Tactical Optimism and Pessimism for Deep Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

In recent years, deep off-policy actor-critic algorithms have become a dominant approach to reinforcement learning for continuous control. One of the primary drivers of this improved performance is the use of pessimistic value updates to address function approximation errors, which previously led to disappointing performance. However, a direct consequence of pessimism is reduced exploration and runs counter to theoretical support for the efficacy of optimism in the face of uncertainty. So which approach is best? In this work, we show that the most effective degree of optimism can vary both across tasks and over the course of learning. Inspired by this insight, we introduce a novel deep actor-critic framework, Tactical Optimistic and Pessimistic estimation (TOP) to switch between optimistic and pessimistic value learning *online* by formulating the selection as a multi-arm bandit problem. We show in a series of continuous control tasks that TOP outperforms existing methods which rely on a fixed degree of optimism, setting a new state of the art in challenging pixel-based environments. Since our changes are simple to implement, we believe these insights can easily be incorporated into a multitude off-policy algorithms.

1 Introduction

Reinforcement learning (RL) has achieved significant empirical successes in recent years [35, 47, 4]. Value function approximation with deep neural networks has been a fundamental part of this success, allowing RL to scale to high-dimensional domains and complex tasks. However, it has long been known that the inherent error that comes with any function approximator can lead to a positive bias in value computation [51]. This overestimation of the expected reward that the current policy will achieve can be seen as a form of *optimism*—the agent believes it will achieve greater reward than it is actually expected to reach under the current policy.

Recent state-of-the-art approaches to continuous control problems in reinforcement learning have divergent views on this bias towards optimism. One side seeks to correct the overestimation, for example, by using the minimum of two value estimates as a form of approximate lower bound [20]. This approach can be seen as *pessimistic* with respect to the current value function. On the other hand, [14] have recently argued that the inherent optimism of approximate value estimates is actually useful for encouraging exploration of the environment and/or action space. Interestingly, both sides have used their respective positions to derive state-of-the-art algorithms. How can this be, if their views are seemingly opposed? Our key hypothesis is the following:

The degree of estimation bias, and subsequent efficacy of an optimistic strategy, is dependent on both the environment and the stage of optimization.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

This hypothesis is the key motivation for our work. We demonstrate that, conditioned on a given function approximator, different environments require different overall levels of optimism or pessimism, and furthermore, that it may be important to adapt the level of optimism or pessimism over the course of training. This observation is linked to both varying stochasticity across environments, measured by *aleatoric uncertainty*, and to the non-stationarity of bias during training, which is based in *epistemic uncertainty*. Aleatoric and epistemic uncertainty are addressed in more detail in Section 5.1. With this in mind, we also posit the following:

When the level of bias is unknown, an adaptive strategy can be highly effective.

In this work, we explore both these hypotheses and introduce a new framework for value estimation in deep RL called *Tactical Optimism and Pessimism* (TOP). This approach acknowledges the inherent uncertainty in the level of estimation bias present, and rather than adopt a blanket optimistic or pessimistic strategy, it estimates the optimal approach *on the fly*, by formulating the optimism/pessimism dilemma as a multi-armed bandit problem. Furthermore, TOP explicitly isolates both aleatoric and epistemic uncertainty by representing the environmental return using a distributional critic and the model uncertainty with an ensemble. This process is summarized in Figure 1.

We show in a series of experiments that not only does the efficacy of optimism indeed vary as we suggest, but TOP is able to capture the best of both worlds, achieving a new state of the art for challenging continuous control problems.

Our main contributions are as follows:

- This work is the first, to our knowledge, to show that the efficacy of optimism for a fixed function approximator varies across environments and during training for reinforcement learning with function approximation.
- We propose a novel framework for value estimation, *Tactical Optimism and Pessimism* (TOP), which learns to balance optimistic and pessimistic value estimation online. TOP frames the decision to be optimistic or pessimistic as a multi-armed bandit problem, which we show recovers optimal performance with respect to variants with fixed optimism.
- Our experiments demonstrate that these insights, which require only simple changes to popular algorithms, lead to state of the art results on both state- and pixel-based control.

2 Related Work

Much of the recent success of off-policy actor critic algorithms build on DDPG [33], which extended the deterministic policy gradient [47] approach to off-policy learning with deep networks, using insights from DQN [35]. Like D4PG [7], we combine DPG with distributional value estimation. However, unlike D4PG, we use two critics, a quantile representation rather than a categorical distribution [9], and, critically, adapt value estimation to be optimistic or pessimistic. Since then, several improvements have been made, such as TD3, SAC, DrQ, and PI-SAC [20, 21, 56, 32], which have become the state-of-the-art for continuous control.

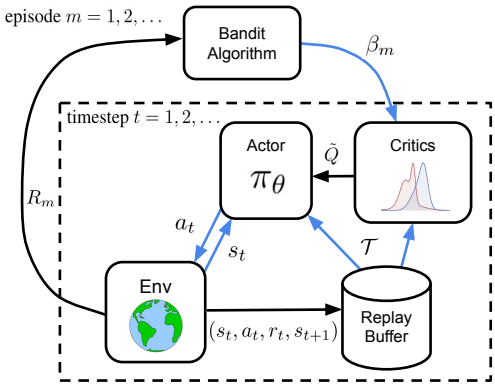

Figure 1: Visualization of the TOP framework. Blue arrows denote stochastic variables.

The principle of optimism in the face of uncertainty [2, 29] can be used to design algorithms that trade off the agent's desire to maximize its reward against its need to explore state-action pairs with high epistemic uncertainty. In a sense, this framework encourages agents to explore regions of the state-action space where the agent does not have enough information to rule out the possibility of collecting a large reward. Optimism has been a principle used to design RL algorithms for many years [11], and produces state-of-the-art regret bounds in the tabular setting [27, 18, 19, 3, 8, 53], and even in some simple theoretical models for function approximation [28, 55].

However, optimistic approaches have had limited success when combined with deep neural networks [14]. To be successful, these approaches need to be optimistic enough to upper bound the true value

function while maintaining low estimation error [39]. This becomes challenging when using function approximation, which often results in overestimation bias.

Recently, there has been increasing evidence in support of the efficacy of adaptive algorithms [5, 46, 44, 43]. A prominent use of bandits for varying components of RL algorithms is Agent57 [4], the first agent to outperform the human baseline for all 57 games in the Arcade Learning Environment [10]. Agent57 adaptively switches among different exploration strategies, whereas our approach adopts differing levels of optimism in its value estimates.

Finally, our work is also related to automated RL (AutoRL), as we can consider TOP to be learning a learning rule on-the-fly [15, 37]. An exciting area of future work will be to consider the interplay between the degree of optimism and model hyperparameters such as architecture and learning rate, and whether they can be adapted simultaneously for example using PBT [26].

3 Preliminaries

Reinforcement learning considers the problem of training an agent to interact with its environment so as to maximize its cumulative reward. Typically, a task and environment are cast as a Markov decision process (MDP), formally defined as a tuple $(\mathcal{S}, \mathcal{A}, p, r, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the space of possible actions, $p : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ is a transition kernel, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\gamma \in [0, 1)$ is a discounting factor. For a given policy π, the *return* $Z^\pi = \sum_t \gamma^t r_t$, is a random variable representing the sum of discounted rewards observed along one trajectory of states while following π and up to some, potentially infinite, time horizon T. The goal is to update the agent's policy $\pi_\theta : \mathcal{S} \to \mathcal{A}$ with parameters θ so as to maximize the *expected return*, or discounted cumulative reward $J(\theta) = \mathbb{E}_\pi \left[\sum_t \gamma^t r_t\right] = \mathbb{E}[Z^\pi]$.

Actor-critic algorithms are a family of approaches to this problem in which the policy π, or *actor* is trained to maximize expected return, while a *critic* evaluates the actions of the policy. Typically, the critic takes the form of a value function which predicts the expected return under the current policy $Q^\pi(s, a) := \mathbb{E}_\pi[Z_t | s_t = s, a_t = a]$. When the state space is large, Q^π may be parameterized by a model with parameters ϕ. The deterministic policy gradient (DPG) theorem [48] shows that gradient ascent on J can be performed via

$$\nabla_\theta J(\theta) = \mathbb{E}_\pi[\nabla_a Q^\pi(s, a)|_{a=\pi(s)} \nabla_\theta \pi_\theta(s)] \tag{1}$$

The critic is updated separately, usually via SARSA [49], which, given a transition $s_t, a_t \to r_{t+1}, s_{t+1}$, constructs a learning signal via semi-gradient descent on the squared TD error δ_t^2, where

$$\delta_t := y_t - Q^\pi(s_t, a_t) = r_{t+1} + \gamma Q^\pi(s_{t+1}, \pi(s_{t+1})) - Q^\pi(s_t, a_t), \tag{2}$$

where y_t is the *Bellman target*. Rather than simply predict the mean of the return Z^π under the current policy, it's recently been shown to be advantageous to learn the full *distribution* of Z^π given the current state and action, $Z^\pi(s_t, a_t)$ [9, 17, 16, 45]. In this framework, the return distribution is typically parameterized via a set of K chosen statistics (e.g., quantiles or expectiles) which are learned via minimization of an appropriate loss function. For example, the kth quantile of the distribution at state s and associated with action a, $q_k(s, a)$, can be learned via the gradient descent on the *Huber loss* [25] of the *distributional Bellman error* $\delta_k = \hat{Z} - q_k(s, a)$, $\hat{Z} \sim Z^\pi(\cdot|s, a)$. While $\hat{Z}$ is formally defined as a sample from the return distribution, δ_k is typically computed in practice as $K^{-1} \sum_{j=1}^{K} r + \gamma q_j(s, a) - q_k(s, a)$ [17].

4 Optimism versus Pessimism

Reducing overestimation bias with pessimism [51] originally observed that Q-learning [54] with function approximation was biased towards overestimation. This overestimation bias can be seen as an undesired over-optimism about the value function of the current policy and can introduce instability in training [20]. To reduce the overestimation bias in the critic, [20] introduced the Twin Delayed Deep Deterministic (TD3) policy gradient algorithm, a pessimistic approach. In TD3, values are estimated via a SARSA-like variant of double Q-learning [24] in which the Bellman target is constructed by taking the minimum of *two* critics:

$$y_t = r_{t+1} + \gamma \min_{i \in \{1,2\}} Q^\pi_{\phi_i}(s, \pi_\theta(s) + \epsilon), \tag{3}$$

where $\epsilon \sim \text{clip}(\mathcal{N}(0, s^2), -c, c)$ is noise drawn from a clipped Gaussian distribution (c is a constant). This added noise is used for smoothing in order to prevent the actor from overfitting to narrow peaks in the value function. Secondly, TD3 *delays* policy updates, updating value estimates several times between each policy gradient step. By taking the minimum of two separate critics and increasing the number of critic updates for each policy update, this approach takes a pessimistic view on the policy's value in order to reduce overestimation bias. These ideas have become ubiquitous in state-of-the-art continuous control algorithms [6], such as SAC, RAD, (PI)-SAC [21, 30, 32].

Optimism in the face of uncertainty While the pessimistic approaches discussed above produce strong results, there are also potential benefits in optimistic exploration, a cornerstone of many theoretically grounded RL algorithms [27, 3]. In essence, the goal of optimistic value estimation is to explore regions of the state space with high epistemic uncertainty, encouraging the agent to collect more data in unexplored regions. Moreover, [14] found that reducing value estimates, as done in pessimistic algorithms, can lead to *pessimistic underexploration*, in which actions that could lead to experience that gives the agent a better understanding of its environment are never selected. To address this, they introduced the Optimistic Actor-Critic (OAC) algorithm, which trains an *exploration policy* using an optimistic upper bound on the value function while constructing targets for learning using the lower bound of [20]. OAC demonstrated improved performance compared to SAC, hinting at the complex interplay between optimism and pessimism in deep RL algorithms.

Trading off optimism and pessimism As we have discussed, there are arguments for both optimistic and pessimistic approaches. Optimism can aid exploration, but if there is significant estimation error, then a more pessimistic approach may be needed to stabilize learning. Moreover, both approaches are supported by strong empirical evidence. While this might seem contradictory at first, we hypothesize that the performance of either approach is highly dependent on the nature of the task. To test this hypothesis, we trained "Optimistic" and "Pessimistic" versions of the same deep actor-critic algorithm (details in Section 6) for two different tasks and compared their performance in Figure 2. As we can see, in the HalfCheetah task, the Optimistic agent outperforms the Pessimistic agent, while in the Hopper task, the opposite is true. This result suggests that a one-size-fits-all approach is not optimal for the optimism-pessimism trade-off. Indeed, we believe this motivates the use of an adaptive approach, in which the degree of optimism or pessimism is adjusted dynamically during training. As a consequence, the optimal degree of optimism may vary across tasks and over the course of a single training run as the model improves.

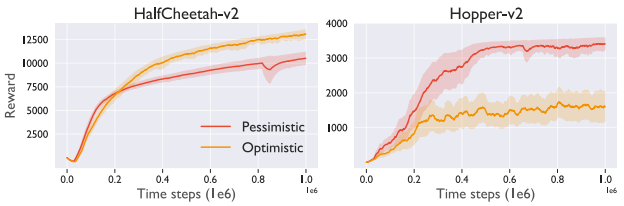

Figure 2: Optimistic and Pessimistic algorithms averaged over 10 seeds. Shading is one half std.

5 Tactical Optimistic and Pessimistic Value Estimation

TOP is based on the idea of adaptive optimism in the face of uncertainty. We begin this section by discussing how TOP represents uncertainty, before introducing the mechanism by which it dynamically adapts during learning.

5.1 Representing uncertainty in TOP estimation

TOP distinguishes between two types of uncertainty, *aleatoric* and *epistemic* uncertainty, and represents them using two separate mechanisms.

Aleatoric uncertainty reflects the noise that is inherent to the environment regardless of the agent's understanding of the task. Following [9, 17, 16, 45], TOP represents this uncertainty by learning the full return distribution $Z^\pi(s, a)$ for a given policy π and state-action pair (s, a) rather than only the expected return $Q^\pi(s, a) = \mathbb{E}[Z^\pi(s, a)]$. Depending on the stochasticity of the environment, the distribution $Z^\pi(s, a)$ is more or less spread out, thereby acting as a measure of aleatoric uncertainty.

Epistemic uncertainty reflects lack of knowledge about the environment and is expected to decrease as the agent gains experience. TOP uses this uncertainty to quantify how much an optimistic belief about the return differs from a pessimistic one. Following [14], we can model epistemic

uncertainty via a Gaussian distribution with mean $\bar{Z}(s,a)$ and standard deviation $\sigma(s,a)$. However, unlike in [14] where the parameters of the Gaussian are deterministic functions, we treat both $\bar{Z}(s,a)$ and $\sigma(s,a)$ as random variables in accordance with the distributional framework we use to represent aleatoric uncertainty. Hence, the distribution of returns admits the following decomposition:

$$Z^\pi(s,a) = \bar{Z}(s,a) + \epsilon\sigma(s,a), \tag{4}$$

where $\epsilon \sim \mathcal{N}(0,1)$. As we describe next, only $Z^\pi(s,a)$ is directly modeled (via a quantile representation), hence $\bar{Z}(s,a)$ and $\sigma(s,a)$ are unknown. The following theorem shows how to recover these components from $Z^\pi(s,a)$.

Proposition 1. *Let $\mathbb{E}_\epsilon$ be the expectation w.r.t. $\epsilon \sim \mathcal{N}(0,1)$. The quantile function $q_{\bar{Z}(s,a)}$ of $\bar{Z}$ is given by:*

$$q_{\bar{Z}(s,a)} = \mathbb{E}_{\epsilon,\sigma}\left[q_{Z^\pi(s,a)}\right] \tag{5}$$

where $q_{Z^\pi(s,a)}$ is the quantile function of $Z^\pi(s,a)$ knowing ϵ and $\sigma(s,a)$. Moreover, $\sigma^2(s,a)$ satisfies:

$$\sigma^2(s,a) = \mathbb{E}_\epsilon[\|\bar{Z}(s,a) - Z^\pi\|^2] \tag{6}$$

The proof is given in Appendix E.

Quantile approximation Following [17], TOP represents the return distribution $Z^\pi(s,a)$ using a *quantile approximation*, meaning that it estimates K statistics $q^{(k)}(s,a)$ as a discrete approximation of the quantiles of $Z^\pi(s,a)$. The quantiles $q^{(k)}(s,a)$ can be learned as the outputs of a parametric function—in our case, a deep neural network—with parameter vector ϕ. To measure epistemic uncertainty, TOP stores two estimates of the distribution $Z_1^\pi(s,a)$ and $Z_2^\pi(s,a)$ with respective quantile functions $q_1^{(k)}(s,a)$ and $q_2^{(k)}(s,a)$ and parameters ϕ_1 and ϕ_2. This representation allows for simple estimation of the mean $\bar{Z}(s,a)$ and variance $\sigma(s,a)$ in (4) using Proposition 1. Indeed, applying (5) and (6) and treating $Z_1^\pi(s,a)$ and $Z_2^\pi(s,a)$ as two draws from (4), we can approximate the quantiles $q_{\bar{Z}(s,a)}$ and $q_{\sigma(s,a)}$ of $\bar{Z}(s,a)$ and $\sigma(s,a)$ by

$$\bar{q}^{(k)}(s,a) = \frac{1}{2}\left(q_1^{(k)}(s,a) + q_2^{(k)}(s,a)\right), \qquad \sigma^{(k)}(s,a) = \sqrt{\sum_{i=1}^{2}\left(q_i^{(k)}(s,a) - \bar{q}^{(k)}(s,a)\right)^2}. \tag{7}$$

Next, we will show these approximations can be used to define an exploration strategy for the agent.

5.2 An uncertainty-based strategy for exploration

We use the quantile estimates of $\bar{Z}(s,a)$ and $\sigma(s,a)$ defined in (7) to construct a *belief distribution* $\tilde{Z}^\pi(s,a)$ over the expected return whose quantiles are defined by

$$q_{\tilde{Z}^\pi(s,a)} = q_{\bar{Z}(s,a)} + \beta q_{\sigma(s,a)}. \tag{8}$$

This belief distribution $\tilde{Z}^\pi(s,a)$ is said be *optimistic* when $\beta \geq 0$ and *pessimistic* when $\beta < 0$. The amplitude of optimism or pessimism is measured by $\sigma(s,a)$, which quantifies epistemic uncertainty. However, the degree of optimism depends on β and is adjusted dynamically during training as we will see in Section 5.3. Note that β replaces $\epsilon \sim \mathcal{N}(0,1)$, making the belief distribution non-Gaussian.

Learning the critics. TOP uses the belief distribution in (8) to form a target for both instances of the distribution, $Z_1^\pi(s,a)$ and $Z_2^\pi(s,a)$. To achieve this, TOP computes an approximation of $\tilde{Z}^\pi(s,a)$ using K quantiles $\tilde{q}^{(k)}$:

$$\tilde{q}^{(k)} = \bar{q}^k + \beta\sigma^{(k)}. \tag{9}$$

The temporal difference error for each $Z_i^\pi(s,a)$ is then given by $\delta_i^{(j,k)} := r + \gamma\tilde{q}^{(j)} - q_i^{(k)}$ with $i \in \{1,2\}$ and where (j,k) ranges over all possible combinations of quantiles. Finally, following the quantile regression approach in [17], we minimize the Huber loss $\mathcal{L}_{\text{Huber}}$ evaluated at each distributional error $\delta_i^{(j,k)}$, which provides a gradient signal to learn the distributional critics:

$$\Delta\phi_i \propto \sum_{1\leq k,j\leq K} \nabla_{\phi_i}\mathcal{L}_{\text{Huber}}(\delta_i^{(j,k)}). \tag{10}$$

This process is summarized in Algorithm 2.

Learning the actor. The actor is trained to maximize the expected value $\tilde{Q}(s,a)$ under the belief distribution $\tilde{Z}^\pi(s,a)$. Using the quantile approximation, $\tilde{Q}(s,a)$ is simply given as an average over $\tilde{q}^{(k)}$: $\tilde{Q}(s,a) = \frac{1}{K}\sum_{k=1}^{K} \tilde{q}^{(k)}(s,a)$. The update of the actor follows via the DPG gradient:

$$\Delta\theta \propto \nabla_a \tilde{Q}(s,a)|_{a=\pi_\theta(s)} \nabla_\theta \pi_\theta(s). \tag{11}$$

This process is summarized in Algorithm 3. To reduce variance and leverage past experience, both critic and actor updates in (10) and (11) are averaged over a N transitions $(s,a,r,s')_{n=1}^{N}$ sampled from a replay buffer $\mathcal{B}$ [34].

In the special case of $\beta = -1/\sqrt{2}$, the average of (8) reduces to $\min_i Z_i^\pi(s,a)$ and (11) recovers a distributional version of TD3, a pessimistic algorithm. On the other hand, when $\beta > 0$, the learning target is optimistic with respect to the current value estimates, recovering a similar procedure to a distributional version of the optimistic algorithm of [14]. However, in our case, when $\beta \geq 0$ the learning target is optimistic as well. Hence, (10) and (11) can be seen as a generalization of these works to a distributional framework that can recover both optimistic and pessimistic value estimation depending on the sign of β. Next, we propose a principled way to adapt β dynamically during training to benefit from the advantages of both pessimistic and optimistic approaches.

5.3 Optimism and Pessimism as a Multi-Arm Bandit Problem

As we have seen (see Figure 2), the optimal degree of optimism or pessimism for a given algorithm may vary across environments. As we shall see, it can also be beneficial to be more or less optimistic over the course of a single training run. It's therefore sensible for an agent to adapt its degree of optimism dynamically in response to feedback from the environment. As we established in the previous section, the agent can behave optimistically or pessimistically depending on the value of β in (8). The natural question, then, is how to adapt β to maximize performance.

An intuitive solution is to be optimistic when optimism has resulted in improved performance, and to be pessimistic when pessimism has resulted in better performance. To formalize this idea, we use the Exponentially Weighted Average Forecaster multi-armed bandit framework [13] to derive an adaptive algorithm, with each bandit arm representing a different value of β. In this setup, we consider D experts making recommendations at the beginning of each episode. After sampling a decision $d_m \in \{1,\dots,D\}$ at episode m from a distribution $\mathbf{p}^m \in \Delta_D$ with the form $\mathbf{p}^m(d) \propto \exp(w_m(d))$ the learner receives a feedback signal $f_m \in \mathbb{R}$. Δ_D denotes the space of categorical distributions with D dimensions. The distribution $\mathbf{p}^m$ is updated by updating w_m as follows:

$$w_{m+1}(d) = \begin{cases} w_m(d) + \eta \frac{f_m}{\mathbf{p}^m(d)} & \text{if } d = d_m \\ w_m(d) & \text{otherwise} \end{cases} \tag{12}$$

for some step size parameter η. Intuitively, we can see that if the feedback signal obtained is high and the current probability of selecting a given arm is low, the likelihood of selecting that arm again will increase. The arm index d_m selects a β from a pre-specified set $\{\beta_d\}_{d=1}^{D}$. For the feedback signal f_m, we use *improvement in performance*. Concretely, we set $f_m = R_m - R_{m-1}$, where R_m is the cumulative reward obtained in episode m. Henceforth, we denote by $\mathbf{p}_m^\beta$ the exponential weights distribution over β values at episode m.

Our approach can be thought of as implementing a form of model selection similar to that of [42], where instead of maintaining distinct critics for each optimism choice, we simply update the same pair of critics using the choice of β proposed by the bandit algorithm. For a more thorough discussion of TOP's connection to model selection, see Appendix D.

5.4 The TOP Framework

The general framework described can be applied to any off-policy actor-critic framework. As an example, the full procedure integrated with TD3 (TOP-TD3) is shown in Algorithm 1 with key differences from TD3 highlighted in purple. Like TD3, we apply target networks, which use slow-varying averages of the current parameters, θ, ϕ_1, ϕ_2, to provide stable updates for the critic functions. The target parameters $\theta', \phi_1', \phi_2'$ are updated every b time steps along with the policy. We use two critics, which has been shown to be sufficient for capturing epistemic uncertainty [14]. However, it is likely that the ensemble would be more effective with more value estimates, as demonstrated in [38].

Algorithm 1: TOP-TD3

Initialize critic networks Q_{ϕ_1}, Q_{ϕ_2} and actor π_θ
Initialize target networks $\phi'_1 \leftarrow \phi_1$, $\phi'_2 \leftarrow \phi_2$, $\theta' \leftarrow \theta$
Initialize replay buffer and bandit probabilities $\mathcal{B} \leftarrow \emptyset$, $\mathbf{p}_1^\beta \leftarrow \mathcal{U}([0,1]^D)$
for episode in $m = 1, 2, \ldots$ **do**
  Initialize episode reward $R_m \leftarrow 0$
  Sample optimism $\beta_m \sim \mathbf{p}_m^\beta$
  for time step $t = 1, 2, \ldots, T$ **do**
    Select noisy action $a_t = \pi_\theta(s_t) + \epsilon$, $\epsilon \sim \mathcal{N}(0, s^2)$, obtain r_{t+1}, s_{t+1}
    Add to total reward $R_m \leftarrow R_m + r_{t+1}$
    Store transition $\mathcal{B} \leftarrow \mathcal{B} \cup \{(s_t, a_t, r_{t+1}, s_{t+1})\}$
    Sample N transitions $\mathcal{T} = (s, a, r, s')_{n=1}^N \sim \mathcal{B}$.
    `UpdateCritics`$(\mathcal{T}, \beta_m, \theta', \phi'_1, \phi'_2)$
    if $t \mod b$ **then**
      `UpdateActor`$(\mathcal{T}, \beta_m, \theta, \phi_1, \phi_2)$
      Update ϕ'_i: $\phi'_i \leftarrow \tau\phi_i + (1-\tau)\phi'_i$, $i \in \{1, 2\}$
      Update θ': $\theta' \leftarrow \tau\theta + (1-\tau)\theta'$
  end for
  Update bandit $\mathbf{p}^\beta$ weights using (12)
end for

6 Experiments

The key question we seek to address with our experiments is whether augmenting state-of-the-art off-policy actor-critic methods with TOP can increase their performance on challenging continuous control benchmarks. We also test our assumption that the relative performance of optimistic and pessimistic strategies should vary across environments and during training and perform ablations to ascertain the relative contributions of different components of the framework to performance.

State-based control To answer our first question, we first augmented TD3 [20] with TOP (TOP-TD3) and evaluated its performance on seven state-based continuous control tasks from the MuJoCo framework [52] via OpenAI Gym [12]. As baselines, we also trained standard TD3 [20], SAC [21], OAC [14], as well as two ablations of TOP. The first, QR-TD3, is simply TD3 with distributional critics, and the second, non-distributional (ND) TOP-TD3 is our bandit framework applied to TD3 without distributional value estimation. TD3, SAC, and OAC use their default hyperparameter settings, with TOP and its ablations using the same settings as TD3. For tactical optimism, we set the possible β values to be $\{-1, 0\}$, such that $\beta = -1$ corresponds to a pessimistic lower bound, and $\beta = 0$ corresponds to simply using the average of the critic. It's important to note that $\beta = 0$ is an optimistic setting, as the mean is biased towards optimism. We also tested the effects of different settings for β (Appendix, Figure 6). Hyperparameters were kept constant across all environments. Further details can be found in Appendix B. We trained all algorithms for 1 million time steps and repeated each experiment with 10 random seeds.

Table 1: Final average reward over 10 trials on Mujoco control tasks, trained for 1M time steps. $\pm$ values denote one std across trials. All values within one std of the highest performance are listed in **bold**. $\star$ indicates gains over base TD3 are statistically significant ($p < 0.05$).

Task	TOP-TD3	ND TOP-TD3	QR-TD3	TD3	OAC	SAC
Humanoid	**5899±142***	5445	5003	5386	5349	5315
HalfCheetah	**13144 ± 701***	**12477**	11170	9566	11723	10815
Hopper	**3688 ± 33***	3458	3392	3390	2896	2237
Walker2d	**5111 ± 220***	4832	4560	4412	4786	4984
Ant	**6336 ± 181***	6096	5642	4242	4761	3421
InvDoublePend	**9337 ± 20***	**9330**	9299	8582	**9356**	**9348**
Reacher	**−3.85 ± 0.96**	**−3.91**	**−3.95**	**−4.22**	**−4.15**	**−4.14**

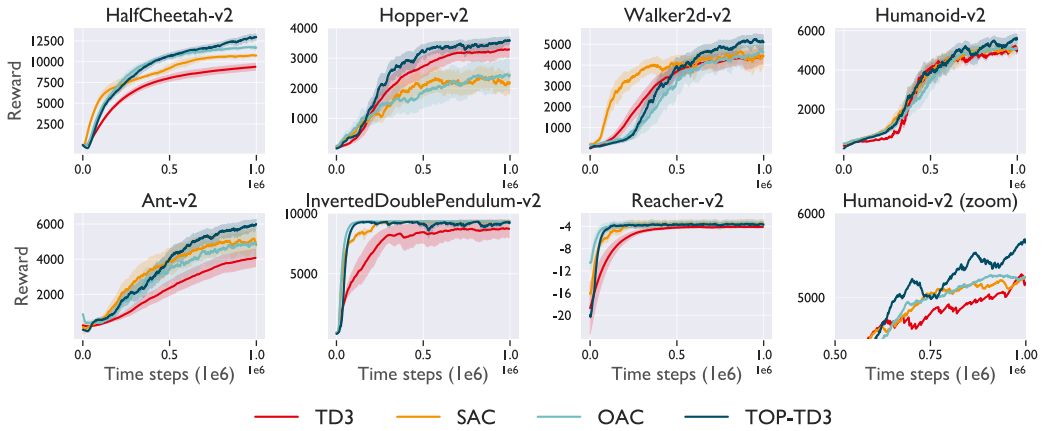

Figure 3: Reward curves for Mujoco tasks. The shaded region represents half a std over 10 runs. Curves are uniformly smoothed and the second half of learning without shading on Humanoid is highlighted for clarity.

Our results, displayed in Figure 3 and Table 1, demonstrate that TOP-TD3 is able to outperform or match baselines across all environments, with state-of-the-art performance in the 1M time step regime for the challenging Humanoid task. In addition, we see that TOP-TD3 matches the best optimistic and pessimistic performance for HalfCheetah and Hopper in Fig. 2. Without access to raw scores for all environments we cannot make strong claims of statistical significance. However, it is worth noting that the mean minus one standard deviation of TOP-RAD outperforms the mean performance all baselines in five out of the seven environments considered.

Table 2: Final average reward over 10 trials on DMControl tasks, trained for 100k time steps. $\pm$ values denote one std across trials. All values within one std of the highest performance are listed in **bold**. $\star$ indicates gains over base RAD are statistically significant ($p < 0.05$).

Task (100k)	**TOP-RAD**	**RAD**	**DrQ**	**PI-SAC**	**CURL**	**PlaNet**	**Dreamer**
Cheetah, Run	$\mathbf{674 \pm 31}^\star$	499	344	460	299	307	235
Finger, Spin	873 ± 69	813	901	**957**	767	560	341
Walker, Walk	$\mathbf{862 \pm 43}^\star$	644	612	514	403	221	277
Cartpole, Swing	$\mathbf{887 \pm 13}^\star$	864	759	816	582	563	326
Reacher, Easy	$\mathbf{991 \pm 3}^\star$	772	601	758	538	82	314
Cup, Catch	$\mathbf{970 \pm 12}^\star$	950	913	933	769	710	246
Task (500k)	**TOP-RAD**	**RAD**	**DrQ**	**PI-SAC**	**CURL**	**PlaNet**	**Dreamer**
Cheetah, Run	$\mathbf{910 \pm 4}^\star$	774	660	801	518	568	570
Finger, Spin	$\mathbf{928 \pm 74}$	**907**	**938**	$\mathbf{957}^*$	**926**	718	796
Walker, Walk	$\mathbf{988 \pm 4}^\star$	917	921	946	902	478	897
Cartpole, Swing	$\mathbf{890 \pm 28}^\star$	858	**868**	816^*	845	787	762
Reacher, Easy	$\mathbf{993 \pm 5}^\star$	930	942	950	929	588	793
Cup, Catch	$\mathbf{972 \pm 53}^\star$	**970**	**963**	$\mathbf{933}^*$	$\mathbf{959}^\star$	**939**	879

Pixel-based control We next consider a suite of challenging pixel-based environments, to test the scalability of TOP to higher dimensional tasks. We introduce TOP-RAD, a new algorithm dynamically switching between optimism and pessimism while using SAC with data augmentation (as in [30]). We evaluate TOP-RAD on both the 100k and 500k benchmarks on six tasks from the DeepMind (DM) Control Suite [50], a benchmark which has received a great deal of attention lately due to dramatic recent progress. In addition to the original RAD, we also report performance from DrQ [56], PI-SAC [32], CURL [31], PlaNet [23] and Dreamer [22], representing the state-of-the-art methods. All algorithms use their standard hyperparameter settings, with TOP using the same settings as in the state-based tasks, with no further tuning. We report results for both settings averaged over ten seeds (Table 2). As we can see, TOP-RAD sets a new state of the art in every task except one (Finger, Spin), and in that case there is still significant improvement compared to standard RAD. Note that this is a very simple method, requiring only the a few lines of change vs. vanilla RAD - and yet the gains over the baseline method are sizeable and statistically significant.

Does the efficacy of optimism vary across environments? To provide insight into how TOP's degree of optimism changes across tasks and over the course of learning, we plotted the average arm choice made by the bandit algorithm over time for each environment in Figure 4. Optimistic choices were given a value of 1 and and pessimistic selections were assigned 0. A mean of 0.5 indicates that $\beta = 0$ (optimism) and $\beta = -1$ (pessimism) were equally likely.

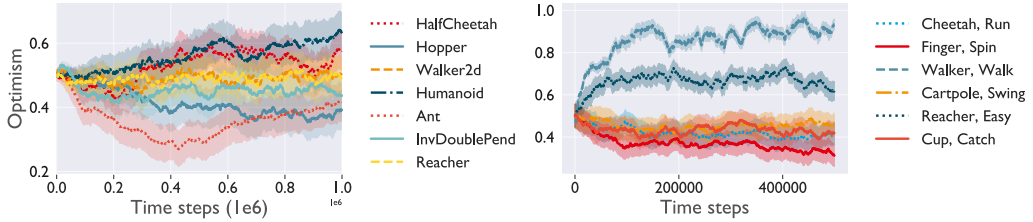

Figure 4: Mean optimism plotted across 10 seeds. The shaded areas represent one half std.

From the plot, we can see that in some environments (e.g., Humanoid and Walker, Walk), TOP learned to be more optimistic over time, while in others (e.g., Hopper and Finger, Spin), the agent became more pessimistic instead. Importantly, these changes were not always monotonic. On Ant, for example, TOP becomes steadily more pessimistic until around halfway through training, at which point it switches and grows more optimistic over time. The key question, then, is whether this flexibility contributes to improved performance?

Does tactical optimism matter? While we've established that TOP varies its degree of optimism over time and between tasks, it's unclear whether this ability actually confers a performance advantage. To test this, we compared TOP to two baselines, a "Pessimistic" version in which $\beta = -1$ for every episode, and an "Optimistic" version in which β is fixed to 0. If TOP is able to accurately gauge the degree of optimism that's effective for a given task, then it should match the best performing baseline in each task even if these vary. We tested this hypothesis in the HalfCheetah and Hopper environments, with the results in Figure 5. We can see TOP matches the Optimistic performance for HalfCheetah and the Pessimistic performance in Hopper, both of which outperform in the given environment. This aligns with Figure 4, where we see that TOP does indeed favor a more Optimistic strategy for HalfCheetah, with a more Pessimistic one for Hopper. This result can be seen as connected to the bandit regret guarantees referenced in Section 5.3, in which an adaptive algorithm is able to perform at least as well as the best *fixed* optimism choice in hindsight.

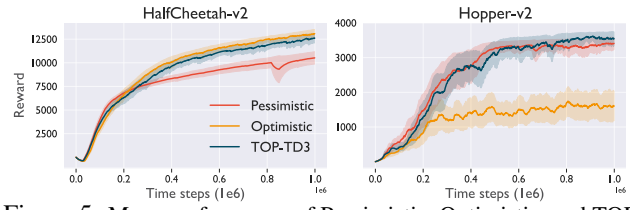

Figure 5: Mean performance of Pessimistic, Optimistic, and TOP across 10 seeds. Shaded regions are one half std.

7 Conclusion

We demonstrated empirically that differing levels of optimism are useful across tasks and over the course of learning. As previous deep actor-critic algorithms rely on a fixed degree of optimism, we introduce TOP, which is able to dynamically adapt its value estimation strategy, accounting for both aleatoric and epistemic uncertainty to optimize performance. We then demonstrate that TOP is able to outperform state-of-the-art approaches on challenging continuous control tasks while appropriately modulating its degree of optimism.

One limitation of TOP is that the available settings for β are pre-specified. It would be interesting to learn β, either through a meta-learning or Bayesian framework. Nevertheless, we believe that the bandit framework provides a useful, simple-to-implement template for adaptive optimism that could be easily be applied to other settings in RL. Other future avenues could involve adapting other parameters online, such as regularization [41, 36], constructing the belief distribution from more than two critics, and learning a weighting over quantiles rather than simply taking the mean. This would induce a form of optimism and/or pessimism specifically with respect to aleatoric uncertainty and has connections to risk-sensitive RL, as described by [16].

References

[1] A. Agarwal, H. Luo, B. Neyshabur, and R. E. Schapire. Corralling a band of bandit algorithms. In *Conference on Learning Theory*, pages 12–38. PMLR, 2017.

[2] J. Audibert, R. Munos, and C. Szepesvári. Tuning bandit algorithms in stochastic environments. In *Algorithmic Learning Theory, 18th International Conference, ALT 2007, Sendai, Japan, October 1-4, 2007, Proceedings*, volume 4754 of *Lecture Notes in Computer Science*, pages 150–165. Springer, 2007.

[3] M. G. Azar, I. Osband, and R. Munos. Minimax regret bounds for reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 263–272, 2017.

[4] A. P. Badia, B. Piot, S. Kapturowski, P. Sprechmann, A. Vitvitskyi, D. Guo, and C. Blundell. Agent57: Outperforming the atari human benchmark. In *Proceedings of the 37th International Conference on Machine Learning*. 2020.

[5] P. Ball, J. Parker-Holder, A. Pacchiano, K. Choromanski, and S. Roberts. Ready policy one: World building through active learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 591–601, 13–18 Jul 2020.

[6] P. J. Ball and S. J. Roberts. Offcon^3: What is state of the art anyway?, 2021.

[7] G. Barth-Maron, M. W. Hoffman, D. Budden, W. Dabney, D. Horgan, D. TB, A. Muldal, N. Heess, and T. Lillicrap. Distributional policy gradients. In *International Conference on Learning Representations*, 2018.

[8] P. L. Bartlett and A. Tewari. REGAL: A regularization based algorithm for reinforcement learning in weakly communicating MDPs. *CoRR*, abs/1205.2661, 2012.

[9] M. G. Bellemare, W. Dabney, and R. Munos. A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, page 449–458. JMLR.org, 2017.

[10] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *CoRR*, abs/1207.4708, 2012.

[11] R. I. Brafman and M. Tennenholtz. R-max - a general polynomial time algorithm for near-optimal reinforcement learning. *J. Mach. Learn. Res.*, 3(null):213–231, Mar. 2003.

[12] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym. *CoRR*, 2016.

[13] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games.* Cambridge University Press, 2006.

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

[15] J. D. Co-Reyes, Y. Miao, D. Peng, Q. V. Le, S. Levine, H. Lee, and A. Faust. Evolving reinforcement learning algorithms. In *International Conference on Learning Representations*, 2021.

[16] W. Dabney, G. Ostrovski, D. Silver, and R. Munos. Implicit quantile networks for distributional reinforcement learning. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1096–1105, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[17] W. Dabney, M. Rowland, M. G. Bellemare, and R. Munos. Distributional reinforcement learning with quantile regression. In *AAAI*. 2018.

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

[19] R. Fruit, M. Pirotta, A. Lazaric, and R. Ortner. Efficient bias-span-constrained exploration-exploitation in reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML*, volume 80, pages 1573–1581, 2018.

[20] S. Fujimoto, H. van Hoof, and D. Meger. Addressing function approximation error in actor-critic methods. 2018.

[21] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *CoRR*, abs/1801.01290, 2018.

[22] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

[23] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2555–2565. PMLR, 09–15 Jun 2019.

[24] H. Hasselt. Double q-learning. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23, pages 2613–2621, 2010.

[25] P. J. Huber. Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35(1):73–101, 1964.

[26] M. Jaderberg, V. Dalibard, S. Osindero, W. M. Czarnecki, J. Donahue, A. Razavi, O. Vinyals, T. Green, I. Dunning, K. Simonyan, C. Fernando, and K. Kavukcuoglu. Population based training of neural networks. *CoRR*, abs/1711.09846, 2017.

[27] T. Jaksch, R. Ortner, and P. Auer. Near-optimal regret bounds for reinforcement learning. *J. Mach. Learn. Res.*, 11:1563–1600, Aug. 2010.

[28] C. Jin, Z. Yang, Z. Wang, and M. I. Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143. PMLR, 2020.

[29] L. Kocsis and C. Szepesvári. Bandit based Monte-Carlo planning. In *Machine Learning: ECML 2006, 17th European Conference on Machine Learning, Berlin, Germany, September 18-22, 2006, Proceedings*, volume 4212 of *Lecture Notes in Computer Science*, pages 282–293. Springer, 2006.

[30] M. Laskin, K. Lee, A. Stooke, L. Pinto, P. Abbeel, and A. Srinivas. Reinforcement learning with augmented data. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19884–19895. Curran Associates, Inc., 2020.

[31] M. Laskin, A. Srinivas, and P. Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *Proceedings of the 37th International Conference on Machine Learning, Vienna, Austria, PMLR 119*, 2020. arXiv:2004.04136.

[32] K.-H. Lee, I. Fischer, A. Liu, Y. Guo, H. Lee, J. Canny, and S. Guadarrama. Predictive information accelerates learning in rl. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 11890–11901. Curran Associates, Inc., 2020.

[33] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. In *ICLR*, 2016.

[34] L.-J. Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Mach. Learn.*, 8(3–4):293–321, May 1992.

[35] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, Feb. 2015.

[36] T. Moskovitz, M. Arbel, F. Huszar, and A. Gretton. Efficient wasserstein natural gradients for reinforcement learning. In *International Conference on Learning Representations*. 2021.

[37] J. Oh, M. Hessel, W. M. Czarnecki, Z. Xu, H. van Hasselt, S. Singh, and D. Silver. Discovering reinforcement learning algorithms. In *Advances in Neural Information Processing Systems 33*. 2020.

[38] I. Osband, C. Blundell, A. Pritzel, and B. Van Roy. Deep exploration via bootstrapped dqn. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29, pages 4026–4034, 2016.

[39] A. Pacchiano, P. Ball, J. Parker-Holder, K. Choromanski, and S. Roberts. On optimism in model-based reinforcement learning. *CoRR*, 2020.

[40] A. Pacchiano, C. Dann, C. Gentile, and P. Bartlett. Regret bound balancing and elimination for model selection in bandits and rl. *arXiv preprint arXiv:2012.13045*, 2020.

[41] A. Pacchiano, J. Parker-Holder, Y. Tang, K. Choromanski, A. Choromanska, and M. Jordan. Learning to score behaviors for guided policy optimization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 7445–7454, 13–18 Jul 2020.

[42] A. Pacchiano, M. Phan, Y. Abbasi-Yadkori, A. Rao, J. Zimmert, T. Lattimore, and C. Szepesvari. Model selection in contextual stochastic bandit problems. *arXiv preprint arXiv:2003.01704*, 2020.

[43] J. Parker-Holder, A. Pacchiano, K. Choromanski, and S. Roberts. Effective diversity in population-based reinforcement learning. In *Advances in Neural Information Processing Systems 34*. 2020.

[44] H. Penedones, C. Riquelme, D. Vincent, H. Maennel, T. A. Mann, A. Barreto, S. Gelly, and G. Neu. Adaptive temporal-difference learning for policy evaluation with per-state uncertainty estimates. In *Advances in Neural Information Processing Systems*. 2019.

[45] M. Rowland, R. Dadashi, S. Kumar, R. Munos, M. G. Bellemare, and W. Dabney. Statistics and samples in distributional reinforcement learning, 2019.

[46] T. Schaul, D. Borsa, D. Ding, D. Szepesvari, G. Ostrovski, W. Dabney, and S. Osindero. Adapting behaviour for learning progress. *CoRR*, abs/1912.06910, 2019.

[47] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[48] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller. Deterministic policy gradient algorithms. In *Proceedings of the 31st International Conference on Machine Learning*, pages 387–395, 2014.

[49] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

[50] Y. Tassa, Y. Doron, A. Muldal, T. Erez, Y. Li, D. de Las Casas, D. Budden, A. Abdolmaleki, J. Merel, A. Lefrancq, T. Lillicrap, and M. Riedmiller. Deepmind control suite, 2018.

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

[52] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *IROS*, pages 5026–5033. IEEE, 2012.

[53] A. C. Y. Tossou, D. Basu, and C. Dimitrakakis. Near-optimal optimistic reinforcement learning using empirical bernstein inequalities. *CoRR*, abs/1905.12425, 2019.

[54] C. J. C. H. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.

[55] L. Yang and M. Wang. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning*, pages 10746–10756. PMLR, 2020.

[56] D. Yarats, I. Kostrikov, and R. Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International Conference on Learning Representations*, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 7 and Section 5.4.
   (c) Did you discuss any potential negative societal impacts of your work? [No] We present a generic framework for improving deep actor-critic algorithms, and so the societal impacts, if any, are difficult to predict.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] Our work has no evident ethical ramifications for the categories listed in the guidelines, nor does it contain or use any private information about individuals.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 5.1.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix E.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Section 6, Appendix B, and attached code.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 6 and Appendix B.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Section 6 and Appendix A.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix B.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See citations throughout the paper.
   (b) Did you mention the license of the assets? [Yes] See Appendix B.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We included the code for our experiments in the supplemental material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]