A Case for Validation Buffer in Pessimistic Actor-Critic

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

Mateusz Ostaszewski
Warsaw University of Technology
mm.ostaszewski@gmail.com

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

Abstract

In this paper, we investigate the issue of error accumulation in critic networks updated via pessimistic temporal difference objectives. We show that the critic approximation error can be approximated via a recursive fixed-point model similar to that of the Bellman value. We use such recursive definition to retrieve the conditions under which the pessimistic critic is unbiased. Building on these insights, we propose Validation Pessimism Learning (VPL) algorithm. VPL uses a small validation buffer to adjust the levels of pessimism throughout the agent training, with the pessimism set such that the approximation error of the critic targets is minimized. We investigate the proposed approach on a variety of locomotion and manipulation tasks and report improvements in sample efficiency and performance.

1 Introduction

Approximation errors, although ubiquitous in machine learning, are particularly exaggerated in the context of value-based Reinforcement Learning (RL). Such exaggeration stems from Temporal Difference (TD) in which the critic is supervised via value estimate calculated at a different state (Silver et al., 2014; Mnih et al., 2015; Barth-Maron et al., 2018). Inaccuracies in this estimate lead to propagated errors in state-action updates, and the use of maximization in value estimation inherently promotes overestimation. Addressing such overestimation has proven to be an effective strategy in discrete and continuous action environments (Hasselt, 2010; Van Hasselt et al., 2016; Hessel et al., 2018). Clipped Double Q-Learning (CDQL), a common solution to overestimation in continuous action actor-critic algorithms aims to mitigate overestimation by balancing errors against a pessimistic lower bound value approximation (Fujimoto et al., 2018). However, challenges arise if the lower bound is insufficiently pessimistic, leading to continued overestimation, or overly pessimistic, causing underestimation (Moskovitz et al., 2021; Cetin & Celiktutan, 2023). The latter, though less recognized, can significantly reduce sample efficiency and degrade actor-critic agents' performance in both low and high replay ratio settings which we show in Figure 1.

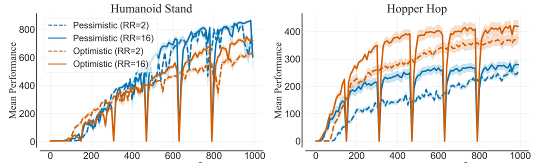


Figure 1: Pessimism can yield improvements exceeding increased replay ratio and full-parameter resets. The pessimism is better Humanoid, whereas the optimistic approach dominates Hopper. 10 seeds and 95% CI.

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

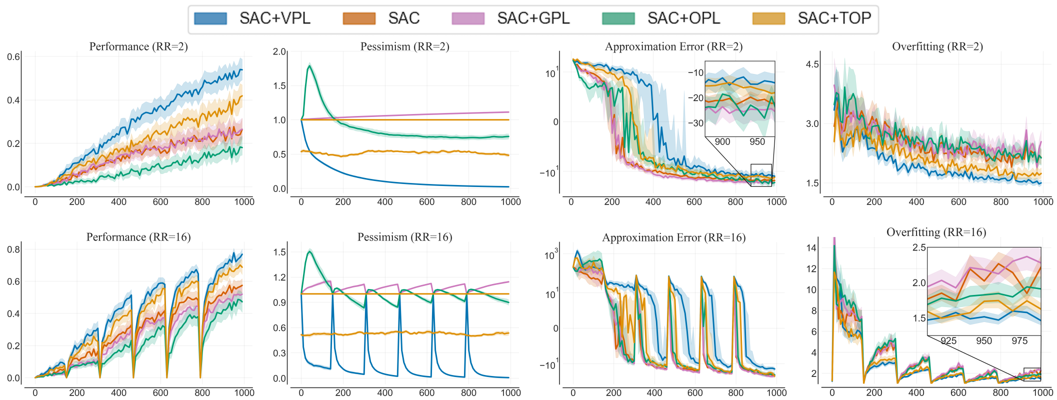


Figure 2: We integrate the Soft Actor-Critic (SAC) and the Scaled-By-Resetting SAC (SR-SAC) with various pessimism adjustment algorithms. Performance is evaluated in both low replay (top row) and high replay (bottom row) regimes. All algorithms use the same network architectures and hyperparameter settings, and performance differences arise solely from the pessimism adjustment. Despite similar motivations, methods exhibit different levels of pessimism. Our proposed Validation Pessimism Learning (VPL) demonstrates the lowest approximation error and mitigates value overfitting more effectively than other approaches, leading to improvements in sample efficiency. The experimental setting is detailed in Sections 5 and E. Results are based on 20 tasks with 10 seeds per task, presented as interquartile mean (IQM) and 95% confidence intervals (CI).

In this paper, we investigate the relationship between pessimism in Q-value approximation and error accumulation in critic networks. We start by characterization of existing strategies for online pessimism adjustment. Furthermore, we analyze the pessimistic critic approximation error and show that such error can be represented recursively forming a fixed-point model, akin to values and Q-values. This recursive representation helps us highlight the bias inherent in pessimistic actor-critic algorithms, examine their convergence dynamics, and identify the conditions under which pessimistic critics can achieve zero error in value approximation. Building on these insights, we propose the Validation Pessimism Learning (VPL) algorithm. VPL employs a small validation replay buffer to adjust the pessimism levels online, aiming to minimize the approximation error of critic targets while preventing overfitting to accumulated experience. We evaluate VPL against existing pessimism adjustment methods on DeepMind control (Tassa et al., 2018) and single-task MetaWorld (Yu et al., 2020a) platforms. Our findings demonstrate that VPL not only achieves performance improvements but also exhibits less sensitivity to hyperparameter settings compared to the baseline algorithms. We summarize our contributions below:

- We show that critic approximation error can be defined recursively through a fixed-point model. We demonstrate that pessimistic TD learning, a method often used in continuous action RL, converges to the true value under strict conditions.
- We present an empirical analysis showing that the performance loss associated with not including every transition in the replay buffer diminishes as training progresses. This observation challenges the traditional belief that every transition must be used in value learning for sample-efficient RL and builds a case for employing a validation buffer in an online RL setting.
- We propose VPL, an algorithm that uses a small validation buffer for online adjustment of pessimism associated with lower bound Q-value approximation. We test the effectiveness of VPL and other pessimism adjustment strategies in low and high replay regimes. We show that VPL offers performance improvements across a variety of tasks.

2 Background

2.1 Maximum Entropy Reinforcement Learning

We analyze an infinite-horizon Markov Decision Process (MDP) (Puterman, 2014), represented by the tuple (S, A, r, p_0, γ). In this model, both states S and actions A are continuous. The transition reward is given by $r_{s,a}$, $p_0(s)$ defines the initial state distribution, and $\gamma \in (0, 1]$ is the discount factor.

The policy, $\pi(a|s)$, is a distribution of actions conditioned on states. At any given state, the policy entropy is denoted as $\mathcal{H}(s)$. In an MDP where all states are positive recurrent, a policy-induced discounted stationary distribution $p_\gamma(s|\pi)$ also exists. The goal of Maximum Entropy Reinforcement Learning (MaxEnt RL) (Ziebart et al., 2008; Haarnoja et al., 2017) is to devise a policy that optimizes the expected cumulative sum of discounted returns and entropy.

$$\pi^* = \arg\max_\pi \mathop{\mathrm{E}}_{p_0,\pi} \sum_{t=0}^{\infty} \gamma^t \big(r_{s_t,a_t} + \alpha\mathcal{H}(s_t)\big), \tag{1}$$

where α denotes the temperature parameter which balances the reward and entropy objectives (Haarnoja et al., 2018). Soft Q-value is defined as the expected discounted return from performing an action at a given state and then following the policy $Q^\pi(s,a) = r_{s,a} + \gamma V^\pi(s')$. Soft value, denoted as $V^\pi(s)$ is calculated as follows:

$$V^\pi(s) = \mathop{\mathrm{E}}_{\pi} \big(Q^\pi(s,a) - \alpha \log \pi(a|s)\big). \tag{2}$$

In this context, the term $\log \pi(a|s)$ corresponds to the entropy objective, with $-\mathrm{E}_\pi \log \pi(a|s) = \mathcal{H}(s)$. In algorithms like Soft Actor-Critic (SAC), the policy and Q-value functions are modeled via parameterized function approximators, commonly referred to as the actor and critic, respectively (Silver et al., 2014). The parameters of these components are iteratively updated through gradient descent, following objectives derived from the policy iteration algorithm (Haarnoja et al., 2018). In continuous actor-critic algorithms, the policy parameters θ are updated such that the policy maximized the value approximate at states s, which are sampled from an off-policy replay buffer $\mathcal{D}$:

$$\theta^* = \arg\max_\theta \mathop{\mathrm{E}}_{\mathcal{D}} V^{lb}_\phi(s), \tag{3}$$

where $V^{lb}_\phi(s)$ is the approximate value lower bound calculated via the critic network (Haarnoja et al., 2018) and $s \sim \mathcal{D}$. Similarly, the critic parameters ϕ are updated in the policy evaluation step by minimizing temporal-difference variant (Ciosek & Whiteson, 2020):

$$\phi^* = \arg\min_\phi \mathop{\mathrm{E}}_{\mathcal{D}} \big(Q_\phi(s,a) - r_{s,a} - \bigtriangleup \gamma V^{lb}_\phi(s')\big)^2. \tag{4}$$

Above, we denote the critic outputs for a given state-action as $Q_\phi(s,a)$, $s, a, s' \sim \mathcal{D}$ and use $\bigtriangleup$ to denote the stop gradient operator. Modern actor-critic algorithms leverage a variety of countermeasures to overestimation of Q-value targets, with bootstrapping using target network (Van Hasselt et al., 2016) and Clipped Double Q-Learning (CDQL) (Fujimoto et al., 2018) being most prominent. In CDQL, the algorithm maintains an ensemble of critics to approximate the value lower bound:

$$\begin{aligned} &V^{lb}_\phi(s) \approx Q^{lb}_\phi(s,a) - \alpha \log \pi_\theta(a|s) \quad \text{with } a \sim \pi_\theta, \\ &Q^{lb}_\phi(s,a) = \min\big(Q^1_\phi(s,a), Q^2_\phi(s,a)\big), \end{aligned} \tag{5}$$

where $Q^{lb}_\phi(s,a)$ denotes the Q-value lower bound and $Q^i_\phi(s,a)$ denotes the $i-th$ critic in the critic ensemble. The CDQL was generalized by noticing relation between the minimum operator and ensemble statistics (Ciosek et al., 2019; Moskovitz et al., 2021; Cetin & Celiktutan, 2023):

$$Q^{lb}_\phi(s,a) = Q^\mu_\phi(s,a) - \beta Q^\sigma_\phi(s,a). \tag{6}$$

We denote the critic ensemble mean and standard deviation as Q^μ_ϕ and Q^σ_ϕ respectively. In particular, for $\beta = 1$ the above rule is exactly equal to the CDQL (Ciosek et al., 2019; Cetin & Celiktutan, 2023). Such lower bound updates the actor-critic parameters in the direction corrected by the critic ensemble disagreement. Such targets are referred to as *pessimistic* with the parameter β called *pessimism*.

2.2 Pessimism Adjustment

The success of pessimistic updates in practice has led to various methods for adjusting pessimism online. These techniques aim to improve the performance and efficiency of the agent by reducing the error in critic approximation. Algorithms such as On-policy Pessimism Learning (OPL) (Kuznetsov et al., 2021) and Generalized Pessimism Learning (GPL) (Cetin & Celiktutan, 2023) estimate this error and modify pessimism accordingly. Specifically, GPL views the adjustment of pessimism as a dual optimization problem, resulting in the following update rule:

$$
\begin{aligned}
&\beta = \arg\min_{\beta} \underset{p_0,\pi}{\mathrm{E}}\, \beta \left(Q^{\pi}(s,a) - r_{s,a} - \triangle\, \gamma V_{\phi}^{lb}(s')\right), \\
&V_{\phi}^{lb}(s') \approx Q_{\phi}^{\mu}(s,a) - \beta Q_{\phi}^{\sigma}(s,a) - \alpha \log \pi_{\theta}(a|s).
\end{aligned} \tag{7}
$$

In this context, $\beta \in (0, \infty)$ is a continuous parameter defining the level of pessimism and the true Q-value is represented by $Q^{\pi}(s,a)$. Since the term is not squared, β cannot be trivially optimized by setting it to zero. GPL and OPL focus on aligning pessimism with the error in the pessimistic objective approximation. Since the true Q-values are unknown, they must be estimated. GPL assumes that the critic's output is unbiased for off-policy actions (ie. $Q^{\pi}(s,a) = Q_{\phi}(s,a)$) and calculates the dual optimization pessimism loss using transitions from the replay buffer. However, this approach can lead to overfitting as it relies heavily on the critic output. In contrast, OPL estimates $Q^{\pi}(s,a)$ via λ-returns calculated using recent transitions, bootstrapped by the critic, which reduces the risk of overfitting. Nevertheless, due to frequent policy updates, even recent transitions may be off-policy. A general limitation of the dual optimization method is that the pessimism adjustment does not correlate with the critic disagreement for specific state-action pairs thus impairing the impact of potential changes to β. A different strategy, Tactical Optimism and Pessimism (TOP) (Moskovitz et al., 2021), adjusts pessimism using an external bandit controller to maximize online episodic rewards. However, this controller is discrete and less effective as possible amount of pessimism values are increased. The further discuss the existing approaches for online pessimism adjustment in Appendix C and summarize key characteristics in Table 2.

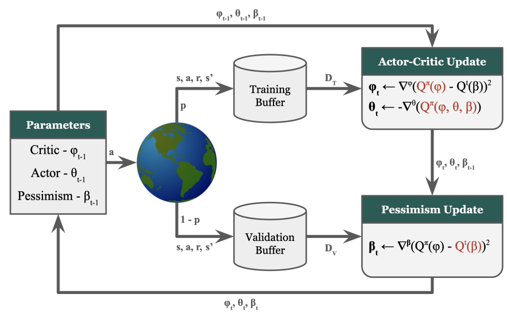


Figure 3: High level overview of the proposed approach. After environment step, the transition is stored in either the training buffer (used for updating actor-critic modules) or the validation buffer (used for updating pessimism module). The pessimism is updated via a "reverse" TD loss, optimisation of which on the training buffer would be prone to overfitting.

3 Approximation Error and Pessimism

In this section, we focus on the analysis of critic approximation errors within the framework of pessimistic updates. For simplicity, we consider a fixed policy π_{θ} and use $V(s)$ and $Q(s,a)$ to represent the value and Q-value under this policy. We define the mean and lower bound approximation errors denoted as U_{ϕ}^{μ} and U_{ϕ}^{lb} respectively:

$$
\begin{aligned}
U_{\phi}^{\mu}(s,a) &\triangleq Q(s,a) - Q_{\phi}^{\mu}(s,a), \\
U_{\phi}^{lb}(s,a) &\triangleq Q(s,a) - Q_{\phi}^{lb}(s,a).
\end{aligned} \tag{8}
$$

Here, $Q(s,a)$ denotes the true Q-value, the term $Q_{\phi}^{\mu}(s,a)$ represents the mean Q-value estimated by an ensemble of k critics, calculated as $Q_{\phi}^{\mu}(s,a) = \frac{1}{k}\sum^{k} Q_{\phi}^{i}(s,a)$, and $Q_{\phi}^{lb}(s,a)$ is the lower bound Q-value as defined in Equation 6. Additionally, we introduce the mean and lower bound temporal critic errors, denoted as u_{ϕ}^{μ} and u_{ϕ}^{lb}, respectively:

$$
\begin{aligned}
u_{\phi}^{\mu}(s,a,s') &\triangleq r_{s,a} + \gamma V_{\phi}^{\mu}(s') - Q_{\phi}^{\mu}(s,a), \\
u_{\phi}^{lb}(s,a,s') &\triangleq r_{s,a} + \gamma V_{\phi}^{lb}(s') - Q_{\phi}^{\mu}(s,a).
\end{aligned} \tag{9}
$$

These temporal critic errors quantify the deviation between the Q-values $Q_{\phi}^{\mu}(s,a)$ and the mean or lower bound Temporal Difference (TD) targets. The value $V_{\phi}^{lb}(s)$ is equal to the expected value of $Q_{\phi}^{lb}(s,a)$ over all state-action pairs under policy π, such that $V_{\phi}^{lb}(s) = \mathrm{E}_{\pi} Q_{\phi}^{lb}(s,a) - \log \pi_{\theta}(a|s)$.

Lemma 3.1 (Approximation error operator)**.** *Given policy π, k on-policy q-value approximations $Q_{\phi}^{1}, Q_{\phi}^{2}, ..., Q_{\phi}^{k}$, sample mean Q_{ϕ}^{μ} and standard deviation Q_{ϕ}^{σ}, the mean and lower bound approxima-*

tion errors follow a recursive formula:

$$U^{\mu}_{\phi}(s,a) = u^{\mu}_{\phi}(s,a,s') + \gamma \mathop{\mathrm{E}}_{a'\sim\pi} U^{\mu}_{\phi}(s',a'),$$
$$U^{lb}_{\phi}(s,a) = u^{lb}_{\phi}(s,a,s') + \beta Q^{\sigma}_{\phi}(s,a) + \gamma \mathop{\mathrm{E}}_{a'\sim\pi} U^{lb}_{\phi}(s',a'),$$
$$U^{lb}_{\phi}(s,a) = U^{\mu}_{\phi}(s,a) + \beta Q^{\sigma}_{\phi}(s,a).$$

We expand on Lemma 3.1 in Appendix B. The lemma reveals that approximation errors exhibit a recurrent pattern analogous to Q-values. Specifically, the temporal errors function as an immediate signal, akin to rewards, while the future approximation errors serve as the bootstrap signal. Furthermore, this observation formalizes the intuitive concept that minimizing the lower-bound approximation error necessitates a precise calibration of the pessimistic correction against the temporal error and the approximation errors of subsequent states. It can be shown that similarly to the Bellman operator, both mean and lower bound error approximation operators are monotonic contractions:

Theorem 3.2 (Approximation error contraction)**.** *Let $\mathcal{F}$ be the space of functions on domain $S \times A$. We define the mean error and lower bound error operators $\mathcal{U}^{\mu}, \mathcal{U}^{lb} : \mathcal{F} \to \mathcal{F}$ as:*

$$\mathcal{U}^{\mu}\big(f(s,a)\big) \triangleq u^{\mu}_{\phi}(s,a,s') + \gamma \mathop{\mathrm{E}}_{a'\sim\pi} f(s',a'),$$
$$\mathcal{U}^{lb}\big(f(s,a)\big) \triangleq u^{lb}_{\phi}(s,a,s') + \beta Q^{\sigma}_{\phi}(s,a) + \gamma \mathop{\mathrm{E}}_{a'\sim\pi} f(s',a').$$

Above, $f(s,a) : S \times A \to \mathbb{R}$ represents an estimate of the approximation error. Then it follows that both $\mathcal{U}^{\mu}$ and $\mathcal{U}^{lb}$ are monotonic contractions for any f_1 and f_2:

$$||\mathcal{U}(f_1) - \mathcal{U}(f_2)||_{\infty} \leq \gamma ||f_1 - f_2||_{\infty}.$$

We provide the relevant derivations in Appendix B. As follows from Theorem 3.2, repeated application of the approximation error operator yields a Cauchy sequence, and therefore leads to a fixed point:

Corollary 3.3 (Approximation error fixed point)**.** *We denote repeated k applications of either approximation error operator to function f as $\mathcal{U}_k(f)$. Then, due to Banach fixed point theorem:*

$$\mathcal{U}^{\infty}(f) = f^* \ \wedge \ \mathcal{U}(f^*) = f^*.$$

The corollary shows that the approximation error of values can be effectively modeled using a fixed-point approach, analogous to the treatment of values themselves. The potential ramifications and applications of this concept are further explored in Appendix B. Principally, the convergence of a pessimistic value model signifies that the approximation errors converge to zero, implying $U^{\mu}_{\phi} = U^{lb}_{\phi} = 0$. The convergence proof of CDQL indicates that the value model should align with the true on-policy values under the conventional Q-learning convergence assumptions (Watkins & Dayan, 1992; Fujimoto et al., 2018). Lemma 3.1 explicitly shows that for all s, a and s', both approximation errors equate to zero iff the following conditions are satisfied:

$$Q^{\mu}_{\phi}(s,a) = r + \gamma V^{\mu}_{\phi}(s') \ \wedge \ \beta Q^{\sigma}_{\phi}(s,a) = 0. \tag{10}$$

Consequently, the convergence of a pessimistic model necessitates either the absence of critic ensemble disagreement (i.e., $Q^{\sigma}_{\phi}(s,a) = 0$ for all state-action pairs) or an algorithmic ability to diminish the level of pessimism over time, culminating in $\beta = 0$ asymptotically. Figure 10 shows that the critic disagreement does not completely diminish on popular DeepMind Control and MetaWorld benchmarks. Given the improbability of achieving zero critic disagreement in overparameterized deep RL contexts, the adjustment of β emerges as a compelling strategy. Additionally, it can be demonstrated that under the scenario of critic underestimation, the lower-bound approximation error exceeds the mean approximation error:

$$U^{\mu}_{\phi}(s,a) > 0 \implies |U^{\mu}_{\phi}(s,a)| \leq |U^{lb}_{\phi}(s,a)|. \tag{11}$$

As follows, pessimistic learning is advantageous only in overestimation, whereas it becomes detrimental in cases of underestimation. To this end, the pessimism levels should be adjusted in tandem with changes in the approximation errors. In practical terms, achieving a zero approximation error for either mean or lower bound is an unrealistic. Given that $U_{\phi}(s,a) \in \mathbb{R}$, one might be interested in optimization of norm of $U^{\mu}_{\phi}(s,a)$ or $U^{lb}_{\phi}(s,a)$. This leads to the possibility of defining an "optimal" level of pessimism, where optimality is considered in relation to minimizing the respective approximation error norm. We note that our analysis yields a different approach as compared to the method derived from dual optimization Cetin & Celiktutan (2023), which we discuss in Section 2.2.

4 Validation Pessimism Learning Algorithm

Building on the analysis conducted in the previous Section, we propose Validation Pessimism Learning module (VPL). The goal of the VPL module is to adjust the pessimism parameter such that the critic targets (lower bound Q-value approximation) has the least approximation error. As such, VPL can be used as an alternative to CDQL or GPL in conjuction with any off-policy actor-critic algorithm. For our analysis, we utilize the Soft Actor-Critic (SAC) (Haarnoja et al., 2018) as the backbone algorithm. VPL is based on a simple premise of adjusting pessimism via a TD loss. Given that the critic concurrently optimizes this loss function, such setup is especially prone to overfitting. To mitigate this, the optimization of the pessimism parameter is conducted on a distinct set of *validation* data, which remains unseen by the actor-critic modules. From a theoretical standpoint, VPL can be interpreted as a strategy for pessimism model selection, with the selection process aimed at minimizing the lower bound approximation error delineated in the previous section. A critical aspect of VPL involves conducting the pessimism model selection on validation data. The model selection is achieved through gradient-based optimization of the proposed pessimism loss. The utilization of validation data in this process reduces the probability of overfitting to bootstrapped supervision signals used by TD learning. We summarize VPL approach in Figure 3 and share pseudo-code in Section B.3, where we colour changes wrt. regular SAC.

4.1 Validation Buffer

The employment of validation data is a well-established practice in supervised learning frameworks (Bishop & Nasrabadi, 2006). It serves a dual purpose: providing an unbiased assessment of model performance trained on the training dataset, and facilitating regularization techniques such as early stopping (Prechelt, 2002) or hyperparameter tuning (Bergstra & Bengio, 2012). However, the integration of validation data entails a trade-off, notably the reduction of the training set size. In supervised learning, the regret associated with decreasing the training set can be quantitatively evaluated through the lens of neural scaling laws (Rosenfeld et al., 2019). Such regret is, to the best of our knowledge, a relatively understudied area in the context of online RL. In online RL, the notion of a validation buffer is not popular, primarily due to the requisite sacrifice of actor-critic learning on the validation transitions. Given inherent sample inefficiency of RL, this cost is often deemed as overly burdening. Contrary to supervised learning setup, RL is characterized by a high correlation between successive samples, thereby diminishing the marginal utility of processing additional samples from the same trajectory. Consequently, we posit that in online RL, the cost associated with the use of validation data can be counterbalanced, provided the validation data is leveraged to enhance the learning process. In the case of VPL, we allocate the validation transitions exclusively for the adjustment of the pessimism parameter.

4.2 Pessimism Update Rule

The persistence of critic disagreement throughout training implies that the standard convergence guarantees of the pessimistic temporal difference update towards on-policy values are not upheld when $\beta \neq 0$. Moreover, in cases where minimizing the mean approximation error is not achievable, particularly in scenarios characterized by strong overestimation, the presence of non-zero critic disagreement can be leveraged to decrease the lower bound approximation error by increasing β. This observation forms the basis for our proposed method of adjusting β. The aim is to minimize the expected lower bound approximation error $U_\phi^{lb}(s,a)$, formulated as follows:

$$\beta^* = \arg\min_{\beta} \mathop{\mathrm{E}}_{p_0,\pi} \sum_{t=0}^{\infty} \gamma^t U_\phi^{lb}(s,a). \tag{12}$$

Unfortunately, obtaining $U_\phi^{lb}(s,a)$ is challenging as it necessitates an estimate of the true on-policy Q-value. Typically, such estimates are derived through methods like Monte-Carlo (MC) rollouts, TD(n), or TD(λ), with MC being the only unbiased method. However, in the context of off-policy learning or non-terminating environments, employing MC rollouts is impractical. Consequently, we leverage the simple approach proposed by Cetin & Celiktutan (2023) in which it is assumed that the critic output for prerecorded off-policy actions is unbiased. Therefore, we assume that $Q^\pi(s,a) = Q_\phi^\mu(s,a)$ for actions that do not maximize the output of the policy. Additionally, akin to the approach in off-policy actor-critic algorithms, the policy-induced distribution is approximated using an off-policy replay buffer. This approach leads to the formulation of the following:

$$\beta^* \approx \arg\min_{\beta} \mathop{\mathrm{E}}_{\mathcal{D}_v} \big(Q^{\mu}_{\phi}(s,a) - r_{s,a} - \gamma V^{lb}_{\phi}(s')\big)^2. \tag{13}$$

In this formulation, $\mathcal{D}_V$ represents the validation replay buffer, with s, a, s' denoting transitions sampled from this buffer. In line with other stochastic policy algorithms, we approximate value with the critic output for a single action $a' \sim \pi_\theta(a'|s')$. As follows, VPL adjusts the pessimism under assumption that $Q^{\mu}_{\phi}(s,a)$ is a good representation of $Q^{\pi}(s,a)$. Since the actions at which $Q^{\mu}_{\phi}(s,a)$ is evaluated are sampled from the validation buffer and are off-policy, these actions are likely to produce less overestimation than the adversarial actions sampled from a value-maximizing policy. Since $Q^{\mu}_{\phi}(s,a)$ is assumed to be unbiased, VPL thus generally reduces β over the training unless the approximated lower bound value evaluated at on-policy actions is systematically larger than the mean value evaluated at off-policy actions (ie. $Q^{\mu}_{\phi}(s,a)$). This approach contrasts with General Pessimism Learning (GPL) in that it allows for gradient flow through the lower bound approximation, thereby enabling adjustments to β that are in proportion to the level of critic disagreement. Moreover, by computing the pessimism loss on validation samples, which are not utilized by the actor-critic modules, we mitigate the risk of overfitting to the experienced data which we show on Figure 2.

5 Experiments

Our experiments are based on the JaxRL codebase (Kostrikov, 2021). Since all considered algorithms use SR-SAC (D'Oro et al., 2022) as their backbonce, we align the common hyperparameters with those recommended for Scaled-By-Resetting SAC (SR-SAC) as per D'Oro et al. (2022). This includes using the same network architectures and a two-critic ensemble, in accordance with established practices (Fujimoto et al., 2018; Haarnoja et al., 2018; Ciosek et al., 2019; Moskovitz et al., 2021; Cetin & Celiktutan, 2023). We conduct our experiments in two environments: the DeepMind Control (DMC) suite (Tassa et al., 2018) and the single-task MetaWorld (Yu et al., 2020a). Our study encompasses two replay regimes: a compute-efficient setup with 2 gradient steps per environment step without resets, and a sample-efficient setup with 16 gradient steps per environment step, including full-parameter resets every 160k steps, as suggested by D'Oro et al. (2022). We provide robust analysis using the RLiable package (Agarwal et al., 2021) and detail the setting in Appendix E.

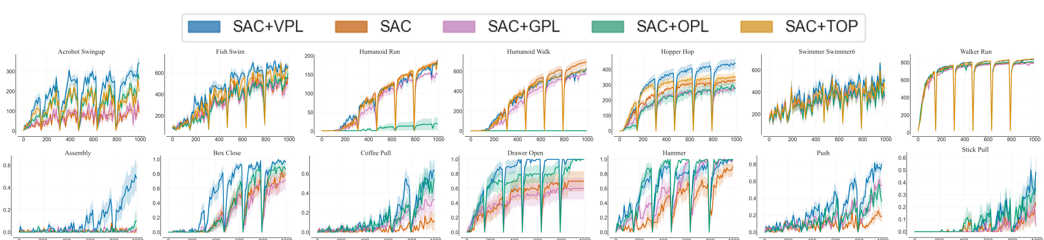


Figure 4: Task-specific performance of high-replay configurations in 14 out of 20 considered tasks. VPL achieves performance improvements, especially in the manipulation tasks. In the case of DMC tasks the y-axis denotes evaluation returns, whereas for MetaWorld tasks it denotes the evaluation success ratio. We detail the experimental setting in Section 5.1. 10 seeds per task.

5.1 Performance and Sample Efficiency

Firstly, we test the performance and sample efficiency of the proposed approach. To this end, we compare SR-SAC (D'Oro et al., 2022) (DMC state of the art) to four algorithms that extend SR-SAC with online pessimism adjustment: GPL (Cetin & Celiktutan, 2023); OPL (Kuznetsov et al., 2021); TOP (Moskovitz et al., 2021); and VPL (the proposed approach). We run the tested algorithms in both replay regimes for 1mln environment steps on 20 medium to hard tasks (10 from DMC and 10 from MetaWorld). We discuss the chosen baselines in Sections 2.2 & C. We discuss hyperparameter selection in Appendix G and the tested tasks in F. We report the results of this experiment in Figures 2, 4 & 5. We find that the proposed approach surpasses baseline algorithms, demonstrating 48% and 27% higher performance than the baseline SR-SAC in low and high replay regimes, respectively. As depicted in Figure 4, VPL exhibits particular effectiveness in MetaWorld manipulation tasks, developing robust policies in environments where other approaches fail, such as the assembly task.

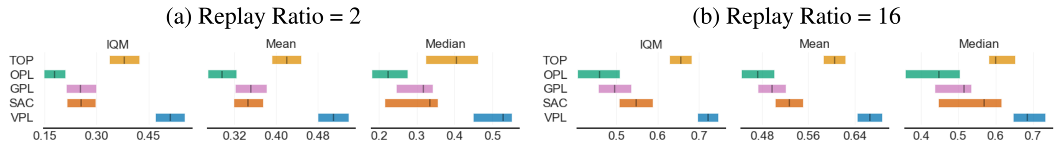


Figure 5: Final performance metrics for the experiment detailed in Section 5.1. VPL outperforms baselines in both replay regimes. The metrics are calculated on 20 tasks listed in Table 3 with 10 seeds per task.

5.2 Validation Buffer Regret

To understand the impact of a validation buffer on online RL training, we analyze three distinct agent setups: *baseline* SR-SAC, which operates without a validation buffer, thus updating actor-critic modules with all experienced transitions; *regret* SR-SAC, which maintains a validation buffer but does not employ validation transitions for pessimism adjustment; and SR-SAC-VPL, which not only maintains a validation buffer but also utilizes validation transitions for pessimism adjustment. This comparative analysis aims to isolate the performance loss attributable to the presence of a validation buffer and the efficiency gains derived from employing VPL for updating pessimism. We evaluate these agents in high-replay regime on 4 tasks (listed in Table 4) over 1mln environment steps, using varying ratios of validation to training samples, specifically at proportions of $\frac{1}{128}$, $\frac{1}{32}$, $\frac{1}{8}$, and $\frac{1}{2}$. The results for this experiment are presented in Figure 6. We observe that the regret associated with maintaining a validation buffer, and thus not utilizing it for actor-critic updates, diminishes over the course of training. Specifically, the *regret* SR-SAC reaches parity with the SR-SAC in performance for all validation proportions except at $\frac{1}{2}$. We note that the rate of regret reduction correlates with the size of the validation proportion, with smaller proportions converging to baseline performance more rapidly. When examining the effectiveness of pessimism adjustment, we observe its most pronounced impact during the early stages of training. This trend aligns with the expectation of reducing critic disagreement over time. Additionally, the extent of performance gain appears to be influenced by the size of the validation buffer, where larger proportions yield greater improvements. This effect is likely due to the increased diversity of environment transitions available for pessimism adjustment in larger buffers. When considering the combined effects on performance, our findings indicate that, except for the $\frac{1}{2}$ proportion, all validation proportions successfully compensate for the performance loss due to validation buffer maintenance. This result is in line with the broader experimental results presented in Figures 2 & 5.

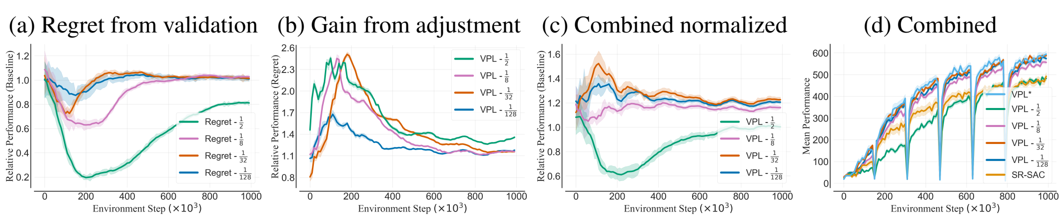


Figure 6: We examine the impact of maintaining a validation buffer on performance distinct from pessimism adjustment across varying proportions of validation samples. Figure 6a demonstrates whether validation agents can match the performance of their validation-free counterparts without utilizing validation samples for pessimism updates, enabling quantification of the regret associated with allocating samples to a validation buffer. Figure 6b quantifies the performance gains attributable to pessimism adjustment by contrasting agents that do not update pessimism against those that do. Figures 6c & 6d illustrate the cumulative effect of validation pessimism adjustment for different validation ratios, benchmarking against the baseline performance of SR-SAC and VPL agents with "free" validation (denoted as VPL*).

5.3 Hyperparameter Sensitivity & Other Experiments

We investigate the sensitivity of VPL to varying pessimism learning rates as compared other pessimism adjustment algorithms. Given the dependency of such learning rate on reward scales and environmental dynamics, determining an optimal rate a priori is challenging, which is a significant restriction for practical applications. To address this, we test the performance of VPL, GPL, and OPL across four environments detailed in Table 4 in the high-replay regime. We evaluate agents after 500k

Table 1: We measure runtimes for 2000 runs of each algorithm and find that the pessimism adjustment methods have trivial wall-clock overhead as compared to SAC/SR-SAC.

METHOD	GPL	OPL	TOP	VPL
RR= 2	0.3%	6.3%	0.3%	3.5%
RR= 16	0.5%	1.1%	0.1%	3.8%

environments steps for learning rates of $[5e-5, 5e-4, 5e-3, 5e-2]$. The results, presented in Figure 7, indicate that VPL exhibits less sensitivity to changes in the pessimism learning rate than the other considered algorithms. Furthermore, we investigate the importance of the two proposed design elements: the use of a validation buffer and the VPL pessimism loss as formulated in Equation 13. To this end, we compare the performance of six agents, each employing different combinations of pessimism loss – either the dual optimization pessimism loss or the VPL pessimism loss – along with varying sources for pessimism updates. These sources include samples from the replay buffer, the validation buffer, and the most recent transitions. The results of this analysis are presented in Figure 9. In our final analysis, we focus on validating the premise of VPL: its effectiveness in reducing approximation error and mitigating overfitting compared to baseline algorithms. Our methodology for quantifying approximation error and overfitting are described in Appendix E. We conducted these measurements across both low and high replay regimes, using a selection of 20 tasks from the DMC and MetaWorld as listed in Table 3. The findings, depicted in Figure 2 and Appendix H, confirm that VPL achieves the lowest levels of critic overfitting and approximation error in both replay scenarios.

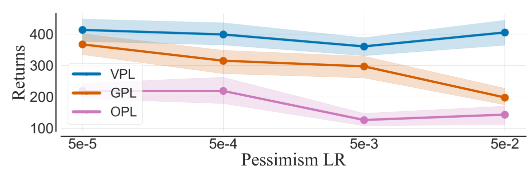


Figure 7: VPL exhibits substantially less sensitivity to the learning rate of the pessimism module. 4 tasks, 10 seeds per task.

6 Limitations

The primary challenge of VPL lies in estimating the lower-bound approximation error necessary for the pessimism adjustment mechanism. This estimation currently relies on a simplistic assumption from inherited from GPL and discussed in Section 4.2. Exploring alternative estimation methods is a promising avenue for future research. Surprisingly, our experiments (see Figure 6) reveal that using a validation buffer does not detrimentally impact agent performance in high-replay scenarios, except in extremely sample-scarce environments (eg. fewer than 250k environment steps).

7 Conclusions

In this paper, we examined the approximation error in critic networks optimized via temporal difference variants. We introduced a fixed-point model for estimating mean and lower bound errors and used this model to analyze the convergence of pessimistic actor-critic algorithms. We proposed the VPL algorithm, which dynamically adjusts pessimism levels to minimize approximation errors of critic supervision in validation samples. We tested VPL against baseline algorithms in various locomotion and manipulation tasks, showing improvements in performance and sample efficiency. We explored the impact of VPL components and their sensitivity to hyperparameter selection. Our results confirm VPLs effectiveness in complex continuous action tasks. We share our code under this link.

Acknowledgements

We thank the Polish high-performance computing infrastructure PLGrid (HPC Center: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2023/016783. Marek Cygan was partially supported by an NCBiR grant POIR.01.01.01-00-0433/20. Mateusz Ostaszewski was funded by the National Science Center Poland under the grant agreement 2020/39/B/ST6/01511.

References

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

Andrychowicz, M., Raichuk, A., Stańczyk, P., Orsini, M., Girgin, S., Marinier, R., Hussenot, L., Geist, M., Pietquin, O., Michalski, M., et al. What matters in on-policy reinforcement learning? a large-scale empirical study. In *ICLR 2021-Ninth International Conference on Learning Representations*, 2021.

Asadi, K., Misra, D., and Littman, M. Lipschitz continuity in model-based reinforcement learning. In *International Conference on Machine Learning*, pp. 264–273. PMLR, 2018.

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

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

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

Barth-Maron, G., Hoffman, M. W., Budden, D., Dabney, W., Horgan, D., Dhruva, T., Muldal, A., Heess, N., and Lillicrap, T. Distributed distributional deterministic policy gradients. In *International Conference on Learning Representations*, 2018.

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

Bergstra, J. and Bengio, Y. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

Bishop, C. M. and Nasrabadi, N. M. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

Buckman, J., Hafner, D., Tucker, G., Brevdo, E., and Lee, H. Sample-efficient reinforcement learning with stochastic ensemble value expansion. *Advances in neural information processing systems*, 31, 2018.

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

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

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

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

De Farias, D. P. and Van Roy, B. On the existence of fixed points for approximate value iteration and temporal-difference learning. *Journal of Optimization theory and Applications*, 105:589–608, 2000.

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

Farahmand, A.-m., Szepesvári, C., and Munos, R. Error propagation for approximate policy and value iteration. *Advances in Neural Information Processing Systems*, 23, 2010.

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2020.

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

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

Ha, D. and Schmidhuber, J. Recurrent world models facilitate policy evolution. *Advances in neural information processing systems*, 31, 2018.

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pp. 1352–1361. PMLR, 2017.

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

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

Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., and Silver, D. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

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

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

Januszewski, P., Olko, M., Królikowski, M., Swiatkowski, J., Andrychowicz, M., Kuciński, Ł., and Miłoś, P. Continuous control with ensemble deep deterministic policy gradients. In *Deep RL Workshop NeurIPS 2021*, 2021.

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

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

Kumar, A., Fu, J., Soh, M., Tucker, G., and Levine, S. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32, 2019.

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

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

Kuznetsov, A., Grishin, A., Tsypin, A., Ashukha, A., Kadurin, A., and Vetrov, D. Automating control of overestimation bias for reinforcement learning. *arXiv preprint arXiv:2110.13523*, 2021.

Lee, H., Cho, H., Kim, H., Gwak, D., Kim, J., Choo, J., Yun, S.-Y., and Yun, C. Plastic: Improving input and label plasticity for sample efficient reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

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

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

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

Mendonca, R., Rybkin, O., Daniilidis, K., Hafner, D., and Pathak, D. Discovering and achieving goals via world models. *Advances in Neural Information Processing Systems*, 34:24379–24391, 2021.

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

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

Munos, R. Error bounds for approximate value iteration. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, pp. 1006. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

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

Munos, R. and Szepesvári, C. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9(5), 2008.

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

Pan, F., He, J., Tu, D., and He, Q. Trust the model when it is confident: Masked model-based actor-critic. *Advances in neural information processing systems*, 33:10537–10546, 2020.

Prechelt, L. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pp. 55–69. Springer, 2002.

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

Rosenfeld, J. S., Rosenfeld, A., Belinkov, Y., and Shavit, N. A constructive prediction of the generalization error across scales. In *International Conference on Learning Representations*, 2019.

Rowland, M., Dadashi, R., Kumar, S., Munos, R., Bellemare, M. G., and Dabney, W. Statistics and samples in distributional reinforcement learning. In *International Conference on Machine Learning*, pp. 5528–5536. PMLR, 2019.

Rowland, M., Munos, R., Azar, M. G., Tang, Y., Ostrovski, G., Harutyunyan, A., Tuyls, K., Bellemare, M. G., and Dabney, W. An analysis of quantile temporal-difference learning. *arXiv preprint arXiv:2301.04462*, 2023.

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

Shang, W., Sohn, K., Almeida, D., and Lee, H. Understanding and improving convolutional neural networks via concatenated rectified linear units. In *international conference on machine learning*, pp. 2217–2225. PMLR, 2016.

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

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

Thrun, S. and Schwartz, A. Issues in using function approximation for reinforcement learning. In *Proceedings of the 1993 connectionist models summer school*, pp. 255–263. Psychology Press, 2014.

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

Van Roy, B. Performance loss bounds for approximate value iteration with state aggregation. *Mathematics of Operations Research*, 31(2):234–244, 2006.

Wang, Z., Wang, J., Zhou, Q., Li, B., and Li, H. Sample-efficient reinforcement learning via conservative model-based actor-critic. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8612–8620, 2022.

Watkins, C. J. and Dayan, P. Q-learning. *Machine learning*, 8:279–292, 1992.

Yao, Y., Xiao, L., An, Z., Zhang, W., and Luo, D. Sample efficient reinforcement learning via model-ensemble exploration and exploitation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4202–4208. IEEE, 2021.

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

Yu, T., Thomas, G., Yu, L., Ermon, S., Zou, J. Y., Levine, S., Finn, C., and Ma, T. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33:14129–14142, 2020b.

Zhang, Z., Pan, Z., and Kochenderfer, M. J. Weighted double q-learning. In *IJCAI*, pp. 3455–3461, 2017.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., Dey, A. K., et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.

A Broader Impact

Although this work is primarily academic, it advances the development of more capable autonomous agents. While our contributions do not directly lead to any negative societal impacts, we encourage the community to remain mindful of potential ethical and societal implications when applying and extending our research.

Appendix Contents

We divide the Appendix into the following sections:

1. Derivations (Appendix B) - we present the derivations associated with statements presented in Section 3. Furthermore, we discuss the further implications of our propositions.
2. VPL pseudocode (Appendix B.3 – we present the pseudocode of one step in the Validation Pessimism Learning algorithm.
3. Related Work (Appendix C) - we discuss the works related to the proposed method. In particular, we discuss pessimistic actor-critic algorithms, approaches for online pessimism adjustment and theoretical work on approximation error in TD learning.
4. Future Work (Appendix D) - we discuss avenues for potential further research related to the proposed method.
5. Experimental Details (Appendix E) - we detail all experiments presented throughout the manuscript.
6. Tested Environments (Appendix F) - we list all tested environments from the DeepMind Control and MetaWorld environments.
7. Additional Experimental Results (Appendix H) - we present additional experimental results.
8. Hyperparameters (Appendix G) - we discuss the procedure for hyperparamenter selection for all algorithms and list all used hyperparameters.
9. Learning Curves (Appendix I) - we present learning curves for the experiments.

B Derivations

In this section, we derive statements presented in Section 3. For simplicity, we consider a fixed policy π_θ and use $V(s)$ and $Q(s,a)$ to represent the value and Q-value under this policy. We define the mean and lower bound approximation errors denoted as U_ϕ^μ and U_ϕ^{lb} respectively:

$$\begin{aligned} U_\phi^\mu(s,a) &\triangleq Q(s,a) - Q_\phi^\mu(s,a) \\ U_\phi^{lb}(s,a) &\triangleq Q(s,a) - Q_\phi^{lb}(s,a) \end{aligned} \tag{14}$$

$Q(s,a)$ denotes the true Q-value, the term $Q_\phi^\mu(s,a)$ represents the mean Q-value estimated by an ensemble of k critics, calculated as $Q^\mu\phi(s,a) = \frac{1}{k}\sum^k Q^i\phi(s,a)$, and $Q^{lb}\phi(s,a)$ is the lower bound Q-value as defined as follows:

$$Q_\phi^{lb}(s,a) = Q_\phi^\mu(s,a) - \beta Q_\phi^\sigma(s,a) \tag{15}$$

Similarly, we define lower bound value:

$$V_\phi^{lb}(s) = \mathop{\mathrm{E}}_{a\sim\pi_\theta}\big(Q_\phi^\mu(s,a) - \beta Q_\phi^\sigma(s,a) - \alpha\log\pi_\theta(a|s)\big) \tag{16}$$

We also introduce the mean and lower bound temporal critic errors, denoted as u_ϕ^μ and u_ϕ^{lb}, respectively:

$$u^{\mu}_{\phi}(s,a,s') \triangleq r_{s,a} + \gamma V^{\mu}_{\phi}(s') - Q^{\mu}_{\phi}(s,a)$$
$$u^{lb}_{\phi}(s,a,s') \triangleq r_{s,a} + \gamma V^{lb}_{\phi}(s') - Q^{\mu}_{\phi}(s,a) \tag{17}$$

These temporal critic errors quantify the deviation between the Q-values $Q^{\mu}_{\phi}(s,a)$ and the mean or lower bound Temporal Difference (TD) targets.

B.1 Approximation Error Operator

Firstly, we note that for the true Q-value the following always holds:

$$Q(s,a) = r_{s,a} + \gamma V(s') = r_{s,a} + \gamma \mathop{\mathrm{E}}_{a'\sim\pi_\theta}\big(Q(s',a') - \alpha \log \pi_\theta(a'|s')\big) \tag{18}$$

Then, using Equations 14, 17 & 18 we write:

$$\begin{aligned}
U^{\mu}_{\phi}(s,a) &= Q(s,a) - Q^{\mu}_{\phi}(s,a) \\
&= r_{s,a} + \gamma V(s') - r_{s,a} - \gamma V^{\mu}_{\phi}(s') + u^{\mu}_{\phi}(s,a,s') \\
&= u^{\mu}_{\phi}(s,a,s') + \gamma\big(V(s') - V^{\mu}_{\phi}(s')\big) \\
&= u^{\mu}_{\phi}(s,a,s') + \gamma \mathop{\mathrm{E}}_{a'\sim\pi_\theta}\big(Q(s',a') - \alpha\log\pi_\theta(a'|s') - Q^{\mu}_{\phi}(s',a') + \alpha\log\pi_\theta(a'|s')\big) \\
&= u^{\mu}_{\phi}(s,a,s') + \gamma \mathop{\mathrm{E}}_{a'\sim\pi_\theta} U^{\mu}_{\phi}(s',a')
\end{aligned} \tag{19}$$

Similarly, we calculate $U^{lb}_{\phi}(s,a)$:

$$\begin{aligned}
U^{lb}_{\phi}(s,a) &= Q(s,a) - Q^{\mu}_{\phi}(s,a) + \beta Q^{\sigma}_{\phi}(s,a) \\
&= u^{\mu}_{\phi}(s,a,s') + \beta Q^{\sigma}_{\phi}(s,a) + \gamma \mathop{\mathrm{E}}_{a'\sim\pi_\theta} U^{\mu}_{\phi}(s',a') \\
&= u^{lb}_{\phi}(s,a,s') + \beta Q^{\sigma}_{\phi}(s,a) + \gamma \mathop{\mathrm{E}}_{a'\sim\pi_\theta} U^{lb}_{\phi}(s',a')
\end{aligned} \tag{20}$$

As such, both $U^{\mu}_{\phi}(s,a)$ and $U^{lb}_{\phi}(s,a)$ can be expressed as a function of combination of $U^{\mu}_{\phi}(s',a')$ or $U^{lb}_{\phi}(s',a')$ and $u^{\mu}_{\phi}(s,a,s')$ or $u^{lb}_{\phi}(s,a,s')$.

B.2 Approximation Error Contraction

We show that both approximation error operators are contractions wrt. infinity norm with similar argument to Bellman values (Puterman, 2014).

$$\begin{aligned}
||\mathcal{U}(f_1) - \mathcal{U}(f_2)||_\infty &= \sup_{s,a} |u^{\mu}_{\phi}(s,a,s') + \gamma \mathop{\mathrm{E}}_{a'\sim\pi_\theta} f_1(s',a') - u^{\mu}_{\phi}(s,a,s') - \gamma \mathop{\mathrm{E}}_{a'\sim\pi_\theta} f_2(s',a')| \\
&= \gamma |\mathop{\mathrm{E}}_{a'\sim\pi_\theta} f_1(s',a') - \mathop{\mathrm{E}}_{a'\sim\pi_\theta} f_2(s',a')| \\
&\leq \gamma \mathop{\mathrm{E}}_{a'\sim\pi_\theta} |f_1(s',a') - f_2(s',a')| \\
&\leq \gamma ||f_1 - f_2||_\infty .
\end{aligned} \tag{21}$$

B.3 VPL pseudocode

Algorithm 1 Validation Pessimism Learning Step

Input: π_θ - actor; Q_ϕ - critic; α - temperature; $\mathcal{D}_T$ - replay buffer; β - pessimism; $\mathcal{D}_V$ - validation buffer
Hyperparameters: B - batch size; v - validation rate
$s', r = \text{ENV.STEP}(a) \quad with \quad a \sim \pi_\theta(a|s)$
$p \sim U(0,1)$
if $p > v$: **then**
  $\mathcal{D}_T.\text{ADD}(s, a, r, s')$
end if
if $p \leq v$: **then**
  $\mathcal{D}_V.\text{ADD}(s, a, r, s')$
end if
for $i = 1$ **to** ReplayRatio **do**
  $s, a, r, s' \sim \mathcal{D}_T.\text{SAMPLE(B)}$
  $s_V, a_V, r_V, s'_V \sim \mathcal{D}_V.\text{SAMPLE(vB)}$
  $\phi \leftarrow \phi - \nabla_\phi \big(Q^\pi_\phi(s,a) - r - \triangle\, \gamma V^{lb}_\phi(s')\big)^2$
  $\theta \leftarrow \theta + \nabla_\theta V^\pi_\theta(s)$
  $\alpha \leftarrow \alpha - \nabla_\alpha \alpha(-\log \pi(a|s) - \mathcal{H}^*)$
  $\beta \leftarrow \beta - \nabla_\beta \big(Q^\pi_\phi(s_V, a_V) - r_V - \gamma V^{lb}_\phi(s'_V)\big)^2$
end for

C Related Work

C.1 Pessimistic Actor-Critic

Recent model-free, off-policy algorithms address the overestimation bias in critic's TD-targets through diverse methods (Thrun & Schwartz, 2014; Hasselt, 2010). These include leveraging multiple function approximators to conservatively estimate expected returns (Fujimoto et al., 2018; Haarnoja et al., 2018; Ciosek et al., 2019; Lee et al., 2021; Andrychowicz et al., 2021). Notably, Clipped Double Q-learning (CDQL) employs a pessimistic approach by calculating the critic's TD-targets as the minimum of two action-value model outputs (Fujimoto et al., 2018). Weighted Double Q-Learning (WDQL) introduces a weighted sum of mean and minimum targets for TD calculations (Zhang et al., 2017). Furthermore, Kuznetsov et al. (2020) suggest using a quantile distributional critic with interquantile statistics for TD target computations. An alternative method proposes reducing approximation errors in TD loss by varying batch sample weights as to counteract the negative interaction of approximation bias and the data-collecting distribution (Kumar et al., 2020). Pessimism was also studied in the context of model-based RL (Ha & Schmidhuber, 2018; Asadi et al., 2018; Janner et al., 2019; Ball et al., 2020; Seyde et al., 2022; Wang et al., 2022). A popular approach is to avoid or reduce the impact of simulated trajectories which the dynamics model deems uncertain (Buckman et al., 2018; Yu et al., 2020b; Yao et al., 2021; Mendonca et al., 2021). In this context, similarly to value, model ensemble disagreement is a very popular approach to uncertainty quantification (Janner et al., 2019; Yu et al., 2020b; Pan et al., 2020; Yao et al., 2021).

C.2 Pessimism Adjustment

Adjusting pessimism levels has become more dynamic with the development of methods that represent the minimum target as a function of the mean and critic ensemble disagreement (Kuznetsov et al., 2021; Moskovitz et al., 2021; Cetin & Celiktutan, 2023). GPL, for instance, modifies pessimism using a dual optimization objective, calculating the loss on replay samples (Cetin & Celiktutan, 2023). OPL uses an online approach to adjust pessimism levels by comparing critic outputs with on-policy return estimators (Kuznetsov et al., 2021). TOP uses an auxiliary bandit to select optimal pessimism levels for maximizing online returns (Moskovitz et al., 2021). These approaches are detailed in Table 2.

Table 2: Considered algorithms differ by the pessimism domain, strategy for critic error estimation, as well as the pessimism update rule.

	β DOMAIN	CRITIC ERROR ESTIMATED VIA	PESSIMISM UPDATE RULE
TOP	$0, 1$	NA	Auxiliary bandit maximizing episodic returns
GPL	$[0, \infty]$	Critic output on the replay transitions	Dual optimization update
OPL	$[0, \infty]$	Bootstrapped λ-returns on recent transitions	Dual optimization update
VPL	$[0, \infty]$	Critic output on the validation transitions	Minimization of approximation error

C.3 Approximation Error in RL

The regret caused by errors in critic approximation has been explored in approximate value iteration algorithms (De Farias & Van Roy, 2000; Van Roy, 2006; Munos, 2005, 2007; Munos & Szepesvári, 2008; Farahmand et al., 2010). When a policy is greedy with respect to the critic estimates, value approximation errors can greatly influence the policy and the resulting returns. Therefore, there's been significant work to understand how these approximation errors affect performance (Munos, 2005, 2007; Munos & Szepesvári, 2008; Farahmand et al., 2010). These ideas have also been revisited in the area of deep reinforcement learning (Kumar et al., 2019, 2020). In particular, Kumar et al. (2020) examines the detailed patterns in non-pessimistic value approximation errors. Those results remain relevant for off-policy actor-critic algorithms such as SAC, as it can be described as an approximate policy iteration algorithm (Haarnoja et al., 2018).

D Future Work

While our implementation is based on the vanilla SR-SAC algorithm, recent studies have demonstrated that simple regularization methods applied to the critic can significantly enhance performance (Hiraoka et al., 2021; Li et al., 2022; Ball et al., 2023). Consequently, integrating VPL with network regularization appears to be a promising approach. Specifically, layer normalization and spectral normalization have been effective in continuous action off-policy agents (Ba et al., 2016; Gogianu et al., 2021). Similarly, it has been observed that deep RL agents experience a reduced ability to learn over time, a phenomenon known as 'plasticity loss'. Addressing this diminishing capacity has been shown to be empirically beneficial (Janner et al., 2019; Nikishin et al., 2022; D'Oro et al., 2022; Lyle et al., 2023). Although our approach involves full-parameter resets in the high replay regime, employing multiple techniques to address plasticity loss has proven advantageous Lee et al. (2023). Therefore, combining VPL with strategies like CReLU Shang et al. (2016) or Sharpness Aware Minimization (SAM) Foret et al. (2020) could potentially lead to further performance improvements. Given that VPL employs a more controlled use of the critic ensemble compared to standard SAC/TD3 methods, increasing the critic ensemble size in VPL may create synergies, potentially surpassing the benefits seen in conventional ensemble AC approaches (Chen et al., 2020; Lee et al., 2021; Januszewski et al., 2021; Ball et al., 2023). Additionally, the integration of a distributional critic setup into the pessimism adjustment framework (Moskovitz et al., 2021), which has been shown to enhance RL learning (Bellemare et al., 2017; Rowland et al., 2019, 2023), suggests that incorporating distributional critics into VPL could yield notable performance gains.

E Experimental Details

E.1 Performance and Sample Efficiency

We run the tested algorithms for 1mln environment steps on 20 DMC/MetaWorld tasks listed in Table 3 using hyperparameters described in Section G. All algorithms are evaluated via greedy policies every 10k environment steps. We calculate the final performance presented in Figure 5 by averaging over last 10 policy evaluations (ie. the last 100k environment steps). The results for this setup are presented in Figures 2, 4 and 5, as well as in the final Appendix section.

E.2 Approximation Error and Overfitting

Throughout the manuscripts we present estimates of critic approximation error and overfitting. Here, we discuss our methodology for calculating both.

E.2.1 Approximation Error

We calculate the critic approximation via:

$$U^{\mu}_{\phi}(s,a) = -(Q(s,a) - Q^{\mu}_{\phi}(s,a)) \tag{22}$$

Where we add the negative sign such that the positive approximation error represents overestimation, and negative approximation error represents underestimation. Above, $Q^{\mu}_{\phi}(s,a)$ represents the output of the critic. Estimating the reference Q-value $Q(s,a)$ in our context demands solving two problems. Firstly, as all algorithms use SAC as their backbone, the Q-value is soft, ie. it represents the sum of returns and entropies. This contrasts with regular DDPG-style critic which approximates the returns alone. Secondly, both MetaWorld and DMC are infinite-horizon MDPs. As such, obtaining an unbiased Monte-Carlo rollout value is non-trivial. To this end, we calculate the reference Q-value via:

$$Q(s,a) = \frac{1}{1-\gamma}\big(\hat{R}_{s,a} - \alpha \log \hat{\pi}(a|s)\big) \tag{23}$$

Where $\hat{R}_{s,a}$ denotes the average reward gathered when performing action a at state s and following the policy afterwards, and $\log \hat{\pi}$ denotes the average policy log-probability when following the policy from a given state-action. The term $\frac{1}{1-\gamma}$ stems from the sum of a geometric series, reflecting the infinite-horizon that the critic models. We estimate $\hat{R}$ with using a Monte-Carlo rollout, and use the entropy target to calculate $\log \hat{\pi}$. For each approximation error measurement, we average the error over 5 different starting states.

E.2.2 Overfitting

We calculate the critic overfitting (denoted as $\mathcal{O}(\phi)$) with the following equation:

$$\mathcal{O}(\phi) = \frac{\underset{\mathcal{D}_V}{\mathrm{E}}\, u^{lb}_{\phi}(s_V, a_V, s^{'}_V)}{\underset{\mathcal{D}_T}{\mathrm{E}}\, u^{lb}_{\phi}(s, a, s')} \tag{24}$$

Where $\mathcal{D}_T$ and $\mathcal{D}_V$ denote the training and validation replay buffers respectively. Furthermore, $s_V, a_V, s^{'}_V$ denote the transitions sampled from the validation buffer, and s, a, s' denote the training replay buffer transitions. As such, we calculate our overfitting metric by comparing the temporal difference for unseen validation transitions and the training transitions which were the critic is trained on. For each algorithm, we gather such validation samples during the evaluation rollouts. As such, the validation buffer gathers 5,000 new transitions every 10,000 training transitions. We estimate the expectation on batches of 256 transitions sampled from each buffer. Such definition of overfitting is easily interpretable - when $\mathcal{O}(\phi)$ is close to 1 then both validation and training TD errors are comparable and therefore there is little to none overfitting. When $\mathcal{O}(\phi)$ is greater than 1 then it means that the validation TD errors are relatively larger than the ones in the training, indicating overfitting.

E.3 Validation Buffer Regret

Our study examines three agent configurations: (1) baseline SR-SAC, which updates actor-critic modules with all transitions and lacks a validation buffer; (2) regret SR-SAC, featuring a validation buffer but not using it for pessimism adjustment; and (3) SR-SAC-VPL, which includes a validation buffer and employs validation transitions for pessimism adjustment. The analysis focuses on identifying performance differences caused by the validation buffer and the benefits of using VPL for pessimism updates. We tested these agents across four tasks (see Table 4) for 1 million environment steps, exploring various validation to training sample ratios, namely $\frac{1}{128}$, $\frac{1}{32}$, $\frac{1}{8}$, and $\frac{1}{2}$. The experiments are performed in high replay setting with full parameter resets every 160k environment steps (D'Oro et al., 2022). Experimental results are detailed in Figure 6.

E.4 Learning Rate Sensitivity

We run GPL, OPL and VPL for 500k environment steps in high replay setup on 4 DMC tasks listed in Table 4. We test each algorithms performance on four pessimism learning rate values: $[5e-5, 5e-4, 5e-3, 5e-2]$. We present the results in Figure 7.

E.5 Design Choices

Finally, we evaluate the impact of each of VPL contributions, namely the VPL update rule and performing updates on the validation buffer. We evaluate six agents, each utilizing different forms of pessimism loss - either dual optimization or VPL pessimism loss - in combination with various sources for pessimism updates. These sources encompass samples from the replay buffer (ie. GPL), the validation buffer (ie. VPL), and the most recent online transitions (ie. OPL). We run the agents for 1mln environment steps on 5 dmc tasks shown in Figure 9.

F Tested Environments

Tables below list tasks from DeepMind Control and MetaWorld considered in our experiments.

Table 3: 20 DMC and MetaWorld tasks used for the main evaluation.

DeepMind Control	MetaWorld
acrobot-swingup	assembly
fish-swim	box-close
hopper-hop	button-press
hopper-stand	coffee-pull
humanoid-run	coffee-push
humanoid-stand	drawer-open
humanoid-walk	hammer
quadruped-run	push
swimmer-swimmer6	stick-pull
walker-run	sweep

Table 4: 4 DMC tasks used in additional experiments.

DeepMind Control
acrobot-swingup
hopper-hop
humanoid-walk
quadruped-run

G Hyperparameters

All considered algorithms use SR-SAC (D'Oro et al., 2022) as their backbonce, we align the common hyperparameters with those recommended for SR-SAC (D'Oro et al., 2022). This includes using the same network architectures, a two-critic ensemble (Fujimoto et al., 2018; Haarnoja et al., 2018; Ciosek et al., 2019; Moskovitz et al., 2021; Cetin & Celiktutan, 2023) and ADAM optimizer (Kingma & Ba, 2014). We choose VPL, GPL, and OPL pessimism adjustment learning rate by performing search over the same domain for all algorithms which we present in Figure 7. We choose the validation ratio for VPL in experiments presented in Figure 6. We choose TOP bandit setting following the best performing configurations presented Moskovitz et al. (2021). We use consistent hyperparamenters between all environments and both replay regimes. All experiments are run without any action repeat wrappers (ie. we use an action repeat of 1). The hyperparameters are summarized in Table 5.

H Additional Experimental Results

We evaluate the impact of each of VPL contributions, namely the VPL update rule and performing updates on the validation buffer. We evaluate six agents, each utilizing different forms of pessimism loss - either dual optimization (denoted as "Dual") or VPL pessimism loss (denoted as "VPL") - in

Table 5: Hyperparameter values used in the experiments.

Hyperparameter	Notation	Value
Joint		
Network Size	NA	$(256, 256)$
Optimizer	NA	Adam
Learning Rate	NA	$3e-4$
Batch Size	B	256
Discount	γ	0.99
Initial Temperature	α_0	1.0
Initial Steps	NA	10000
Target Entropy	$\mathcal{H}^*$	$\lvert\mathcal{A}\rvert/2$
Polyak Weight	τ	0.005
TOP		
Pessimism Values	β	$0, 1$
Bandit Learning Rate	NA	0.1
GPL		
Pessimism Learning Rate	NA	$5e-5$
Initial Pessimism	β	1.0
OPL		
Pessimism Learning Rate	NA	$5e-5$
On-policy Trajectory Length	NA	8
TD-λ	NA	0.95
Initial Pessimism	β	1.0
VPL		
Pessimism Learning Rate	NA	$5e-5$
Validation Proportion	V	$1/32$
Initial Pessimism	β	1.0

combination with one of three sources of data for pessimism updates. These sources encompass samples from the validation buffer (denoted as "Validation"), the replay buffer (denoted as "Replay" and used originally in GPL (Cetin & Celiktutan, 2023)), and the most recent online transitions (denoted as "Online" and used originally in OPL (Kuznetsov et al., 2021)). The performance, pessimism, approximation error and overfitting of these agents is presented in Figures 8 and 9.

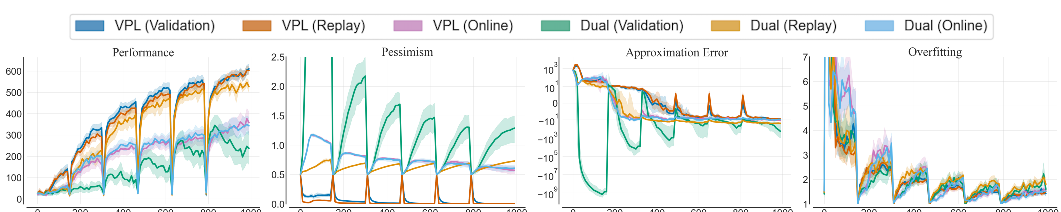


Figure 8: Ablation on design of the pessimism module. 4 tasks, 10 seeds per task.

We find that the VPL pessimism adjustment loss accounts for the majority of the performance improvements of VPL algorithm. As such, we find that updating the pessimism on validation buffer accounts for minor performance improvements over updates performed on the replay buffer. We still deem this result as significant, since the validation agent skips $\frac{1}{16}$ of training transitions. Furthermore, we find that performing VPL pessimism updated on recent transitions leads to pessimism increases and subpar performance. Interestingly, we find that performing dual optimization pessimism updates on the validation buffer leads to the worse performing agent. In terms of pessimism patterns, we observe that performing pessimism updates on the online transitions leads to more pessimistic agents than then other data sources for pessimism updates. Finally, we find that the VPL pessimism updates performed on the replay data leads to sharper decreases of pessimism that the regular validation VPL.

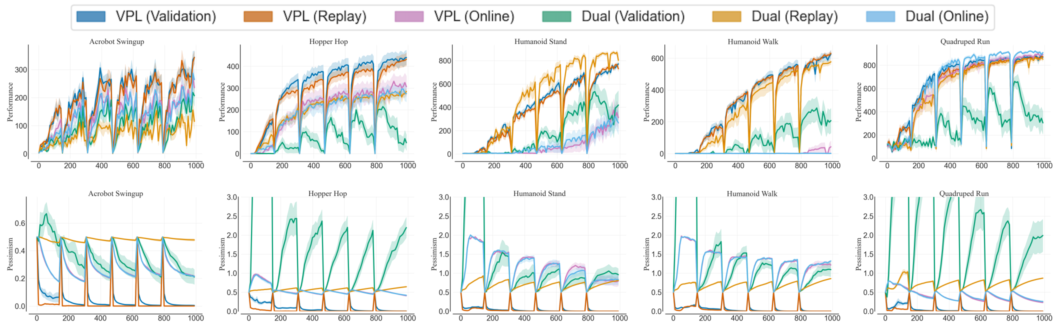


Figure 9: Ablation on design of the pessimism module. 10 seeds per task.

Finally, we investigate whether the critic disagreement diminishes to zero in a popular training regime of 1mln environment steps. As noted in the paper, the convergence of pessimistic actor-critic depends on the critic disagreement being equal to zero. We find that the critic disagreement indeed does not completely diminish. Interestingly, we observe that the same environments yield the most disagreement in both low and high replay regimes.

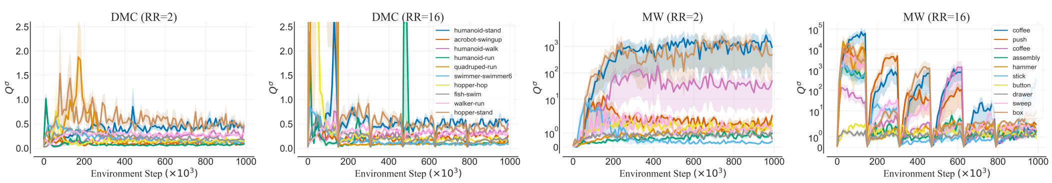
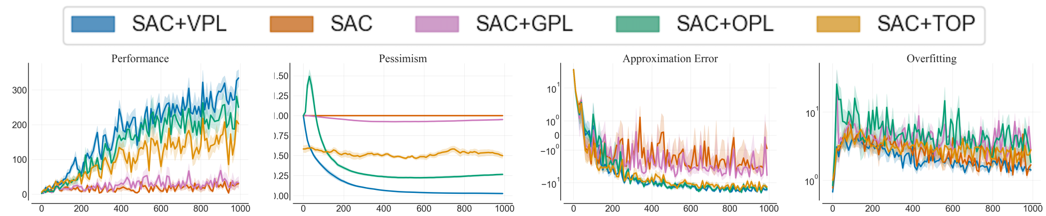


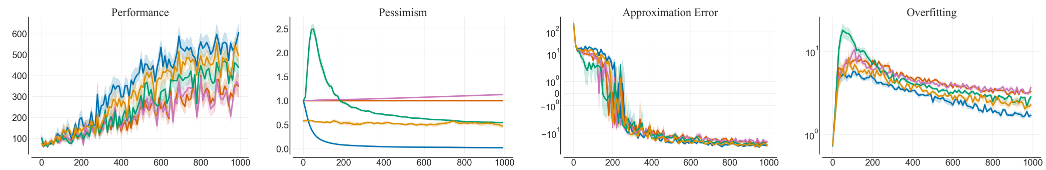
Figure 10: The critic disagreement of SAC algorithm does not completely diminish in the considered training regime of 1 million environment steps, making adjustments to the pessimism a viable strategy. 10 seeds per task.

I Learning Curves

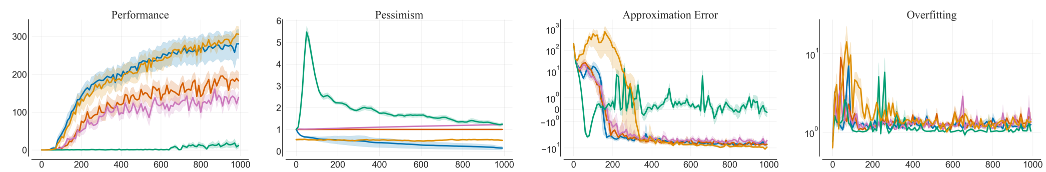
Finally, we present the detailed training curves for performance, pessimism, approximation error and overfitting. The low replay regime results are presented in Figures 11, 12, 13 and 14. the high replay regime results are presented in Figures 15, 16, 17 and 18.



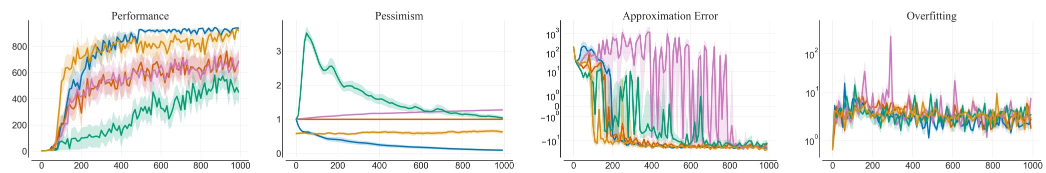
(a) Acrobot Swingup



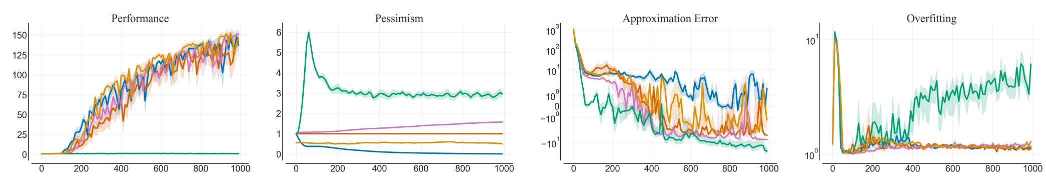
(b) Fish Swim



(c) Hopper Hop



(d) Hopper Stand



(e) Humanoid Run

Figure 11: Low replay regime results for each considered task (1/4). 10 seeds per task, mean and 3 standard deviations.

NeurIPS Paper Checklist

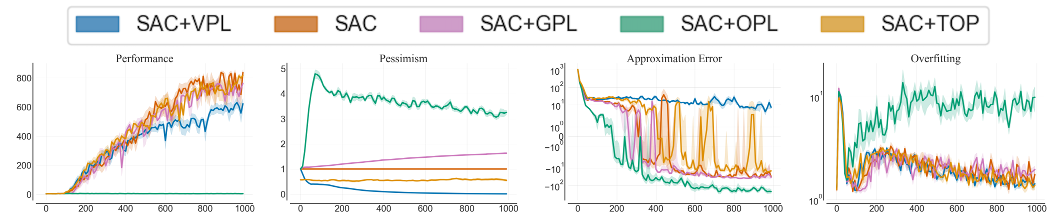
1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

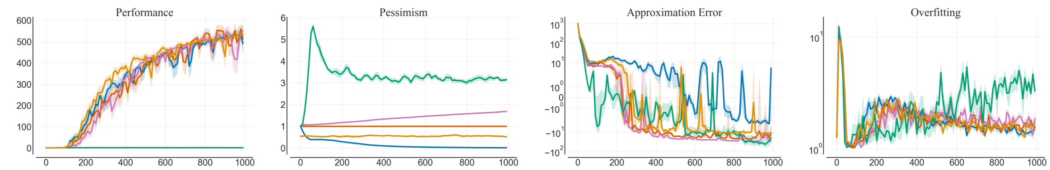
   Answer: [Yes]

   Justification: Abstract and introduction clearly state the claims, contributions, assumptions and limitations made in the text.

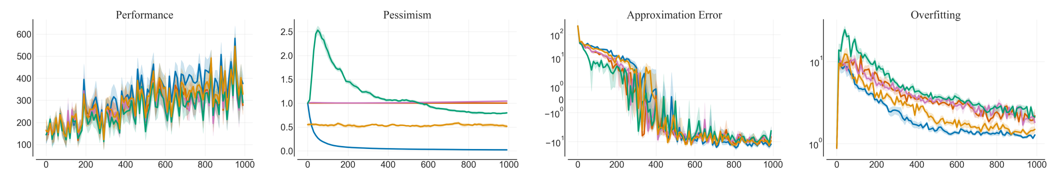
   Guidelines:



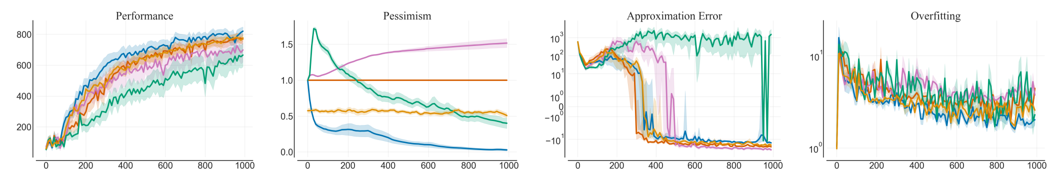
(a) Humanoid Stand



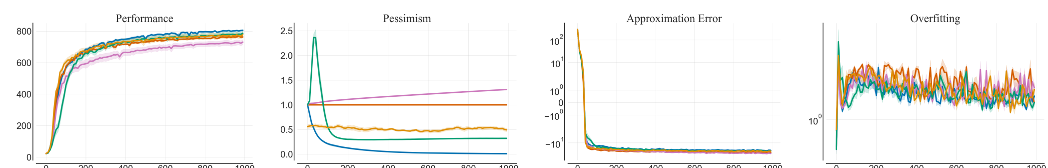
(b) Humanoid Walk



(c) Swimmer Swimmer6



(d) Quadruped Run



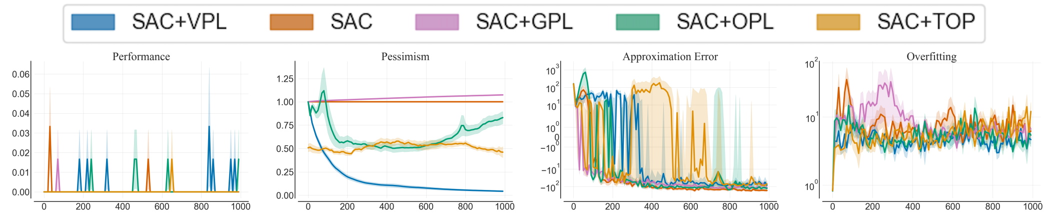
(e) Walker Run

Figure 12: Low replay regime results for each considered task (2/4). 10 seeds per task, mean and 3 standard deviations.

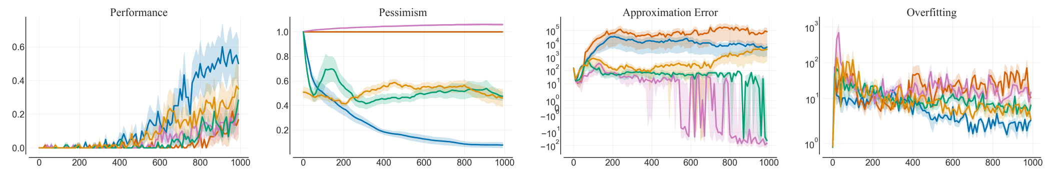
- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

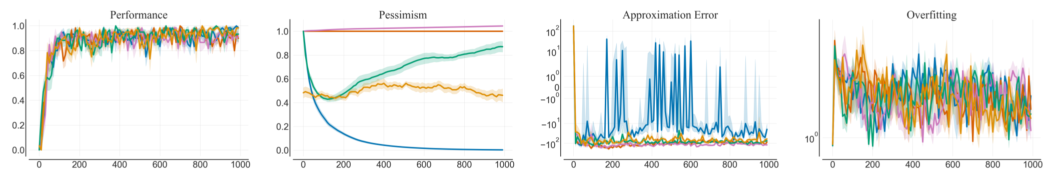
   Question: Does the paper discuss the limitations of the work performed by the authors?



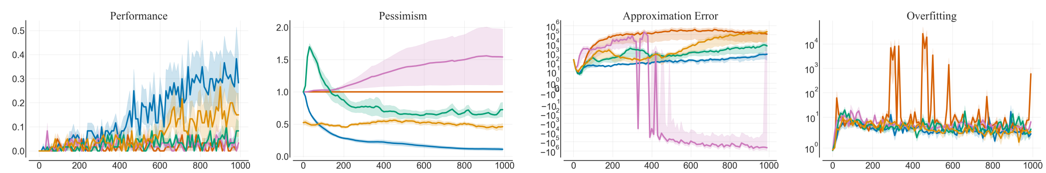
(a) Assembly



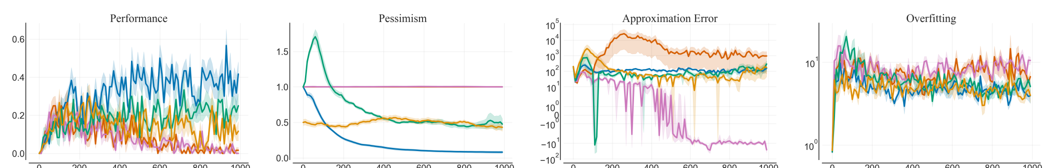
(b) Box Close



(c) Button Press



(d) Coffee Pull



(e) Coffee Push

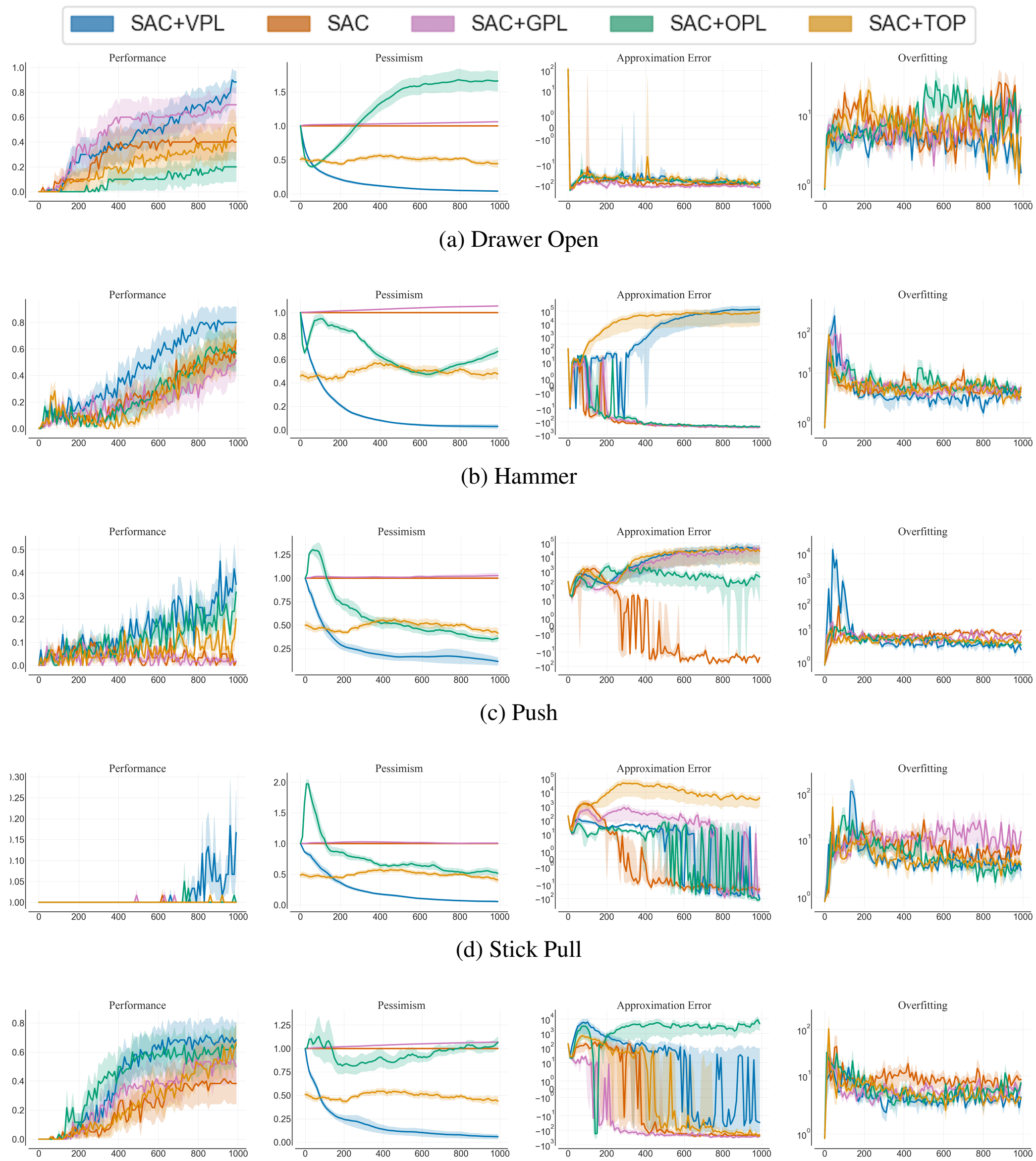
Figure 13: Low replay regime results for each considered task (3/4). 10 seeds per task, mean and 3 standard deviations.

Answer: [Yes]

Justification: In the main text we have Section 6.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors



(a) Drawer Open

(b) Hammer

(c) Push

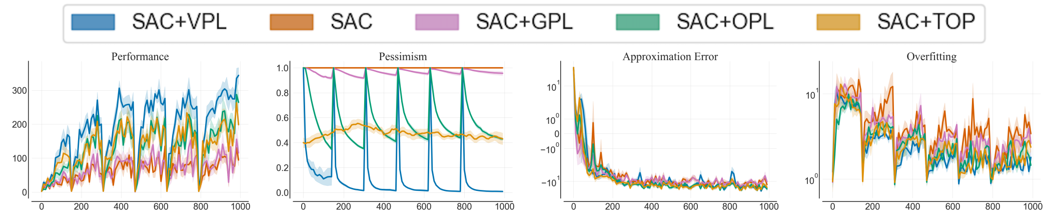
(d) Stick Pull

(e) Sweep

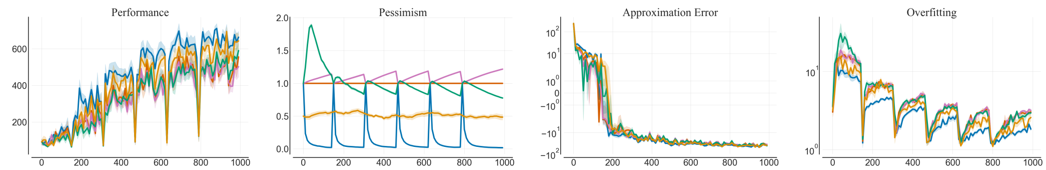
Figure 14: Low replay regime results for each considered task (4/4). 10 seeds per task, mean and 3 standard deviations.

should reflect on how these assumptions might be violated in practice and what the implications would be.

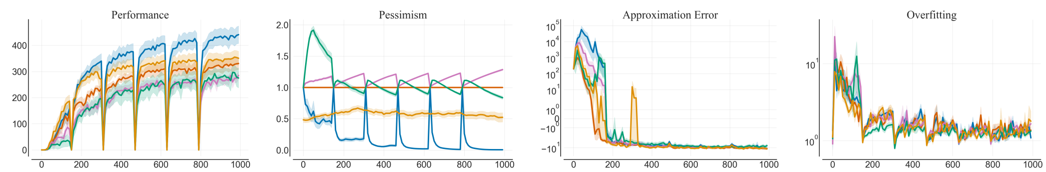
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.



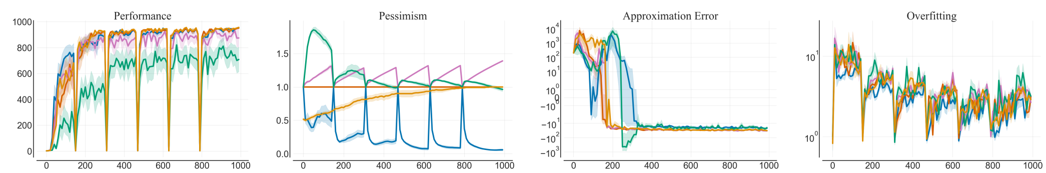
(a) Acrobot Swingup



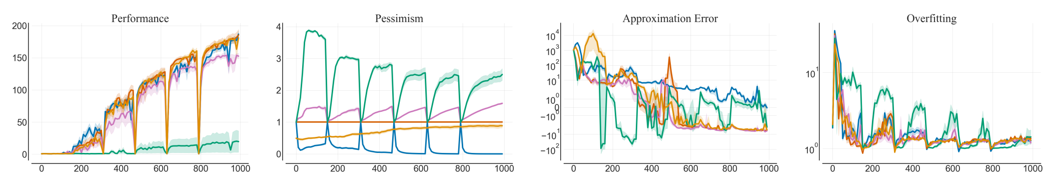
(b) Fish Swim



(c) Hopper Hop



(d) Hopper Stand

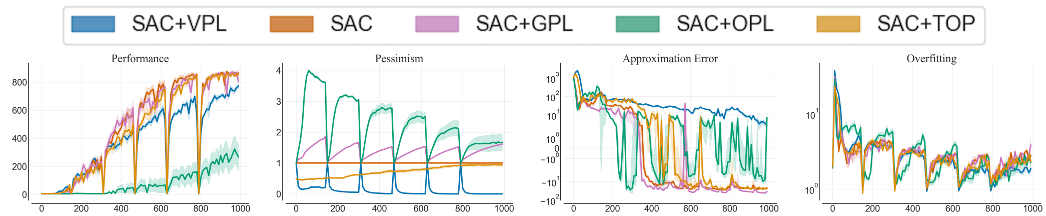


(e) Humanoid Run

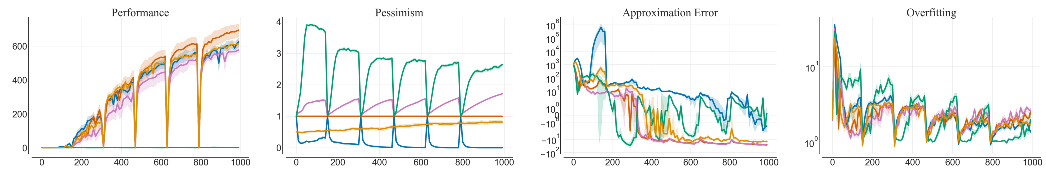
Figure 15: High replay regime results for each considered task (1/4). 10 seeds per task, mean and 3 standard deviations.

- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

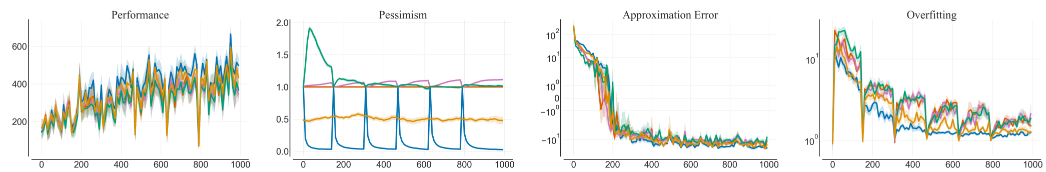
3. **Theory Assumptions and Proofs**



(a) Humanoid Stand



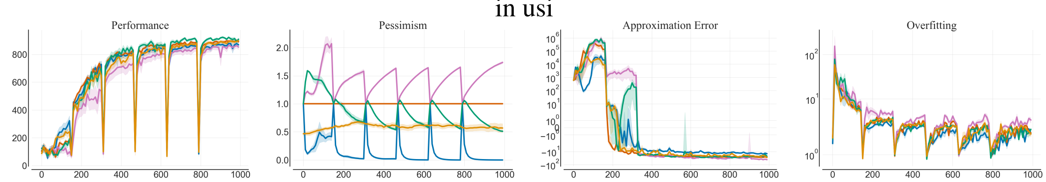
(b) Humanoid Walk



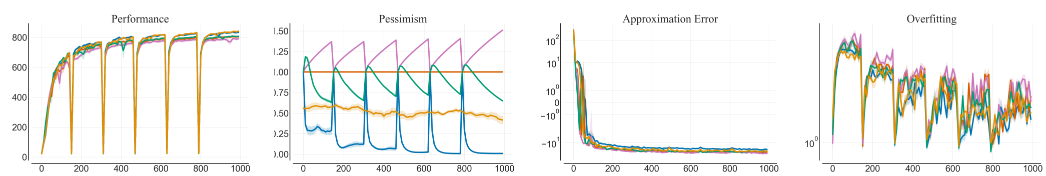
Issues

(c) Swimmer Swimmer6

in usi



(d) Quadruped Run



(e) Walker Run

Figure 16: High replay regime results for each considered task (2/4). 10 seeds per task, mean and 3 standard deviations.

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All assumptions are clearly stated, and derivations of formulas are provided in Appendix B.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.

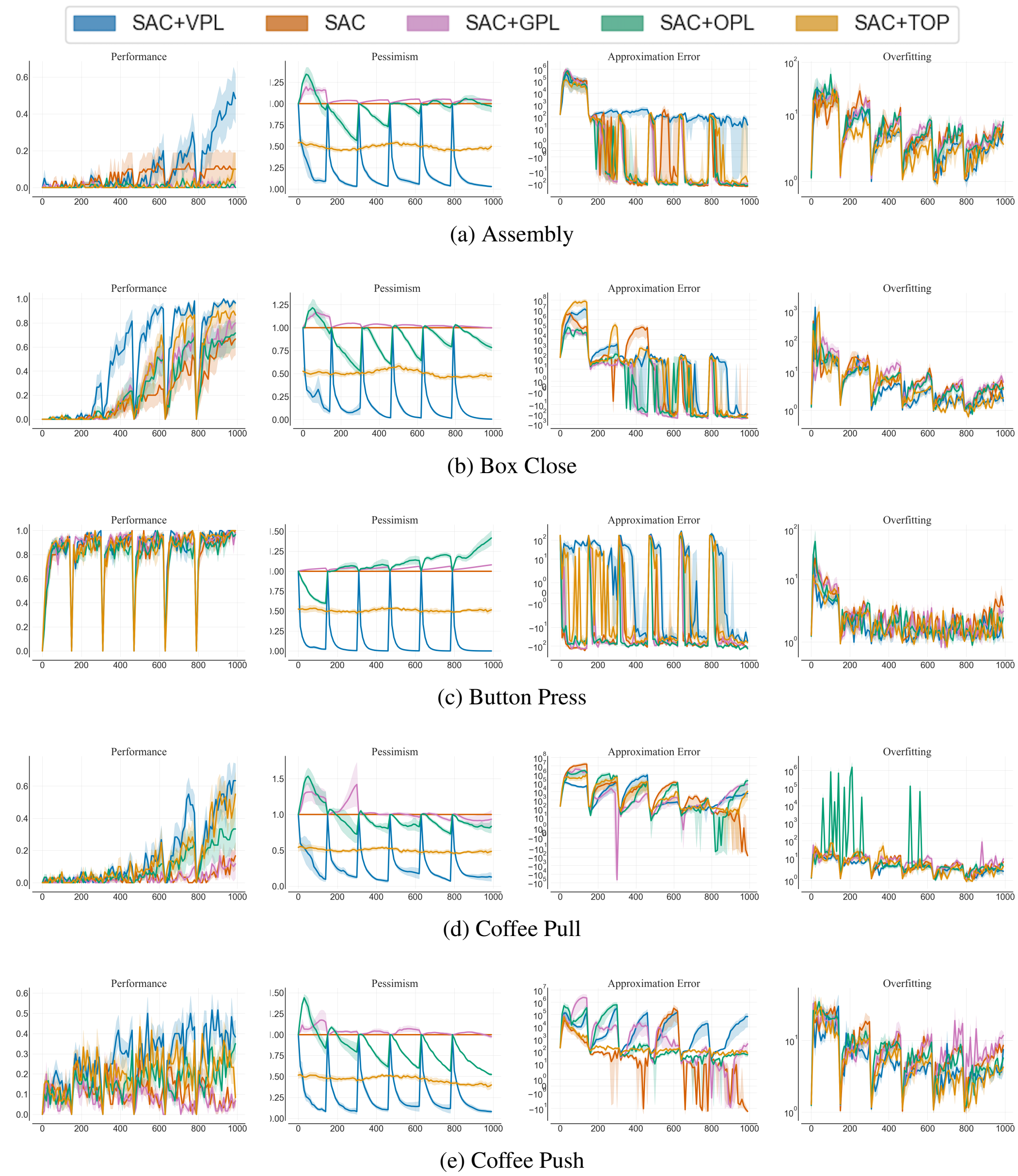
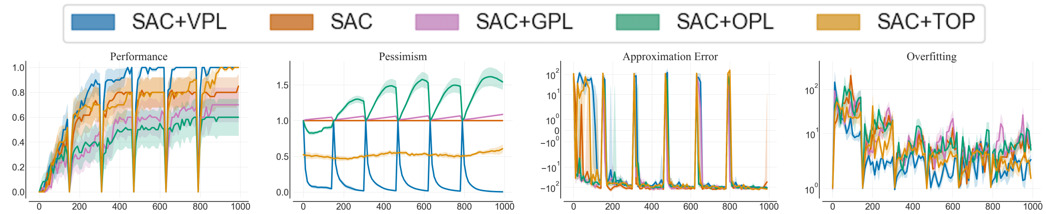


Figure 17: High replay regime results for each considered task (3/4). 10 seeds per task, mean and 3 standard deviations.

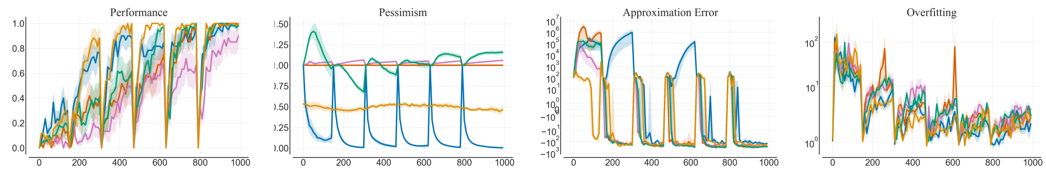
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

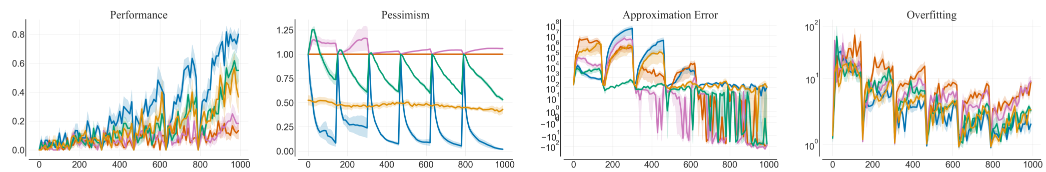
   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?



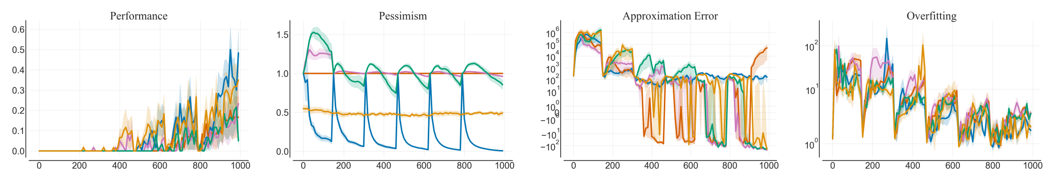
(a) Drawer Open



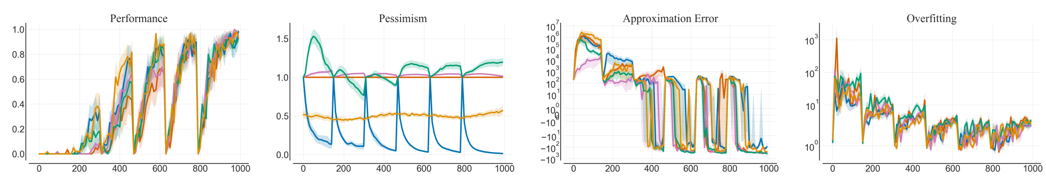
(b) Hammer



(c) Push



(d) Stick Pull



(e) Sweep

Figure 18: High replay regime results for each considered task (4/4). 10 seeds per task, mean and 3 standard deviations.

Answer: [Yes]

Justification: The paper includes a description of the experiments (Sections E, G, F), as well as code used to generate the results (`https://anonymous.4open.science/r/Valdation-Pessimism-Learning-6D4F`).

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.

- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We share the code used to generate the results (`https://anonymous.4open.science/r/Valdation-Pessimism-Learning-6D4F`).

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Experimental details are in the Appendix sectionsGF. Moreover, all details are also in the available code `https://anonymous.4open.science/r/Valdation-Pessimism-Learning-6D4F`

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: All experiments are performed for multiple tasks, with 10 random seeds per task. We calculate 95% bootstrapped confidence intervals using RLiable package (Agarwal et al., 2021).

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We report wallclock time required to run all algorithms, on a uniform compute setup described in Section 5.2.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We include a Broader Impact Section (Section A).

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The research presented in this paper does not pose such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Baseline algorithms were implemented based on the details provided in the articles. Repositories used for benchmarking are distributed under MIT (MetaWorld) and Apache 2.0 (DeepMind Control Suite) licenses.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We discuss the reproducibility of our experiments in Sections E, G, F. Code is available under the following link `https://anonymous.4open.science/r/Valdation-Pessimism-Learning-6D4F`

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: We do not have human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: We do not have human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.