Decoupled Policy Actor-Critic: Bridging Pessimism and Risk Awareness in Reinforcement Learning

Michal Nauman[1], **Marek Cygan**[1,2]
[1]University of Warsaw
[2]Nomagic
nauman.mic@gmail.com, cygan@mimuw.edu.pl

Abstract

Actor-Critic (AC) algorithms like SAC and TD3 were shown to perform well 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 pessimistic and non-pessimistic RL objectives can be interpreted through expected utility maximization using an exponential utility function. This approach reveals that pessimistic policies effectively maximize value certainty equivalent, aligning them with the optimization of risk-aware objectives. Furthermore, we propose Decoupled Policy Actor-Critic (DAC). DAC is a 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 fewer computational resources, matches the performance of leading model-based methods in the complex dog and humanoid domains.

Introduction

Deep Reinforcement Learning (RL) is still in its infancy, with a variety of tasks unsolved (Sutton and 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, Hinton, and Williams 1986; Kingma and Ba 2014). Moreover, the limitations of robot hardware and data acquisition frequency constrain the maximum achievable replay ratio (Smith, Kostrikov, and Levine 2022). As such, it is worthwhile to pursue techniques that can be applied across a variety of RR configurations. One continuously researched theme is how a particular algorithm handles the *exploration-exploitation* dilemma (Ciosek et al. 2019; Moskovitz et al. 2021).

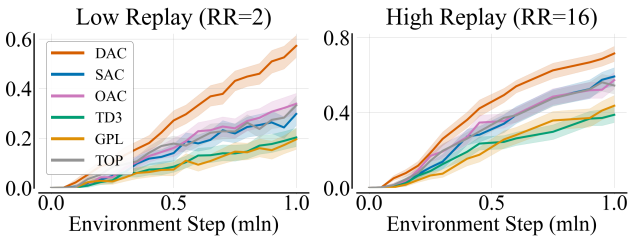

Figure 1: We test the proposed approach (DAC) against various pessimistic and optimistic actor-critic baselines in 30 tasks listed in Table 3. Y-axis reports IQM with 95% CI calculated using 10 seeds, with 1.0 being the maximal score.

Optimism and pessimism of algorithmic agents have been researched in multiple contexts. For instance, optimism in the face of uncertainty was shown to be an effective exploration strategy (Wang et al. 2020; Neu and Pike-Burke 2020). Conversely, pessimistic Q-learning strategies have proven beneficial in counteracting value overestimation caused by temporal difference errors (Hasselt 2010; Fujimoto, Hoof, and Meger 2018). However, there is a disconnect between these strategies and the foundational theories of RL, particularly in how they relate to the goal of value maximization. As a result, despite the empirical success of pessimistic agents like TD3 (Fujimoto, Hoof, and Meger 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 the derivation of both non-pessimistic (e.g., DDPG (Silver et al. 2014)), pessimistic (e.g., TD3 (Fujimoto, Hoof, and Meger 2018)), and optimistic (e.g., OAC (Ciosek et al. 2019)) objectives. 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 (Fei, Yang, and Wang 2021). Further-

more, we introduce Decoupled Policy Actor-Critic (DAC), an off-policy 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 Q-value bound for exploration, while the pessimistic actor optimizes a lower Q-value bound for temporal-difference (TD) learning (Fujimoto, Hoof, and Meger 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 pessimistic and optimistic 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 Decoupled Policy Actor-Critic (DAC), an off-policy actor-critic framework featuring a decoupled 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 of the optimism levels.

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.

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 policy 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, denoted as Q_θ) and the policy (ie. actor, denoted as π_ϕ). 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, Hoof, and Meger 2018; Haarnoja et al. 2018a) according to $\mathcal{L}_\theta = \mathrm{E}_{T\sim\mathcal{D}}\,(Q_\theta(s,a) - r_{s,a} - \gamma V_\theta(s'))^2$. Here, $Q_\theta(s,a)$ is the critic output, $V_\theta(s')$ 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 and Whiteson 2020): $\mathcal{L}_\phi = \mathrm{E}_{s\sim\mathcal{D}}(\alpha\log\pi_\phi(a|s) - Q_\theta(s,a))$ with $a \sim \pi_\phi(s)$.

Pessimism in Actor-Critic Algorithms like SAC or TD3 calculate actor-critic gradients with respect to the lower bound Q-values (Fujimoto, Hoof, and Meger 2018; Haarnoja et al. 2018b). Employing the lower-bound has proven effective in mitigating value overestimation in Temporal Difference (TD) learning (Van Hasselt, Guez, and Silver 2016; Fujimoto, Hoof, and Meger 2018). A popular approach is Clipped Double Q-Learning (CDQ), where the lower bound is calculated by taking the minimum value of an ensemble of critics, most often two (Fujimoto, Hoof, and Meger 2018; Haarnoja et al. 2018b; Ciosek et al. 2019; Hansen, Wang, and Su 2022): $V_\theta(s) \approx \min(Q^1_\theta(s,a), Q^2_\theta(s,a)) - \log\pi_\phi(a|s)$, 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. It was shown that using the minimum is equivalent to evaluating the ensemble statistics (Ciosek et al. 2019):

$$\min(Q^1_\theta(s,a), Q^2_\theta(s,a)) = Q^\mu_\theta(s,a) - Q^\sigma_\theta(s,a) \tag{1}$$

Where (μ, σ) denote the mean and standard deviation of the critic model ensemble and are given by:

$$\begin{aligned} Q^\mu_\theta(s,a) &= \frac{Q^1_\theta(s,a) + Q^2_\theta(s,a)}{2} \\ Q^\sigma_\theta(s,a) &= \frac{|Q^1_\theta(s,a) - Q^2_\theta(s,a)|}{2} \end{aligned} \tag{2}$$

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

$$\begin{aligned} Q^\beta_\theta(s,a) &= Q^\mu_\theta(s,a) + \beta Q^\sigma_\theta(s,a) \\ V^\beta_\theta(s) &= V^\mu_\theta(s) + \beta V^\sigma_\theta(s) \end{aligned} \tag{3}$$

Above, $Q^\beta_\theta(s,a)$ and $V^\beta_\theta(s)$ are the pessimistic Q-values and values respectively, β represents the degree of optimism. 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 level of pessimism is modulated by β. A non-zero β indicates a departure from neutrality: positive β yields an optimistic upper bound, while negative β results in a pessimistic

lower bound. Furthermore, as shown in Equation 1, in case of $\beta = -1$ the objective collapses to CDQ, and it is true that $Q^{\beta}_{\theta} = \min(Q^1_\theta(s,a), Q^2_\theta(s,a))$. In the setup of pessimistic learning the value targets are affected by the uncertainty measured via the value model ensemble disagreement. Thus, given some policy π_ϕ the difference between the pessimistic value (resulting from bootstrapping with Q^β), and neutral values (resulting from bootstrapping with Q^μ) is proportional to the expected values of the discounted sum of critic disagreements (Fujimoto, Hoof, and Meger 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, Gupta, and Levine 2020). Despite this, numerous effective off-policy algorithms adopt a non-neutral objective, leaning towards either pessimism or optimism (Fujimoto, Hoof, and Meger 2018; Ciosek et al. 2019; Cetin and Celiktutan 2023).

The policy derived from Equation 3 deviates from the optimal policy stemming from pure value maximization. This can be observed by unrolling the Bellman Equation used to train the critic. Whereas a non-pessimistic critic learns the sum of discounted returns and entropies, the pessimistic critic learns the critic disagreements as well:

$$\begin{aligned} &Q_\theta(s,a) \approx r_{s,a} + \gamma V^{\beta}_{\theta}(s') \\ &\approx r_{s,a} + \gamma(r_{s',a'} + \alpha\mathcal{H}^{\pi_\phi}(s) + V^{\sigma}_{\theta}(s') + \gamma V_\theta(s'')) \end{aligned} \tag{4}$$

Expected Utility Theorem The Von Neumann Morgenstern Theorem (Von Neumann and 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. Expected Utility Hypothesis (EUH) (Von Neumann and Morgenstern 1947; Kahneman and Tversky 1979) provides a framework for modeling decisions where outcomes are uncertain and as such is often the framework of choice to model risk-aware behavior. EUH 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 a setting, the goal of the agent is to optimize the utility rather than its input (Stiglitz 1997): $x^* = \arg\max_x \mathrm{E}_{x_i \sim X}\, \mathcal{U}(x_i)$. Here, the 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 $\arg\max \mathrm{E}\, x_i$. This discrepancy is reffered to as risk premium and we denote it by Υ. 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, along with their implied risk premium in Figure 5.

Theory of Pessimism in Actor-Critic

Notably, many state-of-the-art algorithms adopt pessimistic update strategies (Moskovitz et al. 2021; Hiraoka et al. 2021; D'Oro et al. 2022). However, the pessimistic correction $V^{\sigma}_{\theta}(s')$, as outlined in the previous Section, is not fully explicated by existing RL theory and does not emerge from pure value maximization problems. To this end, the efficacy of pessimistic or optimistic agents is often attributed to two factors: pessimism is justified by well-documented value overestimation in temporal learning (Hasselt 2010; Fujimoto, Hoof, and Meger 2018; Kumar, Gupta, and Levine 2020) while optimism is supported by lower regret guarantees for exploration (Chen et al. 2017; Ciosek et al. 2019). In this section, we show that off-policy agents that perform updates according to the pessimistic Q-values Q^β presented in Equation 3 (e.g., SAC (Haarnoja et al. 2018a), TD3 (Fujimoto, Hoof, and Meger 2018), OAC (Ciosek et al. 2019), TOP (Moskovitz et al. 2021) or GPL (Cetin and Celiktutan 2023)) can be interpreted as optimizing for the expected utility of values under an exponential utility function. As such, we align the pessimistic and optimistic objectives with risk-averse and risk-loving behaviour as stemming from an exponential utility. To show this, 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 $\mathcal{V}^{\pi}_{\theta}(s)$, which is assumed to have finite moments and expected value denoted by V^{μ}_{θ}, such that $V^{\mu}_{\theta}(s) = \mathrm{E}_{i\sim\mathcal{V}}\, V^{i}_{\theta}(s)$. Assuming that the expected value of $V^{i}_{\theta}(s)$ is an unbiased estimator of the on-policy value, the standard approach requires the policy to optimize for $V^{\mu}_{\theta}(s)$, such that $\pi^* = \arg\max_{\pi\in\Pi} \mathrm{E}_{s\sim\mathcal{D}} V^{\mu}_{\theta}(s)$. 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 $V^{c}_{\theta}(s)$:

$$V^{c}_{\theta}(s) = \mathcal{I} \mathop{\mathrm{E}}_{i\sim\mathcal{V}} \mathcal{U}\, V^{i}_{\theta}(s) = V^{\mu}_{\theta}(s) + \Upsilon(s). \tag{5}$$

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 behavior, then $\Upsilon(s) < 0$ resulting in a certainty equivalent value that is smaller than $V^{\mu}_{\theta}(s)$. Building on the expected utility objective, we define the *certainty equivalence policy* that seeks to amortize the uncertainty associated with an approximation of values:

$$\pi^{c}_{\phi} = \mathop{\arg\max}_{\pi\in\Pi} \mathop{\mathrm{E}}_{s\sim p} \mathop{\mathrm{E}}_{i\sim\mathcal{V}} \mathcal{U}\, V^{i}_{\theta}(s) = \mathop{\arg\max}_{\pi\in\Pi} \mathop{\mathrm{E}}_{s\sim p} V^{c}_{\theta}(s). \tag{6}$$

Above, π^{c}_{ϕ} 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_ϕ optimizes for values that are not generally equal to the risk-neutral values. Specifically, the π^c_ϕ and π^* are equivalent only iff $\Upsilon(s) = 0$, thus making $V^c_\theta(s) = V^\mu_\theta(s)$. This highlights the fact that the utility-based objective is a generalization of the traditional RL objective, with the 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.

To 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:

$$\Upsilon(s) \approx \mathcal{I} \mathop{\mathrm{E}}_{i\sim\mathcal{V}} \sum_{n=1}^{\infty} \frac{\mathcal{U}^n\big(V^\mu_\theta(s)\big)}{n!} \big(V^i_\theta(s) - V^\mu_\theta(s)\big)^n. \quad (7)$$

Above, we denote $\mathcal{U}^n(V^\mu_\theta(s))$ as the nth derivative of $\mathcal{U}$ calculated at $V^\mu_\theta(s)$. As such, the discrepancy between the certainty equivalent value and the risk-neutral value depends on moments of $\mathcal{V}^\pi_\theta(s)$. Using Equation 7 and assuming that the utility is an exponential function $\mathcal{U}(V^i_\theta, \beta) = -e^{2\beta V_i(s)}$ yields a risk premium is equal to:

$$\Upsilon(s) \approx \beta \mathop{\mathrm{E}}_{i\sim\mathcal{V}} \big(V^i_\theta(s) - V^\mu_\theta(s)\big)^2 + \zeta(\beta). \quad (8)$$

Where $\zeta(\beta)$ denotes the sum of Taylor series of higher order than 2. Equation 8 shows that the risk premium is dependent on variance for arbitrary distribution $\mathcal{V}^\pi_\theta(s)$. By assuming the distribution of $\mathcal{V}^\pi_\theta(s)$ to be Gaussian with mean $V^\mu_\theta(s)$ and standard deviation $V^\sigma_\theta(s)$, then:

$$\Upsilon(s) = \beta \mathop{\mathrm{E}}_{i\sim\mathcal{V}} \big(V^i_\theta(s) - V^\mu_\theta(s)\big)^2. \quad (9)$$

Which implies that $\sqrt{\Upsilon(s)} = V^\sigma_\theta(s)$. Then the certainty equivalent value is equal to:

$$V^c(s) \approx \underbrace{\mathop{\mathrm{E}}_{i\sim\mathcal{V}} V_i(s)}_{\text{Ensemble Mean}} + \beta \underbrace{\mathop{\mathrm{E}}_{i\sim\mathcal{V}} \big(V_i(s) - V^\mu(s)\big)^2}_{\text{Ensemble Variance}}. \quad (10)$$

We further detail these results in the Appendix. As a result of these derivations, pessimistic agents like TD3 (Fujimoto, Hoof, and Meger 2018) and SAC (Haarnoja et al. 2018b), as well as optimistic agents like OAC (Ciosek et al. 2019), can be seen as optimizing for the expected utility objective. As such, rather than aiming for expected values, they pursue certainty equivalent values. This insight effectively connects risk-awareness with pessimism and optimism in Actor-Critic methods, as implemented through the objectives in Equation 3. In this context, the pessimistic correction $V^\sigma_\theta(s)$ acts as a risk premium, adjusting the learning to account for errors in the critic network. If $\beta < 0$, then the utility is risk-averse and the agent is pessimistic towards the errors. Conversely, if $\beta > 0$ then the utility is risk-loving, and the agent is optimistic.

Decoupled Policy Actor-Critic

In this section, we propose the Decoupled Policy 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. 2020) 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 a dataset of previous experiences (Haarnoja et al. 2018a).

Algorithm 1: DAC training loop

Inputs: π^p_ϕ - pessimistic actor; π^o_η - optimistic actor; Q_θ - critic; $Q_{\bar\theta}$ - target critic; α - temperature; β^o - optimism; τ - KL weight; β^p - pessimism; $\mathcal{KL}^*$ - target KL; m - exploration multiplier

Sample action from the optimistic actor
$s', r = \text{ENV.STEP}(a) \quad a \sim \pi^o_\eta$
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. 13)
  $a' \sim \pi^p_\phi(s')$
  $\theta \leftarrow \nabla_\theta \big(Q_\theta(s,a) - (r + \gamma(Q^{\beta^p}_{\bar\theta}(s',a') - \alpha \log \pi^p_\phi(a'|s'))\big)^2$
  Update pessimistic actor (Eq. 12)
  $\phi \leftarrow \nabla_\phi \big(Q^{\beta^p}_\theta(s,a) - \alpha \log \pi^p_\phi(a|s)\big), \quad a \sim \pi^p_\phi(s)$
  Update optimistic actor (Eq. 11)
  $\mu(\bar\pi^o_\eta) = \mu(\pi^o_\eta), \quad \sigma(\bar\pi^o_\eta) = m\sigma(\pi^o_\eta)$
  $\eta \leftarrow \nabla_\eta \big(Q^{\beta^p}_\theta(s,a) - \tau KL(\pi^p_\phi|\bar\pi^o_\eta)\big), \quad a \sim \pi^o_\eta(s)$
  Update entropy temperature
  $\alpha \leftarrow \alpha - \nabla_\alpha \alpha \big(\mathcal{H}^* - \mathcal{H}(s)\big)$
  Update optimism (Eq. 14)
  $\beta^o \leftarrow \beta^o - \nabla_{\beta^o}(\beta^o - \beta^p)(\frac{1}{|A|} KL(\pi^p_\phi|\pi^o_\eta) - \mathcal{KL}^*)$
  Update KL weight (Eq. 14)
  $\tau \leftarrow \tau + \nabla_\tau \tau(\frac{1}{|A|} KL(\pi^p_\phi|\pi^o_\eta) - \mathcal{KL}^*)$
  Update target network
  $\bar\theta \leftarrow \text{POLYAK}(\theta, \bar\theta)$
end for

Whereas the results presented in the previous Section validate the use of optimistic and pessimistic policies in terms of the optimality of the pursued solutions, we further develop DAC based 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, Hoof, and Meger 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, Stinchcombe, and White 1989). DAC adjusts the optimism levels associated with the upper bound such that the two policies reach a predefined divergence target, thus alleviating the tandem problem (Ostrovski, Castro, and Dabney 2021) 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 decoupled actor comprising pessimistic and optimistic actors denoted π^p_ϕ and π^o_η respectively. As shown in the previous Section, 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:

$$\mathcal{L}_\eta = \mathop{\mathrm{E}}_{s\sim\mathcal{D}} \tau KL\big(\pi^p_\phi | \bar{\pi}^o_\eta\big) - Q^{\beta^o}_\theta(s,a), \quad a \sim \pi^o_\eta(s). \tag{11}$$

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, $Q^{\beta^o}_\theta(s,a)$ is the risk-aware Q-value defined in Equation 3 for β^o, 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 3, 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 the Appendix.

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

$$\mathcal{L}_\phi = \mathop{\mathrm{E}}_{s\sim\mathcal{D}} \alpha \log \pi^p_\phi(a|s) - Q^{\beta^o}_\theta(s,a), \quad a \sim \pi^p_\phi(s). \tag{12}$$

Here, $\mathcal{L}_\phi$ is the pessimistic actor loss, $\mathcal{D}$ is the experience buffer, α is the entropy temperature, β^p is the pessimism parameter, and $Q^{\beta^o}_\theta(s,a)$ denotes the pessimistic Q-value defined in Equation 3 for β^p. As discussed in the previous Section, optimizing for the expected utility objective under an exponential utility function leads to a risk-aware value. As such, the pessimistic actor convergences to the optimal policy under CDQ assumptions (Fujimoto, Hoof, and Meger 2018). The critic parameters are updated via SARSA under the pessimistic policy:

$$\begin{aligned} \mathcal{L}_\theta &= \mathop{\mathrm{E}}_{T\sim\mathcal{D}} \big(Q_\theta(s,a) - r - \gamma V^{\beta^p}_{\bar{\theta}}(s')\big)^2 \\ V^{\beta^p}_{\bar{\theta}}(s) &\approx Q^{\beta^p}_{\bar{\theta}}(s,a) - \alpha \log \pi^p_\phi(a|s), \quad a \sim \pi^p_\phi(s). \end{aligned} \tag{13}$$

Above, $\mathcal{L}_\theta$ 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, Hoof, and Meger 2018; Haarnoja et al. 2018a). This setup aligns the objectives of the critic and the pessimistic actor in DAC with the traditional 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:

$$\begin{aligned} \mathcal{L}_{\beta^o} &= \mathop{\mathrm{E}}_{s\sim\mathcal{D}} (\beta^o - \beta^p) D(s) \;\text{ for }\; \beta^o \in (\beta^p, \infty), \\ \mathcal{L}_\tau &= \mathop{\mathrm{E}}_{s\sim\mathcal{D}} -\tau D(s) \;\text{ for }\; \tau \in (0, \infty). \end{aligned} \tag{14}$$

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:

$$D(s) = \frac{1}{|A|} KL\big(\pi^p_\phi(s) | \pi^o_\eta(s)\big) - \mathcal{KL}^*. \tag{15}$$

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

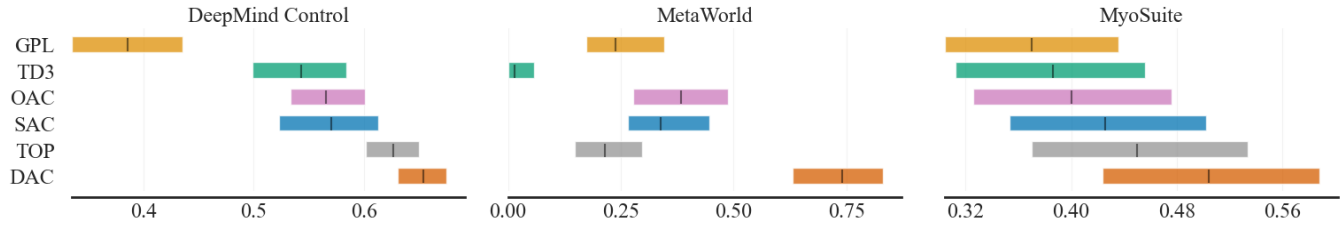

Figure 2: We report final IQM in 30 tasks (Table 3) averaged between low ($RR = 2$) and high replay settings ($RR = 16$). In high replay, all algorithms use resets as proposed by D'Oro et al. (2022). 1.0 denotes the maximal score, 95% CI, and 10 seeds.

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 8. We expand our discussion of DAC in the Appendix.

Experiments

Our 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 3, sourced from the DeepMind Control (DMC) (Tassa et al. 2018), MetaWorld (MW) (Yu et al. 2020) and MyoSuite (MYO) (Caggiano et al. 2022) benchmarks. In DMC we report the returns, whereas in MW and MYO we report the success rates. We calculate evaluation metrics using the RLiable package (Agarwal et al. 2021).

Model-free benchmark We test DAC against an array of both risk-neutral and risk-aware baselines, including TD3 (Fujimoto, Hoof, and Meger 2018), SAC (Haarnoja et al. 2018a), OAC (Ciosek et al. 2019), TOP (Moskovitz et al. 2021), and GPL (Cetin and 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, Hoof, and Meger 2018; Haarnoja et al. 2018a; Ciosek et al. 2019; Moskovitz et al. 2021; Cetin and 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 final performance of the best baseline in only 60% of environment steps. Furthermore, as shown in Figure 2, DAC approach to balancing risk-averse exploitation with risk-loving exploration yields significant performance benefits, particularly visible in the MetaWorld tasks. We present task-dependent training curves in the Appendix.

Design decisions We explore how DAC performance is influenced by variations in its design. We perform evaluations of various DAC modifications in 15 tasks listed in Table 3, 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 effective use of optimism, as evidenced by the 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 (Ostrovski, Castro, and Dabney 2021).

	$RR = 2$	$RR = 16$
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

Table 1: We evaluate design choices associated with DAC.

Hyperparameter Sensitivity As discussed previously, DAC introduces three tunable hyperparameters: KL target (the target divergence between optimistic and pessimistic policies as introduced in Equation 15), exploration multiplier (parameter that determines how much larger the variance of an optimistic policy is compared to a pessimistic policy as discussed under Equation 11), as well as the learning rates for the adjustment mechanism for β^o and τ (for opti-

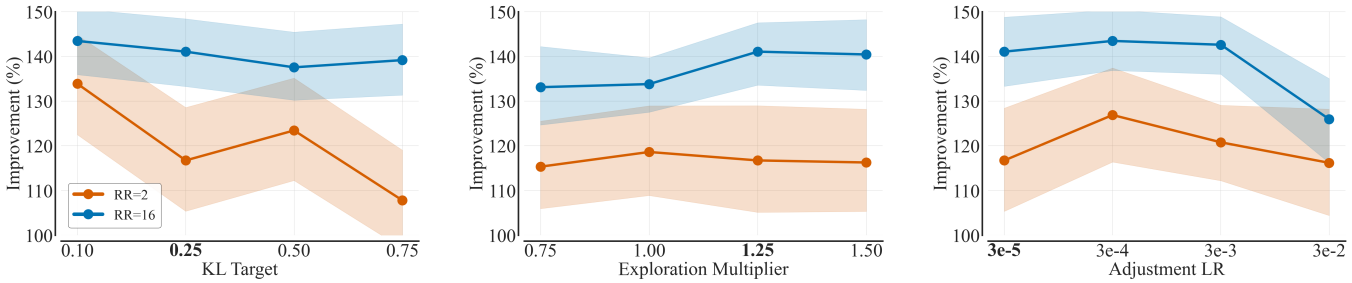

Figure 3: We evaluate DAC when changing the values of its hyperparameters (X-axis) for two replay regimes. The bold value denotes the value used in the main experiment. Y-axis reports the percentage improvement over tuned SAC.

mizing Equation 14). 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 $500k$ steps, in both replay regimes, and on 15 tasks listed in Table 3. We report these results In Figure 3, where we compare the final performance of these agents to SAC with analogical replay ratio. As follows, we find that DAC is stable when changing the values of its hyperparameters, with all variations outperforming the SAC baseline.

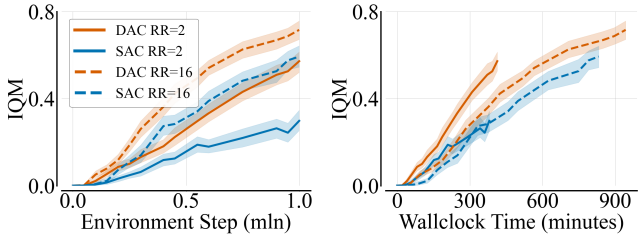

Figure 4: Despite additional compute associated with the decoupled actor, the improvements stemming from DAC lead to a beneficial compute/performance tradeoff. Experiments were conducted on an NVIDIA A100 40GB RAM.

Additional Experiments Firstly, we evaluate the compute costs associated with running a decoupled actor setup proposed by DAC. We find that DAC requires around 15% additional wall clock time to complete 1 mln environment step training compared to SAC. However, as shown in Figure 4, improvements in DAC performance offset additional compute costs, leading to a beneficial trade-off between compute and performance. Furthermore, in Appendix, we detail supplementary experiments conducted to further evaluate DAC. In Figure 9, we compare DAC with model-based TD-MPC when training for 3 mln environment steps in the dog and humanoid domains of DMC. We find that DAC is able to achieve the performance of model-based TD-MPC on these tasks, despite using smaller parameter count. Figure 10 studies 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). We find that using layer normalization further improves DAC performance on locomotion tasks. In Figure 11, we find that the decoupled policy architecture indeed does not increase the value overestimation over the pessimistic baselines. Finally, in Figure 12, we investigate the performance of distributional DAC that explicitly models aleatoric and epistemic uncertainties and is optimistic only with respect to specific types.

Limitations

The main limitation of DAC is its dual-actor framework, which incurs slightly higher memory and computation costs relative to the standard SAC. Furthermore, DAC introduces three additional tunable hyperparameters. Although tests showed DAC robust performance across a variety of hyperparameter values, it is uncertain whether this robustness translates to more complex environments.

Conclusions

In this work, we apply the expected utility theorem to reason about the empirical effectiveness of pessimistic and optimistic 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 risk-aware utility function. This analysis shows that the commonly used pessimistic update rules are, in fact, risk-averse with respect to critic errors. 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, an off-policy actor-critic framework. DAC employs two actors with distinct levels of pessimism to optimize their respective expected utility functions. The pessimistic actor focuses on TD learning and evaluation, while the optimistic actor facilitates exploration. We evaluated DAC in various locomotion and manipulation tasks and compared it with more than 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 the robustness of DAC performance across various 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 also gratefully acknowledge the Polish high-performance computing infrastructure, PLGrid (HPC Center: ACK Cyfronet AGH), for providing computational resources and support under grant no. PLG/2024/017817. 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. 2021. Deep reinforcement learning at the edge of the statistical precipice. *Advances in neural information processing systems*, 34: 29304–29320.

Caggiano, V.; Wang, H.; Durandau, G.; Sartori, M.; and Kumar, V. 2022. MyoSuite–A contact-rich simulation suite for musculoskeletal motor control. *arXiv preprint arXiv:2205.13600*.

Cetin, E.; and Celiktutan, O. 2023. Learning Pessimism for Reinforcement Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 6971–6979.

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

Chen, X.; Wang, C.; Zhou, Z.; and Ross, K. W. 2020. Randomized Ensembled Double Q-Learning: Learning Fast Without a Model. In *International Conference on Learning Representations*.

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

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

D'Oro, P.; Schwarzer, M.; Nikishin, E.; Bacon, P.-L.; Bellemare, M. G.; and Courville, A. 2022. Sample-Efficient Reinforcement Learning by Breaking the Replay Ratio Barrier. In *The Eleventh International Conference on Learning Representations*.

Fei, Y.; Yang, Z.; and Wang, Z. 2021. Risk-sensitive reinforcement learning with function approximation: A debiasing approach. In *International Conference on Machine Learning*, 3198–3207. PMLR.

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

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

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

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

Hansen, N.; Wang, X.; and Su, H. 2022. Temporal Difference Learning for Model Predictive Control. In *International Conference on Machine Learning, PMLR*.

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

Hiraoka, T.; Imagawa, T.; Hashimoto, T.; Onishi, T.; and Tsuruoka, Y. 2021. Dropout Q-Functions for Doubly Efficient Reinforcement Learning. In *International Conference on Learning Representations*.

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

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

Kahneman, D.; and Tversky, A. 1979. Prospect Theory: An Analysis of Decision under Risk. *Econometrica*, 47(2): 263–292.

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

Kostrikov, I. 2021. JAXRL: Implementations of Reinforcement Learning algorithms in JAX.

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

Li, Q.; Kumar, A.; Kostrikov, I.; and Levine, S. 2022. Efficient Deep Reinforcement Learning Requires Regulating Overfitting. In *The Eleventh International Conference on Learning Representations*.

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. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533.

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

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

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

Ostrovski, G.; Castro, P. S.; and Dabney, W. 2021. The difficulty of passive learning in deep reinforcement learning. *Advances in Neural Information Processing Systems*, 34: 23283–23295.

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

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

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

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

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

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

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

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

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

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

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

Wang, Y.; Wang, R.; Du, S. S.; and Krishnamurthy, A. 2020. Optimism in Reinforcement Learning with Generalized Linear Function Approximation. In *International Conference on Learning Representations*.

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

Zawalski, M.; Tyrolski, M.; Czechowski, K.; Odrzygóźdź, T.; Stachura, D.; Piekos, P.; Wu, Y.; Kuciński, Ł.; and Miłoś, P. 2022. Fast and Precise: Adjusting Planning Horizon with Adaptive Subgoal Search. In *The Eleventh International Conference on Learning Representations*.