for risk-sensitive mdps. *Advances in neural information processing systems*, 26, 2013.

Laskin, M., Lee, K., Stooke, A., Pinto, L., Abbeel, P., and Srinivas, A. Reinforcement learning with augmented data. *Advances in neural information processing systems*, 33:19884–19895, 2020.

Lee, K., Laskin, M., Srinivas, A., and Abbeel, P. Sunrise: A simple unified framework for ensemble learning in deep reinforcement learning. In *International Conference on Machine Learning*, pp. 6131–6141. PMLR, 2021.

Li, Q., Kumar, A., Kostrikov, I., and Levine, S. Efficient deep reinforcement learning requires regulating overfitting. In *The Eleventh International Conference on Learning Representations*, 2022.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Lim, S. H. and Malik, I. Distributional reinforcement learning for risk-sensitive policies. *Advances in Neural Information Processing Systems*, 35:30977–30989, 2022.

Liu, Z., Li, X., Kang, B., and Darrell, T. Regularization matters in policy optimization-an empirical study on continuous control. In *International Conference on Learning Representations*, 2020.

Lockwood, O. and Si, M. A review of uncertainty for deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 18, pp. 155–162, 2022.

Lyle, C., Zheng, Z., Nikishin, E., Pires, B. A., Pascanu, R., and Dabney, W. Understanding plasticity in neural networks. *arXiv preprint arXiv:2303.01486*, 2023.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Moskovitz, T., Parker-Holder, J., Pacchiano, A., Arbel, M., and Jordan, M. Tactical optimism and pessimism for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 34:12849–12863, 2021.

Neu, G. and Pike-Burke, C. A unifying view of optimism in episodic reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1392–1403, 2020.

Nikishin, E., Schwarzer, M., D'Oro, P., Bacon, P.-L., and Courville, A. The primacy bias in deep reinforcement learning. In *International conference on machine learning*, pp. 16828–16847. PMLR, 2022.

Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29, 2016.

Osband, I., Aslanides, J., and Cassirer, A. Randomized prior functions for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 31, 2018.

Pathak, D., Gandhi, D., and Gupta, A. Self-supervised exploration via disagreement. In *International conference on machine learning*, pp. 5062–5071. PMLR, 2019.

Protter, M. H., Charles Jr, B., et al. *A first course in real analysis*. Springer Science & Business Media, 2012.

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

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

Schäfer, L., Christianos, F., Hanna, J. P., and Albrecht, S. V. Decoupled reinforcement learning to stabilise intrinsically-motivated exploration. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pp. 1146–1154, 2022.

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

Schwarzer, M., Ceron, J. S. O., Courville, A., Bellemare, M. G., Agarwal, R., and Castro, P. S. Bigger, better, faster: Human-level atari with human-level efficiency. In *International Conference on Machine Learning*, pp. 30365–30380. PMLR, 2023.

Sekar, R., Rybkin, O., Daniilidis, K., Abbeel, P., Hafner, D., and Pathak, D. Planning to explore via self-supervised world models. In *International Conference on Machine Learning*, pp. 8583–8592. PMLR, 2020.

Seyde, T., Gilitschenski, I., Schwarting, W., Stellato, B., Riedmiller, M., Wulfmeier, M., and Rus, D. Is bang-bang control all you need? solving continuous control with bernoulli policies. *Advances in Neural Information Processing Systems*, 34:27209–27221, 2021.

Seyde, T., Schwarting, W., Karaman, S., and Rus, D. Learning to plan optimistically: Uncertainty-guided deep exploration via latent model ensembles. In *Conference on Robot Learning*, pp. 1156–1167. PMLR, 2022.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. Deterministic policy gradient algorithms. In *International conference on machine learning*, pp. 387–395. PMLR, 2014.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *nature*, 550 (7676):354–359, 2017.

Smith, L., Kostrikov, I., and Levine, S. A walk in the park: Learning to walk in 20 minutes with model-free reinforcement learning. *arXiv preprint arXiv:2208.07860*, 2022.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1): 1929–1958, 2014.

Stachowicz, K. and Levine, S. Racer: Epistemic risk-sensitive rl enables fast driving with fewer crashes. *arXiv preprint arXiv:2405.04714*, 2024.

Stiglitz, J. E. *Microeconomics*. New York, NY: WW Norton, 1997.

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

Tang, Y. C., Zhang, J., and Salakhutdinov, R. Worst cases policy gradients. *arXiv preprint arXiv:1911.03618*, 2019.

Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., Casas, D. d. L., Budden, D., Abdolmaleki, A., Merel, J., Lefrancq, A., et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

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

Van Seijen, H., Van Hasselt, H., Whiteson, S., and Wiering, M. A theoretical and empirical analysis of expected sarsa. In *2009 ieee symposium on adaptive dynamic programming and reinforcement learning*, pp. 177–184. IEEE, 2009.

Von Neumann, J. and Morgenstern, O. Theory of games and economic behavior, 2nd rev. 1947.

Wang, C., Wu, Y., Vuong, Q., and Ross, K. Striving for simplicity and performance in off-policy drl: Output normalization and non-uniform sampling. In *International Conference on Machine Learning*, pp. 10070–10080. PMLR, 2020a.

Wang, Y., Wang, R., Du, S. S., and Krishnamurthy, A. Optimism in reinforcement learning with generalized linear function approximation. In *International Conference on Learning Representations*, 2020b.

Xie, T., Foster, D. J., Bai, Y., Jiang, N., and Kakade, S. M. The role of coverage in online reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2022.

Xu, G., Zheng, R., Liang, Y., Wang, X., Yuan, Z., Ji, T., Luo, Y., Liu, X., Yuan, J., Hua, P., et al. Drm: Mastering visual reinforcement learning through dormant ratio minimization. *arXiv preprint arXiv:2310.19668*, 2023.

Yahaya, S. W., Lotfi, A., and Mahmud, M. A consensus novelty detection ensemble approach for anomaly detection in activities of daily living. *Applied Soft Computing*, 83:105613, 2019.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., and Levine, S. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020.

Zawalski, M., Tyrolski, M., Czechowski, K., Odrzygóźdź, T., Stachura, D., Piękos, P., Wu, Y., Kuciński, Ł., and Miłoś, P. Fast and precise: Adjusting planning horizon with adaptive subgoal search. In *The Eleventh International Conference on Learning Representations*, 2022.

Zhan, W., Huang, B., Huang, A., Jiang, N., and Lee, J. Offline reinforcement learning with realizability and single-policy concentrability. In *Conference on Learning Theory*, pp. 2730–2775. PMLR, 2022.

Broader Impact

In this paper, we analyze the pessimistic RL algorithms through the expected utility theorem and introduce a sample-efficient RL algorithm inspired by microeconomic principles. This approach to integrating behavioral theories into RL presents significant opportunities to improve agents. However, it also introduces a reverse application: using RL insights to inform microeconomic understanding and human behavior. This latter brings important ethical considerations. Applying economic theories to RL can enhance machine decision-making to more closely resemble human judgment. However, the consequences of machines operating in roles traditionally occupied by humans must be carefully analyzed. Similarly, applying RL principles to microeconomics offers perspectives on human decision-making and economic models, but should be done cautiously to avoid oversimplifying the human behavior and societal interactions.

A Derivations

We consider a single step of policy evaluation and improvement. We assume the value approximations $V^i(s)$ to be derived from a distribution $\mathcal{V}(s)$. This distribution is characterized by finite moments and an expected value $V^\mu(s)$:

$$V^\mu(s) = \mathop{\mathrm{E}}_{i\sim\mathcal{V}} V_i(s), \quad \text{and} \quad \epsilon_i(s) = V^i(s) - V^\mu(s). \tag{12}$$

Above, $\epsilon_i(s)$ is the deviation of value approximation from the mean, which per definition has expected value of zero (ie. $\mathrm{E}_{i\sim\mathcal{V}}\ \epsilon_i(s) = 0$). The certainty value is defined by:

$$V^c(s) = \mathcal{I}\mathop{\mathrm{E}}_{i\sim\mathcal{V}} \mathcal{U}\ V_i(s) = V^\mu(s) + \Upsilon(s). \tag{13}$$

Where $\Upsilon(s)$ is the risk premium. The pessimistic value is defined as:

$$V^\beta(s) = \mathop{\mathrm{E}}_{i\sim\mathcal{V}} V_i(s) + \beta \mathop{\mathrm{E}}_{i\sim\mathcal{V}} \big(V_i(s) - V^\mu(s)\big)^2 \tag{14}$$

Thus, the pessimistic value is equal to the certainty equivalent value if, for a given utility function, it is true that $\Upsilon(s) = \beta\ V^\sigma(s)$. The exponential utility function is defined as:

$$\mathcal{U}(V_i(s)) = e^{2\beta V_i(s)} \tag{15}$$

We start the derivation by applying the exponential utility to the certainty equivalent value definition:

$$\begin{aligned} e^{2\beta V^c(s)} &= \mathop{\mathrm{E}}_{i\sim\mathcal{V}} e^{2\beta V_i(s)} = \mathop{\mathrm{E}}_{i\sim\mathcal{V}} e^{2\beta(V^\mu(s)+\epsilon_i(s))} = \mathop{\mathrm{E}}_{i\sim\mathcal{V}} e^{2\beta V^\mu(s)} e^{2\beta\epsilon_i(s))} \\ &= e^{2\beta V^\mu(s)} \mathop{\mathrm{E}}_{i\sim\mathcal{V}} e^{2\beta\epsilon_i(s))} \end{aligned} \tag{16}$$

Inverting the utility function on the left-hand side (LHS) leads to:

$$V^c(s) = V^\mu(s) + \frac{1}{2\beta}\ln \mathop{\mathrm{E}}_{i\sim\mathcal{V}} e^{2\beta\epsilon_i(s)} \implies \Upsilon(s) = \frac{1}{2\beta}\ln \mathop{\mathrm{E}}_{s\sim p}\mathop{\mathrm{E}}_{i\sim\mathcal{V}} e^{2\beta\epsilon_i(s)} \tag{17}$$

Given that $\epsilon^i(s)$, by definition, centers around zero, we use the Maclaurin series which is convergent when applied to an exponential function. We use a second order approximation:

$$\Upsilon(s) \approx \frac{1}{2\beta}\ln \mathop{\mathrm{E}}_{i\sim\mathcal{V}} \left(1 + 2\beta\epsilon_i(s) + 2\beta^2\epsilon_i(s)^2\right) = \frac{1}{2\beta}\ln \mathop{\mathrm{E}}_{i\sim\mathcal{V}} \left(1 + 2\beta^2\epsilon_i(s)^2\right) \tag{18}$$

Here, $\epsilon^i(s)^2$ represents the variance of the model distribution at state s. Thus, for optimization purposes, the approximate equality $\nabla V^c(s) \approx \nabla V^\beta(s)$ is practically established. To calculate the

values, not just their gradients, we expand the logarithm function using first-order Maclaurin series. Again, the series are convergent since $1 + 2\beta^2\epsilon_i(s)^2 \geq 1$:

$$\Upsilon(s) \approx \frac{1}{2\beta} \mathop{\mathrm{E}}_{i\sim\mathcal{V}} 2\beta^2\epsilon_i(s)^2 = \beta \mathop{\mathrm{E}}_{i\sim\mathcal{V}} (V_i(s) - V^\mu(s))^2 \tag{19}$$

Which concludes the derivation. Thus, the pessimistic correction arises from the second-order Maclaurin approximation of the model risk's value risk premium. For completeness, we calculate the approximation error for the approximation of the exponential function, which we denote as $\mathcal{E}$:

$$\mathcal{E} = \sum_{n=3}^{\infty} \mathop{\mathrm{E}}_{i\sim\mathcal{V}} \frac{(2\beta\epsilon_i(s))^n}{n!} \tag{20}$$

Additional comment We note that the approach outlined in this Section can be adapted to address uncertainties inherent to a range of RL problems. An interesting example is the estimation of values through single evaluations of the critic. In such scenarios, adopting a risk-loving utility function can lead to objectives resembling Maximum Entropy (MaxEnt) principles. This suggests a potential avenue for integrating other risk preferences into actor-critic like setting. Moreover, the proof for the result, as detailed, can be considerably streamlined when focusing on gradients with respect to value functions in the context of policy optimization. Specifically, the approximation of the logarithm can be abandoned when the interest lies in optimizing policies rather than the values themselves.

B Related Work

B.1 Risk-Awareness Exploration

The exploration-exploitation dilemma has been the subject of extensive research. One prominent principle that has emerged in addressing this dilemma is Optimism in the Face of Uncertainty (OFU) (Auer et al., 2002; Filippi et al., 2010; Ciosek et al., 2019), which prioritizes actions with a balance of high expected rewards and uncertainty. Whereas OFU has been extensively studied in the tabular and bandit RL setting (Auer et al., 2002; Garivier & Moulines, 2011; Kaufmann et al., 2012), it has not yet become as standard in deep RL. However, it has been shown that DQN ensembles used for uncertainty-driven updates can provide performance improvements (Osband et al., 2016; Chen et al., 2017; Osband et al., 2018; Lee et al., 2021). Similarly, OAC (Ciosek et al., 2019) and TOP (Moskovitz et al., 2021) leverage uncertainty estimates over the state-action value function for exploration, albeit still using only pessimistic actor. On that note, Schäfer et al. (2022) propose to use two actors, with one actor using some form of novelty exploration objective.

The optimism-driven exploration was also considered for model-based agents. Sekar et al. (2020) and Seyde et al. (2022) consider exploration driven by reward model ensemble. Similarly to DAC, Seyde et al. (2022) considers using an optimistic upper bound exploration policy and a distinct exploitation policy. Furthermore, agents like RP1 (Ball et al., 2020) leverage reward uncertainty despite access to the nominal rewards. Finally, a variety of agents that leverage MCTS have been proposed (Silver et al., 2017; Schrittwieser et al., 2020; Zawalski et al., 2022).

B.2 Risk-Awareness for Overestimation

Overestimation of values is a long standing problem associated with Q-learning (Hasselt, 2010), which is particularly pronounced in the continuous-action deep RL setting (Fujimoto et al., 2018; Hiraoka et al., 2021; Cetin & Celiktutan, 2023). Undoubtedly, the most popular approach in dealing with overestimation is CDQL (Fujimoto et al., 2018; Haarnoja et al., 2018b; Ciosek et al., 2019; Bhatt et al., 2019; Hafner et al., 2019; Ciosek & Whiteson, 2020; Hiraoka et al., 2021; Hansen et al., 2022). However, it was noticed that overly pessimistic target can also negatively affect the agent performance (Moskovitz et al., 2021). We believe that this observation is particularly sensible in light of our proposed theory - since the agents are evaluated in expected return regime, training them with a distorted, risk-aware objective might not be optimal. To this end, a variety of works considered tuning the pessimism online (Moskovitz et al., 2021; Kuznetsov et al., 2020; Cetin & Celiktutan, 2023).

B.3 Risk-Awareness for Safety

On a different note, multiple works considered utility objectives for control that optimizes for worst-case (Jaquette, 1976; Chung & Sobel, 1987; Chow et al., 2018). In deep RL, there is a substantial body of work on utilizing distributional RL (Bellemare et al., 2017; Dabney et al., 2018b,a) for uncertainty estimation, which in turn is used for building risk-aware policies. For example, Dabney et al. (2018a) uses a quantile Q-network for estimation of the aleatoric uncertainty, and considers a variety of risk measures calculated over this distribution. Similarly, Stachowicz & Levine (2024) uses an ensemble of quantile networks to approximate the epistemic uncertainty and builds a CVaR policy that is sensitive with respect to that uncertainty. Usually, these works focus on optimizing some sort of "worst-case" objective (Tang et al., 2019). In contrast to these works, we focus on the generalizations of CDQL objective (Fujimoto et al., 2018), which is usually used by agents that are evaluated via risk-neutral maximum returns. In fact, our work focuses on showing that these objectives are in fact risk-aware.

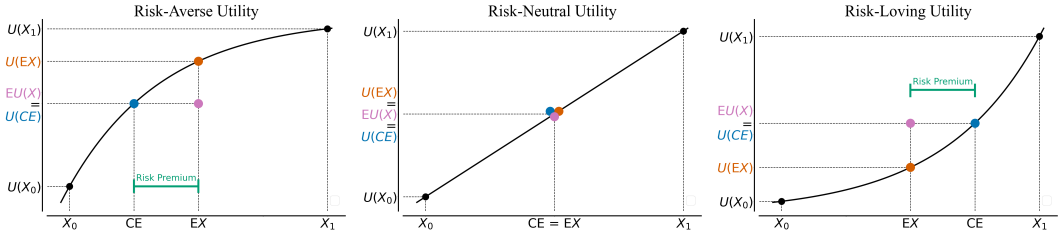


Figure 6: The risk appetite follows from the Jensen inequality: a convex utility indicates risk-seeking behavior, as evidenced by a CE exceeding $\mathrm{E}X$; conversely, a concave utility signifies risk-aversion, reflected in a CE that is smaller than $\mathrm{E}X$.

B.4 Comparison to OAC

DAC leverages two policies: a pessimistic one used for sampling the temporal-difference target and evaluation; and an optimistic one used for sampling transitions added to the experience buffer. Similarly to DAC, OAC performs evaluation and Bellman backups according to a pessimistic lower bound policy. However, DAC differs from OAC on three main design choices: *how to model the optimistic policy*; *how to constraint the optimistic policy*; and *how to set the level of optimism* β^o.

Firstly, OAC models the optimistic policy by combining pessimistic policy with the linear approximation of Q-value upper bound, and as such uses one actor network. The linear approximation combined with constrained optimization results in simplistic solutions along the constraint boundary (Protter et al., 2012). As such, OAC's applicability is limited to small δ values due to the Taylor theorem. In contrast to that, DAC uses two actors. Modeling the second policy via an actor network allows for exploration policies that are far more complex than a linear approximator. Whereas this introduces a computational cost, employing techniques like delayed policy updates can result in costs smaller than that of OAC.

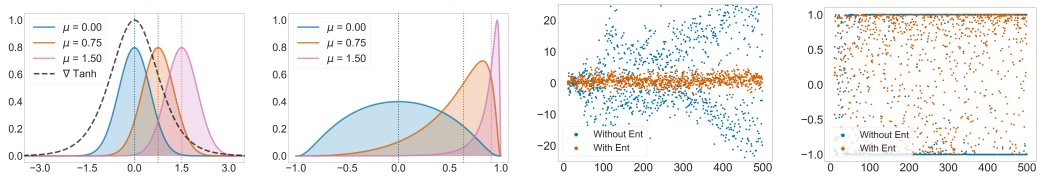
Furthermore, OAC enforces a hard KL constraint by directly solving a Lagrangian. Since the Q-value upper bound is approximated via a linear function, the solution is placed on the constraint boundary unless the slope is zero (Protter et al., 2012). In contrast, DAC imposes KL as a soft constraint. Paired with the neural network approximator, this allows DAC to balance the KL with potential gains to the upper bound and generate complex exploration policies.

Finally, OAC treats β^o as a hyperparameter which is fixed during the training. Since values of Q^μ and Q^σ depend on reward scales, as well as aleatoric and epistemic uncertainty of the environment, the value of β^o has to be searched per task. Furthermore, due to decreasing critic disagreement as the training progresses, fixed levels of β^o yield decreasing the impact of uncertainty on the optimistic policy. DAC leverages that the desired level of optimism can be defined through divergence between the pessimistic baseline policy and the optimistic policy optimizing objective related to β^o. Such definition allows for dynamics adjustment of both β^o and the KL penalty weight τ.

C Further Description of DAC

C.1 Pessimistic Actor

The pessimistic actor, denoted as π^p_ϕ, employs a soft policy target for optimization, as outlined in Equation 8. This actor operates entirely off-policy, utilizing data exclusively from the optimistic actor. The soft policy approach not only promotes state-dependent exploration (Haarnoja et al., 2018a) but also helps in maintaining the hyperbolic tangent (Tanh) output in a non-saturated state (Wang et al., 2020a). Moreover, the pessimistic actor's inherent non-zero variance contributes to stabilizing the TD learning for the critic. While separating exploration and exploitation theoretically allows for zero variance in TD target sampling, our empirical evidence suggests that introducing noise regularizes the critic (see Table 1).



(a) Normal (b) Tanh-Normal (c) Pre-Tanh μ (d) Post-Tanh Actions

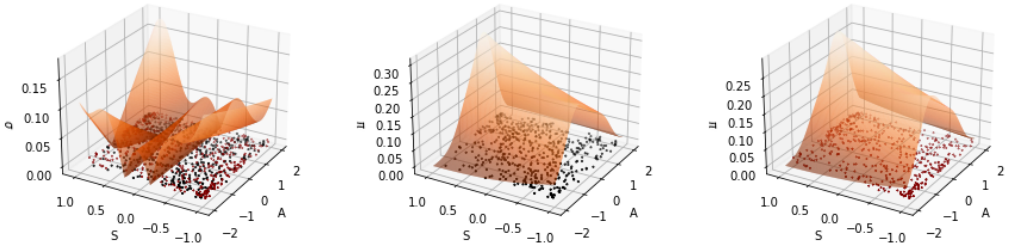
Figure 7: Soft policy learning prevents policy saturation. Figures 7a and 7b show three Gaussians and the Tanh transformed counterparts. Before Tanh transformation, all policies have equal variance (7a). However, since Tanh is nonlinear, applying the transformation changes the policy variance depending on the location of the mean (7b). Because of this effect, when using soft policy learning, the agent incurs loss when moving along the saturated portions of the Tanh activation function. Figures 7c and 7d show policy means and executed actions for SAC with and without soft policy learning on a Humanoid-Stand task. As follows, SAC without the entropy objectives follows a bang-bang policy, whereas regular SAC anchors the policy within the non-saturated portion of Tanh. To this end, soft policy learning reduces the risk of a bang-bang policy.

In the context of representing a policy with a Tanh-Normal distribution, soft policy learning serves two critical functions: promoting entropy and preventing saturation. Firstly, soft policy learning guarantees a certain degree of policy entropy, essential for exploration during the learning process (Haarnoja et al., 2018a). This approach contrasts with methods like TD3 or TOP, where constant entropy is maintained across various states. By optimizing an objective function that balances entropy maximization with Q-values, our policy exhibits variable entropy depending on the state, encouraging exploration particularly in states with lower Q-values. Furthermore, the Tanh function's inherent saturation characteristic impacts the exploration. The further the unsquashed distribution's mean is from zero, the more the variance reduces post-squashing, as depicted in Figures 7a and 7b. Implementing soft policy learning thus incurs a loss when the policy deviates from zero. Figures 7c and 7d illustrate how this approach results in policies predominantly residing within the non-saturated Tanh region (Wang et al., 2020a), effectively avoiding extreme bang-bang policies (Seyde et al., 2021).

C.2 Optimistic Actor

The optimistic actor π^o_η, follows the optimistic certainty equivalence objective, detailed in Equation 7. This objective prioritizes the upper bound of Q-values, fostering a policy that embraces uncertainty and encourages actions leading to critic disagreement. Such a policy approach is known to generate more diverse samples and consequently, ensures better coverage of the state-action space (Pathak et al., 2019; Lee et al., 2021). This is significant as ensemble disagreement is often a surrogate for sample novelty (Yahaya et al., 2019; Han et al., 2021). Although traditional RL does not explicitly focus on coverage, recent research highlights the critical role of data diversity in RL (Xie et al., 2022; Foster et al., 2022; Zhan et al., 2022).

In the development of DAC, our aim was to enforce entropy and non-saturation within both the pessimistic and optimistic actors. DAC accomplishes this through distinct yet complementary methods for each actor. For the pessimistic actor, soft policy learning is employed to enforce entropy



(a) Critic Variance (b) Pessimistic Actor (c) Optimistic Actor

Figure 8: Pessimistic underexploration and state-action space coverage on the Pendulum task with state representation embedded into 1 dimension. The dots represent 500 state-action samples gathered using the latest policy (pessimistic (black) or optimistic (red)). Figure 8a displays the standard deviation (σ) of the two critics, with smaller values observed in well-explored state-action regions. In Figure 8b, we depict pessimistic policy probabilities. Due to lower bound optimization, the actor prioritizes state-action subspaces that have already been explored and do not yield critic disagreement. Figure 8c illustrates optimistic policy probabilities. Despite having similar entropy levels, following the upper bound policy results in better coverage within critic disagreement regions.

and prevent policy saturation. In contrast, the optimistic actor uses KL divergence as a means of maintaining similarity with the pessimistic actor and adhering to certain policy constraints. The incorporation of KL divergence in the optimistic actor's objective serves multiple purposes:

1. Reducing the degree of off-policy learning - Given that exploration is conducted solely by the optimistic actor, the pessimistic actor updates entirely off-policy, using transitions sampled from the optimistic policy. This can potentially lead to unstable learning, a phenomenon known as the "deadly triad" (Sutton & Barto, 2018). Integrating KL divergence into the optimistic objective aligns the transitions sampled from the optimistic policy with the expectations under the pessimistic policy, thereby mitigating the extent of off-policy learning.
2. Anchoring the adjustment mechanisms - As discussed in Section 4, the automatic adjustment of parameters β^o (optimism) and τ (KL penalty weight) depends on a target divergence value. Without the divergence penalty, these adjustment mechanisms would have to be designed in a different way.
3. Preventing saturation of the optimistic policy - By minimizing the KL divergence between the optimistic and pessimistic actors, the optimistic actor is encouraged to emulate a policy trained via soft policy learning. This discourages the optimistic actor from policy saturation, maintaining the integrity of its actions.

In DAC, both the pessimistic and optimistic actors are represented as simple diagonal normal distributions, further transformed by the hyperbolic tangent (Tanh) activation. The KL divergence between these actors is computed in a closed form, utilizing the change of variables technique. Though this is a well-established result, we include it here for completeness. Denote x as samples from the policy distributions before the Tanh activation, p^p and p^o as the distributions of the pessimistic and optimistic actors on x, respectively. The application of Tanh is expressed as $y = h(x)$, resulting in the transformed distributions π^p, π^o. Using the change of variables formula, the KL divergence is derived as follows:

$$\begin{aligned} KL\big(p_c|p_o\big) &= \int_{-\infty}^{\infty} p^p(x) \log \frac{p^p(x)}{p^o(x)} dx \\ &= \int_{-\infty}^{\infty} \pi^p(h(x))\, |\frac{dy}{dx}(x)|\ \log \frac{\pi^p(h(x))\, |\frac{dy}{dx}(x)|}{\pi^o(h(x))\, |\frac{dy}{dx}(x)|} dx \\ &= \int_{-\infty}^{\infty} \pi^p(y) \log \frac{\pi^p(y)}{\pi^o(y} dy = KL\big(p^p|p^o\big) \end{aligned} \tag{21}$$

This relationship holds for any distributions p_c and p_o. With the assumption of diagonal Gaussian distributions with dimensionality $|A|$, the KL divergence simplifies to:

$$KL(p_p|p_o) = KL(\pi_c|\pi_o) = \sum_{i=1}^{|A|}\left(\log\frac{\sigma_i^p}{\sigma_i^o} + \frac{(\sigma_i^o)^2 + (\mu_i^o - \mu_i^p)^2}{2\,(\sigma_i^p)^2} - \frac{1}{2}\right) \tag{22}$$

Using KL guarantees that the optimistic policy optimizes for a specified level of variance, which can be distinct from π_ϕ^p. To this end, we define the exploration variance multiplier m, which is a simple standard deviation multiplier. This leads to the following formulation of KL applied to DAC:

$$KL(\pi_c|\pi_o) = \sum_{i=1}^{|A|}\left(\log\frac{\sigma_i^p}{\bar{\sigma}_i^o} + \frac{(\bar{\sigma}_i^o)^2 + (\mu_i^o - \mu_i^p)^2}{2\,(\sigma_i^p)^2} - \frac{1}{2}\right), \quad \bar{\sigma}_i^o = \frac{\sigma_i^o}{m} \tag{23}$$

This approach ensures distinct entropy levels for TD learning and exploration while maintaining the standard convergence properties of Actor-Critic (AC) algorithms. In fact, if $\lim_{\mathcal{D}\to\infty} Q_\theta^\sigma(s,a) = 0$ it follows that in the limit both actors recover a policy that differs only by m. While other differentiable divergence or distance functions could be used, the appeal of KL divergence lies in its closed-form solution for any invertible and differentiable transformation of the given distributions.

C.3 PseudoCode

Input: π_ϕ^p - pessimistic actor; π_η^o - optimistic actor; Q_θ - double critic; $Q_{\bar{\theta}}$ - target critic; α - temperature; β^o - optimism; τ - KL weight;
Hyperparameters: β^p - pessimism; $\mathcal{KL}^*$ - target KL;

Sample action from the optimistic actor
$s', r = \text{ENV.STEP}(a) \quad a \sim \pi_\eta^o$
Add transition to the replay buffer
$\text{BUFFER.ADD}(s, a, r, s')$
for $i = 1$ **to** ReplayRatio **do**
  Sample batch of transitions
  $s, a, r, s' \sim \text{BUFFER.SAMPLE}$
  Update critic using pessimistic actor (Eq. 9)
  $\theta \leftarrow \theta - \nabla_\theta\big(Q_\theta(s,a) - (r + \gamma(Q_{\bar{\theta}}^\mu(s',a') + \beta^p Q_{\bar{\theta}}^\sigma(s',a') - \alpha\log\pi_\phi^p(a'|s'))\big)^2,\ a' \sim \pi_\phi^p(s')$
  Update pessimistic actor (Eq. 8)
  $\phi \leftarrow \phi + \nabla_\phi\big(Q_\theta^\mu(s,a) + \beta^p Q_\theta^\sigma(s,a) - \alpha\log\pi_\phi^p(a|s)\big) \quad \text{with} \quad a \sim \pi_\phi^p(s)$
  Update optimistic actor (Eq. 7)
  $\eta \leftarrow \eta + \nabla_\eta\big(Q_\theta^\mu(s,a) + \beta^o Q_\theta^\sigma(s,a) - \tau KL\big(\pi_\phi^p(s)|\bar{\pi}_\eta^o(s)\big)\big) \quad \text{with} \quad a \sim \pi_\eta^o(s)$
  Update entropy temperature
  $\alpha \leftarrow \alpha - \nabla_\alpha\alpha\big(\mathcal{H}^* - \mathcal{H}(s)\big)$
  Update optimism (Eq. 10)
  $\beta^o \leftarrow \beta^o - \nabla_{\beta^o}(\beta^o - \beta^p)(\frac{1}{|A|}KL(\pi_\phi^p|\pi_\eta^o) - \mathcal{KL}^*)$
  Update KL weight (Eq. 10)
  $\tau \leftarrow \tau + \nabla_\tau\tau(\frac{1}{|A|}KL(\pi_\phi^p|\pi_\eta^o) - \mathcal{KL}^*)$
  Update target network
  $\bar{\theta} \leftarrow \text{POLYAK}(\theta, \bar{\theta})$
end for

Figure 9: Pseudo-code shows DAC training step, where changes with respect to SAC are colored.

C.4 Optimistic Actor Implementation

Both optimistic and pessimistic policies are modelled using the Tanh-transformed Gaussian distribution denoted as $\mathcal{T}(\mu, \sigma)$, where μ and σ denote the parameters of the underlying Gaussian

(Haarnoja et al., 2018a; Wang et al., 2020a). We implement the optimistic actor as a perturbation to the pessimistic policy parameters:

$$\pi^o_\eta(s) = \mathcal{T}((\mu^p_\phi(s) + \mu^o_\eta(s)), (\sigma^p_\phi(s) * \sigma^o_\eta(s))). \tag{24}$$

Above, $\mu^p_\phi(s)$ and $\sigma^p_\phi(s)$ denote mean and standard deviation given by the pessimistic actor network, $\mu^o_\eta(s)$ and $\sigma^o_\eta(s)$ denote the optimistic perturbation to the pessimistic policy. Such design allows the optimistic actor network to trivially reduce the KL divergence - to minimize divergence, the optimistic actor network needs to adjust its output towards 0, independently of the values of the pessimistic policy parameters. This design allows the model to keep the KL low, which as we show in Table 1 is crucial for good performance.

C.5 Adjustment of β^o and τ

Since values of Q^μ_θ and Q^σ_θ depend on reward scales, as well as aleatoric and epistemic uncertainty of the environment, the value of β^o cannot be easily set. DAC leverages an observation that for $\beta^o = -\beta^p$ the optimistic actor recovers the objective of the pessimistic actor. Then, β^o can be defined such that the divergence between the pessimistic baseline policy and the optimistic policy reaches a desired level, as shown in Equation 10. This adaptive approach, as illustrated in Figure 2, accommodates different scales of Q-values and contrasts with setups like OAC, where optimism is predefined by fixing β^o at a specific value. We find that the optimism and KL weight used by DAC are highly variable between tasks and even phase of the training, showing the advantage of the automatic adjustment mechanism as opposed to using predefined levels. Furthermore, we find that DAC uses higher levels of optimism than values that were found optimal for OAC (Ciosek et al., 2019).

D Additional Experiments

D.1 Wallclock performance

Firstly, we assess the wallclock efficiency of DAC as compared to SAC. Figure below investigates the performance of DAC and SAC in both low and high replay regimes.

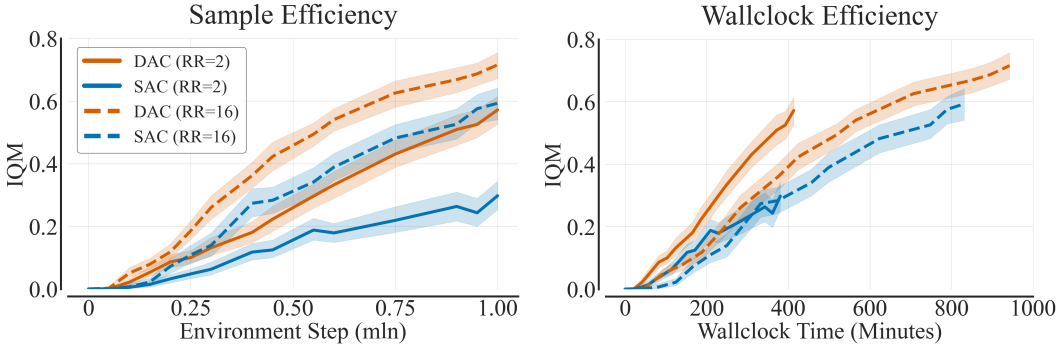


Figure 10: DAC wall-clock runtime for 1mln steps is around 15% longer than SAC. Despite additional computational requirements associated with the dual actor, the improvements stemming from DAC lead to a beneficial compute/performance tradeoff. The Y-axis denotes IQM with 95% CI. The X-axis denotes the environment steps (left) and wallclock (right) for 10 seeds. 30 tasks, 1mln steps and 10 seeds. Experiments were conducted on an NVIDIA A100 GPU with 40GB of RAM.

D.2 Layer Normalization

Recently, there has been a surge of works exploring the importance of critic regularization in high replay regimes (Liu et al., 2020; Laskin et al., 2020; Hiraoka et al., 2021; Gogianu et al., 2021; Li et al., 2022; D'Oro et al., 2022). Whereas there is still a lot to understand about the interplay of TD

learning and network regularization, it is clear that a regularized critic allows a higher replay ratio to be used (Hiraoka et al., 2021; Li et al., 2022; D'Oro et al., 2022). In this paper, we explore only one method that could be considered regularization, that is full-parameter resets (Nikishin et al., 2022; D'Oro et al., 2022). Given the effectiveness of the only regularization we tested, we hypothesize that experimenting with methods like layer normalization (Ba et al., 2016), weight decay (Krogh & Hertz, 1991), spectral normalization (Gogianu et al., 2021) or dropout (Srivastava et al., 2014) could further improve DAC performance.

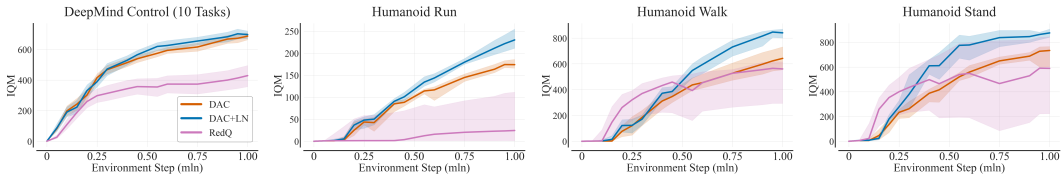


Figure 11: We compare high-replay DAC performance with and without layer normalization, as well as the ensemble based, high-replay RedQ (Hiraoka et al., 2021). We use 10 DMC tasks listed in Table 2 and run 10 seeds for 1mln environment steps.

As shown in Figure 11, we find that using layer normalization on critic within the DAC design significantly improves its performance on locomotion tasks. This results is in line with previous work that augmented other RL algorithms with layer normalization (Li et al., 2022; Hiraoka et al., 2021; Lyle et al., 2023; Ball et al., 2023; Cetin & Celiktutan, 2023)

D.3 Critic Overestimation

Here, we investigate whether the decoupled architecture (using a conservative policy for TD updates and optimistic policy for exploration) indeed mitigates Q-value overestimation characteristic for non-conservative updates (Fujimoto et al., 2018; Kuznetsov et al., 2020; Moskovitz et al., 2021; Cetin & Celiktutan, 2023). As such, we compare the environment returns to the returns implied by the critic, according to the following:

$$O_\theta = \frac{|Q_\theta(s,a)|}{|G^\pi(s,a)|} \tag{25}$$

Where O_θ denotes the overestimation metric, $Q_\theta(s,a)$ is the critic output and $G^\pi(s,a)$ stands for the observed empirical returns. Whereas the metric does not perfectly measure the overestimation, it allows to investigate the relative overestimation in the group of considered algorithms. As shown in Figure 12, we find that the dual architecture prevents the overestimation associated with optimistic TD updates and yields measurements similar to other risk-aware algorithms.

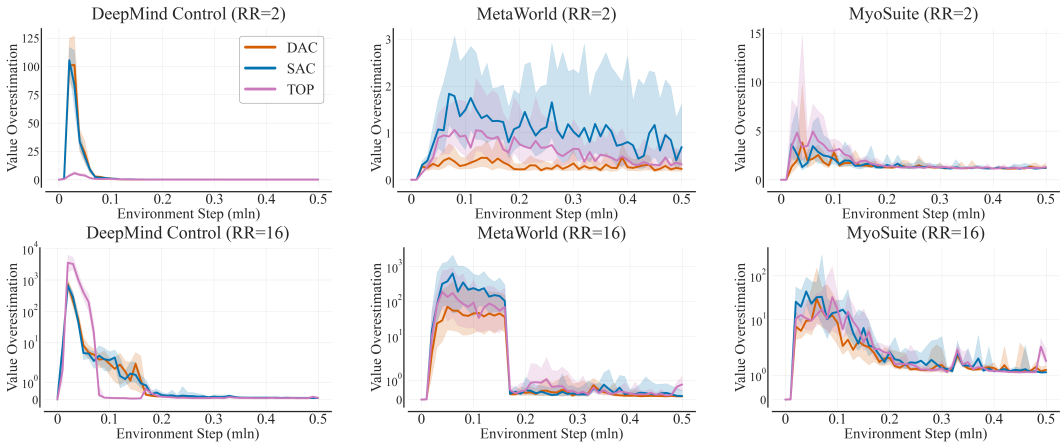


Figure 12: We investigate overestimation in DAC, SAC and TOP. We use 15 tasks listed in Table 2 and run 10 seeds for 500k environment steps.

D.4 Distributional DAC

As discussed in Section 4, our algorithmic implementation follows SAC/TD3 in that it uses a non-distributional double critic, where each critic is taught to output the expected value of returns (Haarnoja et al., 2018a; Fujimoto et al., 2018; Chen et al., 2020; Kostrikov et al., 2020). As such, the critic ensemble disagreement measures the entire uncertainty associated with Q-value approximation. Alternatively, the critic can be implemented to learn the full distribution of returns in an approach referred to as distributional RL Bellemare et al. (2017); Dabney et al. (2018b,a); Moskovitz et al. (2021). In this section, we investigate DAC implementation where the exploration policy is optimistic with respect to aleatoric, epistemic or total uncertainty. Following TOP (Moskovitz et al., 2021) we implement DAC with double quantile distributional critics, where each critic outputs K quantiles of the return distribution (Dabney et al., 2018a). In such setup, the kth quantile of the distribution at state-action (s, a), denoted as $Q^{\pi}_{k}(s, a)$, is learned via optimization of Huber loss of the distributional Bellman error Dabney et al. (2018a) defined as $\delta^{\pi}_{k} = Z^{\pi}(s, a) - Q^{\pi}_{k}(s, a)$ with $Z^{\pi}(s, a)$ representing a sample from the return distribution. We note, that following the Maximum Entropy formulation (Haarnoja et al., 2018a), we define the returns $Z^{\pi}(s, a)$ as a sum of discounted rewards and policy entropy. Furthermore, the point estimate of Q-value is calculated by $Q^{\pi}(s, a) = \frac{1}{K}\sum_{k=1}^{K} Q^{\pi}_{k}(s, a)$. Then, an ensemble of quantile critics can be used to explicitly model both aleatoric and epistemic uncertainties (Moskovitz et al., 2021; Lockwood & Si, 2022). To model aleatoric and epistemic uncertainties, we use the procedure outlined in TOP (Moskovitz et al., 2021). As such, we model aleatoric uncertainty by learning the quantiles of the state-action return distribution, which we denote as $\mathcal{Z}^{\pi}(s, a)$. Thus, we define Q-value as $Q^{\pi} = \mathrm{E}\, Z^{\pi}(s, a)$ with $Z^{\pi}(s, a) \sim \mathcal{Z}^{\pi}(s, a)$. The spread of the learned distribution $\mathcal{Z}^{\pi}(s, a)$ models the aleatoric uncertainty associated with the policy and the environment. Similarly, we model epistemic uncertainty by the ensemble disagreement when approximating a particular quantile of the return distribution. We consider three implementations of the optimistic policy in a distributional quantile DAC. All approaches apply the optimistic policy optimization as defined in Equation 7, but differ in how the point estimate of Q-value upper-bound is calculated:

Optimistic policy wrt. epistemic uncertainty - In this setup, for a fixed policy π, we define the Q-value upper bound as $Q^{\beta^o}(s, a) = \frac{1}{K}\sum_{k=1}^{K} Q^{o}_{k}(s, a)$ and $Q^{o}_{k}(s, a) = Q^{\mu}_{k}(s, a) + \beta^o Q^{\sigma}_{k}(s, a)$, where Q^{μ}_{k} and Q^{σ} represent the ensemble mean and standard deviation in kth quantile. As such, the optimistic policy promotes actions with high epistemic uncertainty measured as critic ensemble disagreement in approximation of the kth quantile.

Optimistic policy wrt. aleatoric uncertainty - Here, for a given policy, we calculate the Q-value upper bound as $Q^{\beta^o}(s, a) = \frac{1}{K}\sum_{k=1}^{K} Q^{\mu}_{k}(s, a) + \beta^o \sigma_Z(s, a)$ where $\sigma_Z(s, a)$ is the standard deviation of K quantile outputs, averaged over the critic ensemble. As such, the optimistic policy promotes actions with high aleatoric uncertainty measured as the spread of the return distribution.

Optimistic policy wrt. both uncertainties - Finally, we consider an optimistic policy that promotes both aleatoric and epistemic uncertainties, albeit still explicitly modeled via an ensemble of quantile critics. Here, for a given policy we calculate the upper-bound Q-value by $Q^{\beta^o}(s, a) = \frac{1}{K}\sum_{k=1}^{K} Q^{o}_{k}(s, a) + \beta^o \sigma_Z(s, a)$, with $Q^{o}_{k}(s, a) = Q^{\mu}_{k}(s, a) + \beta^o Q^{\sigma}_{k}(s, a)$. As such, the optimistic policy promotes actions that are associated with both epistemic and aleatoric uncertainty.

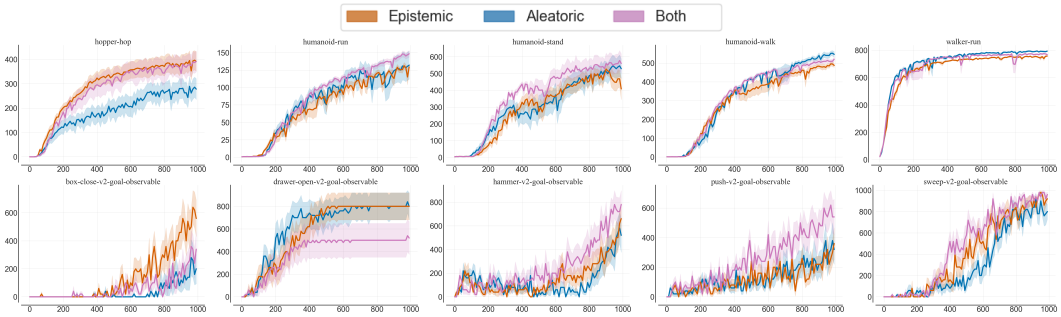


Figure 13: We run three variations of distributional DAC ($RR = 2$) on 10 tasks from DMC and MW benchmarks. Y-axis reports IQM and X-axis reports environment steps. 10 seeds per task.

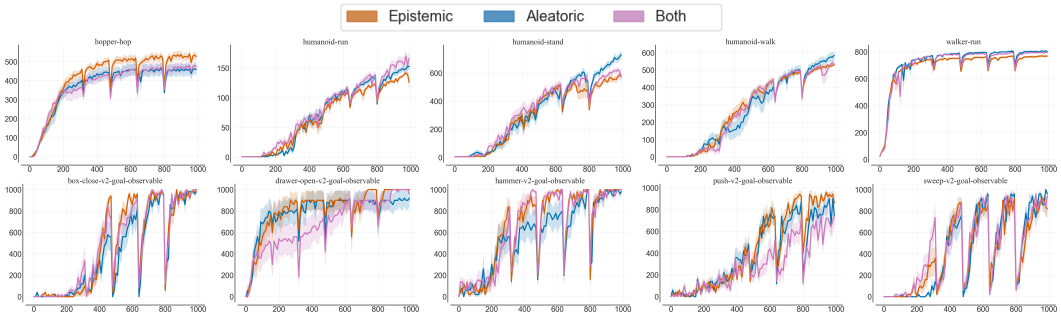


Figure 14: We run three variations of distributional DAC ($RR = 16$) on 10 tasks from DMC and MW benchmarks. Y-axis reports IQM and X-axis reports environment steps. 10 seeds per task.

We run the three variations of Distributional Quantile DAC for 1mln environment steps and both replay regimes. The results are presented in the Figures below. We find that all variations achieve similar performance, with the standard implementation promoting the epistemic uncertainty slightly outperforming other variations.

E Experimental Details

E.1 Environments

The table below lists task used in the main experiment.

Table 2: 30 tasks used for the evaluation of model-free algorithms in low and high replay ration regimes. The 15 bold tasks are used for the hyperparameter sensitivity and design choices ablation.

DeepMind Control	MetaWorld	MyoSuite
Acrobot Swingup	Assembly	**Reach Easy**
Cheetah Run	**Box Close**	Reach Hard
Hopper Hop	Coffee Pull	**Pose Easy**
Humanoid Run	**Drawer Open**	Pose Hard
Humanoid Stand	**Hammer**	**Pen Twirl Easy**
Humanoid Walk	Lever Pull	Pen Twirl Hard
Swimmer Swimmer6	**Push**	**Object Hold Easy**
Pendulum Swingup	Stick Pull	Object Hold Hard
Quadruped Run	Stick Push	**Key Turn Easy**
Walker Run	**Sweep**	Key Turn Hard

E.2 Baseline Algorithms

We standardize the common hyperparameters between all algorithms to minimize the differences between all implementations. We align the basic values with the state-of-the-art SAC implementation (D'Oro et al., 2022), thus limiting the requirement for extensive hyperparameter search across algorithms. We employ uniform network architectures and a standard ensemble of two critics (Fujimoto et al., 2018; Haarnoja et al., 2018a; Ciosek et al., 2019; Moskovitz et al., 2021; Cetin & Celiktutan, 2023). The set of tested algorithms includes:

- DAC - Dual Actor-Critic. The approach proposed in this paper, described in detail in Section 4 and Appendix C.
- SAC - Soft Actor-Critic (Haarnoja et al., 2018a) builds on DDPG (Silver et al., 2014). SAC extends the DDPG algorithm with features such as a stochastic policy using the reparametrization trick, automatic entropy temperature adjustment, Clipped Double Q-learning, and maximum entropy updates for actor and critic networks.

- TD3 - Twin Deep Deterministic Policy Gradient (Fujimoto et al., 2018) is an extension of the DDPG algorithm, including a target actor network and Clipped Double Q-learning.
- OAC - Optimistic Actor-Critic (Ciosek et al., 2019) extends SAC with an optimistic exploration policy approximating the Q-value upper bound.
- TOP - Tactical Optimism and Pessimism (Moskovitz et al., 2021) addresses the optimism-pessimism dilemma with an external discrete bandit learning β^p from predefined values. Based on TD3.
- GPL - Generalized Pessimism Learning (Cetin & Celiktutan, 2023) proposes updating pessimism in the pessimistic value approximation V^{π}_{β} through a dual optimization objective. Based on SAC.
- TD-MPC - Temporal Difference Learning for Model Predictive Control (Hansen et al., 2022) a state-of-the-art model-based algorithm leveraging model of the environment for critic-bootstrapped model-predictive control.

In high replay ratio, all algorithms undergo full-parameter resets according to the schedule proposed by D'Oro et al. (2022). As such, given replay ratio of 16 gradient steps per environment step, the agents are resetted in $(160k, 320k, 480k, 640k, 800k)$ steps. ma

Table 3: For non-shared hyperparameters we take values which were found to perform best by the authors of the respective method.

Method	Hyperparameter	Value
SAC / Shared	Network Size	$(256, 256)$
	Activations	ReLU
	Critic Ensemble Size	2
	Action Repeat	1
	Optimizer	Adam
	Learning Rate	$3e-4$
	Batch Size	256
	Discount	0.99
	Initial Temperature	1.0
	Initial Steps	10000
	Target Entropy	$\|\mathcal{A}\|/2$
	Polyak Weight	0.005
TD3	Policy Update Delay	2
	Exploration σ	0.5
	Target Policy σ	0.2
	Policy Delay	2
	Exploration Noise Clip	$\pm$ 0.5
OAC	Optimism	4.36
	Pessimism	-0.75
	KL Divergence Constraint	3.69
TOP	Optimistic Arm	0.0
	Pessimistic Arm	-1.0
	Bandit Learning Rate	0.1
	Bandit Decay	0.9
GPL	Initial Pessimism	-0.5
	Pessimistic Learning Rate	$1e-4$
DAC	Pessimism	-0.2
	Initial Optimism	1.0
	Initial KL Weight	0.25
	Target KL Divergence	0.25
	Standard Deviation Multiplier	1.25
	Adjustment Learning Rate	$3e-5$

E.3 Training Curves

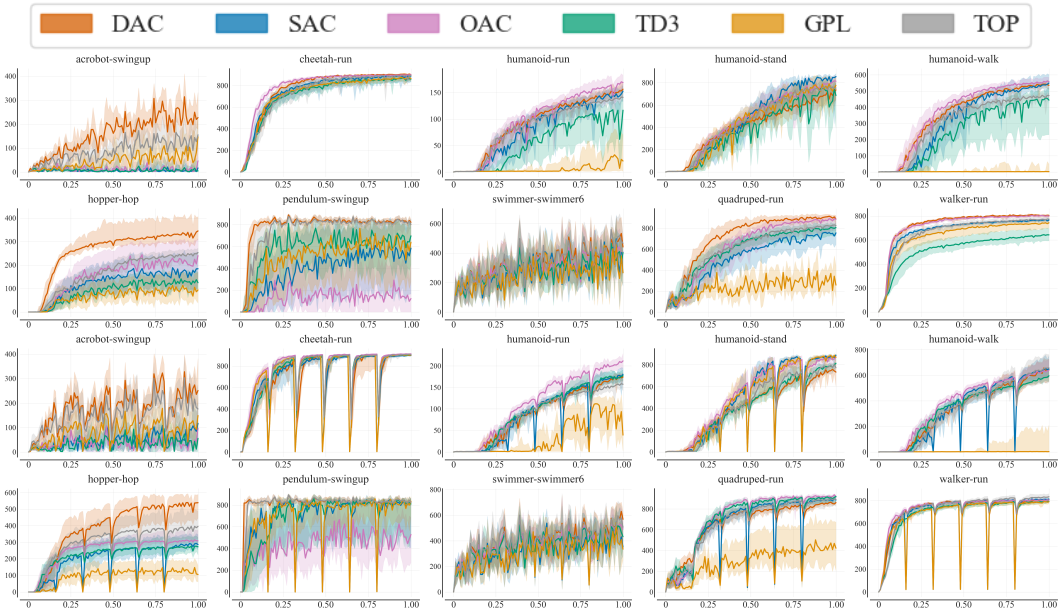


Figure 15: Training curves for DMC ($RR = 2$ rows 1 & 2, $RR = 16$ row 3 & 4). Y-axis denotes IQM and X-axis denotes environment steps. 10 seeds per task.

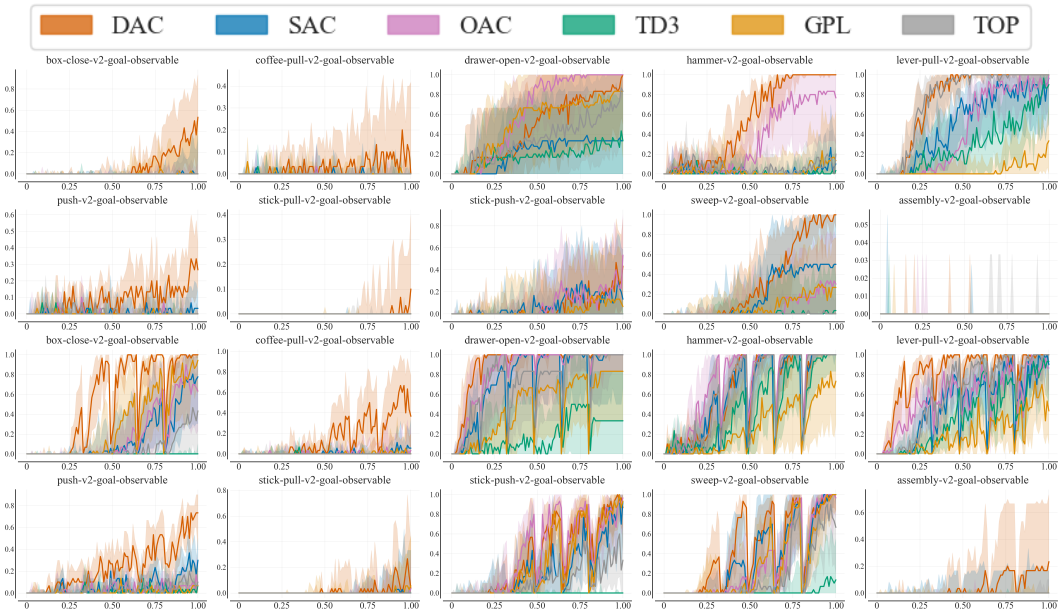


Figure 16: Training curves for MW ($RR = 2$ rows 1 & 2, $RR = 16$ rows 3 & 4). Y-axis denotes IQM success rates and X-axis denotes environment steps. 10 seeds per task.

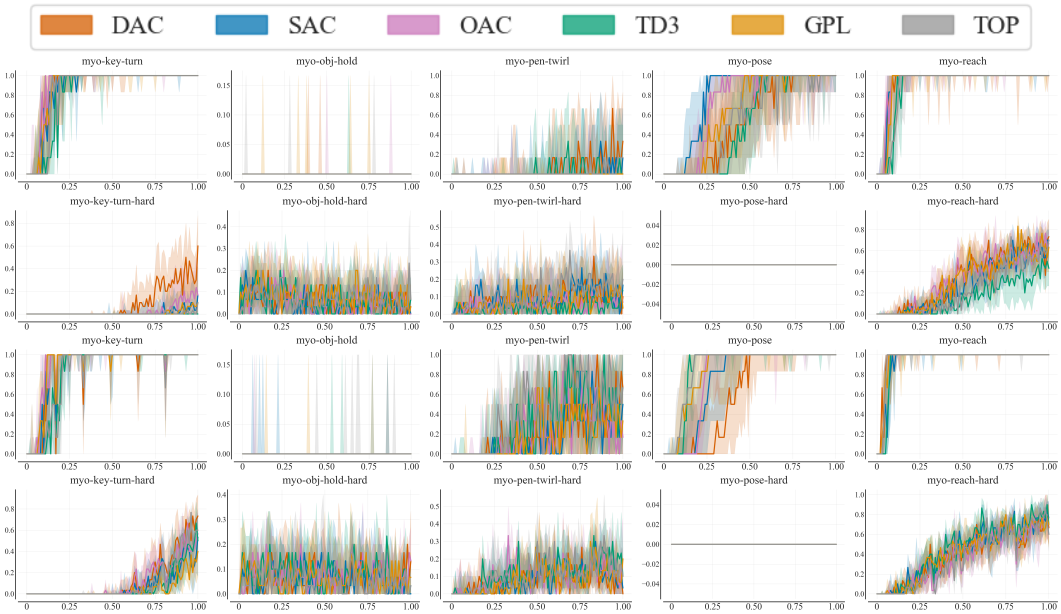


Figure 17: Training curves for MYO ($RR = 2$ rows 1 & 2, $RR = 16$ rows 3 & 4). Y-axis denotes IQM success rates and X-axis denotes environment steps. 10 seeds per task.