LEARNING PESSIMISM FOR ROBUST AND EFFICIENT OFF-POLICY REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Popular off-policy deep reinforcement learning algorithms compensate for overestimation bias during temporal-difference learning by utilizing pessimistic estimates of the expected target returns. In this work, we propose a novel *learnable* penalty to enact such pessimism, based on a new way to quantify the critic's epistemic uncertainty. Furthermore, we propose to learn the penalty alongside the critic with *dual TD-learning*, a strategy to estimate and minimize the bias magnitude in the target returns. Our method enables us to accurately counteract overestimation bias *throughout* training without incurring the downsides of overly pessimistic targets. Empirically, by integrating our method and other orthogonal improvements with popular off-policy algorithms, we achieve *state-of-the-art* results in continuous control tasks from both proprioceptive and pixel observations.

1 INTRODUCTION

Sample efficiency and generality are two directions in which reinforcement learning (RL) algorithms are still lacking, yet, they are crucial for tackling complex real-world problems (Mahmood et al., 2018). Consequently, so far, many RL milestones have been achieved through simulating conspicuous amounts of experience and tuning for effective task-specific parameters (Mnih et al., 2013; Silver et al., 2017). Recent off-policy model-free (Lee et al., 2021; Chen et al., 2021) and model-based algorithms (Chua et al., 2018; Janner et al., 2019), pushed forward the state-of-the-art sample-efficiency on several benchmark tasks (Brockman et al., 2016). We attribute such improvements to two main linked advances: more expressive models to capture uncertainty and better strategies to counteract detrimental biases from the learning process. These advances yielded the stabilization to adopt more aggressive optimization procedures, with particular benefits in lower data regimes.

Modern off-policy algorithms learn behavior by optimizing the expected performance as predicted by a trained parametric deep model of the environment. Within this process, overestimation bias naturally arises from the maximization performed over the model's performance predictions, and consequently, also over the model's possible errors. In the context of model-free RL, such model is trained to predict the agent's future returns via temporal difference (TD-) learning and is referred to as the *critic*. A common strategy to counteract overestimation is to parameterize the critic with multiple, independently-initialized networks and optimize the agent's behavior over the minimum value of the relative outputs (Fujimoto et al., 2018). Empirically, this strategy consistently yields pessimistic target performance measures, avoiding overestimation bias propagating through the TD-learning target bootstraps. However, this approach directly links the critic's parameterization to bias counteraction, making improvements in each of these components hard to pursue independently.

Based on these observations, we propose a more general formulation for counteracting overestimation bias independently. In particular, we compute the critic's target performance predictions by replacing the ubiquitous minimization procedure with an explicit penalty that is agnostic to the critic's parameterization. Our proposed penalty is the output of a linear model of the epistemic uncertainty, computed as the expected Wasserstein distance between the return distributions predicted by the critic. Based on this formulation, we derive ***Generalized Pessimism Learning (GPL)***, a new strategy that *learns* an accurate penalization *throughout* the RL process. Within this strategy, we propose optimizing the penalty's weight with *dual TD-learning*, a new procedure minimizing the estimated bias in the critic's performance predictions with dual gradient descent. *GPL* is the first effective method able to *freely* learn an unbiased performance objective throughout training.

Furthermore, we also extend *GPL* by introducing a new *pessimism annealing* procedure, motivated by the principle of *optimism in the face of uncertainty* (Brafman & Tennenholtz, 2002). This procedure leads the agent to adopt a risk-seeking behavior policy by utilizing a purposely biased estimate of the performance in the initial training stages. Hence, it trades-off expected immediate performance for directed exploration, incentivizing the visitation of states with high epistemic uncertainty from which the critic would gain more information.

We incorporate *GPL* with modern implementations of the *Soft Actor-Critic (SAC)* (Haarnoja et al., 2018a;b) and *Data-regularized Q (DrQ)* (Yarats et al., 2021b;a) algorithms, yielding *GPL-SAC* and *GPL-DrQ*, respectively. On the Mujoco environments from the OpenAI Gym suite (Todorov et al., 2012; Brockman et al., 2016), *GPL-SAC* outperforms prior, more expensive, model-based (Janner et al., 2019) and model-free (Chen et al., 2021) state-of-the-art algorithms. For instance, in the Humanoid environment *GPL-SAC* is able to recover a score of 5000 in less than 100K experience steps, more than nine times faster than regular *SAC*. Additionally, on pixel-based environments from the DeepMind Control Suite (Tassa et al., 2018), *GPL-DrQ* provides significant performance improvements from the recent state-of-the-art *DrQv2* algorithm. These results highlight the effectiveness and applicability of *GPL*, in spite of introducing only negligible computational overheads. We release our implementations to facilitate future comparisons and extensions.

In summary, we make several contributions towards improving off-policy reinforcement learning:

- We propose a novel penalty to counteract overestimation bias, disentangling the critic's parameterization from the enforced pessimism.
- We propose the first optimization method to estimate and precisely counteract overestimation bias throughout training with dual gradient descent.
- We propose a pessimism annealing strategy that exploits epistemic uncertainty to actively seek informative states in the early training stages.
- Integrating our method with *SAC* and *DrQ*, we achieve new state-of-the-art performance results with trivial overheads on both proprioceptive and pixel observations tasks.

2 Related work

Modern model-free off-policy algorithms utilize different strategies to counteract overestimation bias arising in the critic's TD-targets (Thrun & Schwartz, 1993; Pendrith et al., 1997; Mannor et al., 2007). Many approaches combine the predictions of multiple function approximators to estimate the expected returns, for instance, by independently selecting the bootstrap action (Hasselt, 2010). In discrete control, such technique appears to mitigate the bias of the seminal *DQN* algorithm (Mnih et al., 2013), consistently improving performance (Van Hasselt et al., 2016; Hessel et al., 2018). In continuous control, similar strategies successfully stabilize algorithms based on the policy gradient theorem (Silver et al., 2014; Lillicrap et al., 2015). Most notably, Fujimoto et al. (2018) proposed to compute the critic's TD-targets by taking the minimum over the outputs of two different action-value models. This particular minimization strategy has become ubiquitous, being employed in many popular follow-up algorithms (Haarnoja et al., 2018b; Yarats et al., 2021b). To better trade-off optimism and pessimism, Zhang et al. (2017) proposed using a weighted combination of the original and minimized targets. Instead, Kuznetsov et al. (2020) proposed to parameterize a distributional critic and drop a fixed fraction of the predicted quantiles to compute the targets. Moreover, as in our approach, several works also considered explicit penalties based on heuristic measures of epistemic uncertainty (Lee et al., 2013; Ciosek et al., 2019). Recently, Kumar et al. (2020) proposed to complement these strategies by further reducing bias propagation through actively weighing the TD-loss of different experience samples. Aleatoric uncertainty is also an additional source of bias in TD-learning, due to the practical inability of considering multiple transition samples in stochastic environments (Baird, 1995). This phenomenon is known as the *double-sample* issue, but has been rarely addressed in prior off-policy literature (Dai et al., 2018).

Within model-based RL (Atkeson & Santamaria, 1997), recent works have achieved remarkable sample efficiency by learning large ensembles of dynamic models for better predictions (Chua et al., 2018; Wang & Ba, 2019; Janner et al., 2019). In the model-free framework, prior works used large critic ensembles for more diverse scopes. Anschel et al. (2017) proposed to build an ensemble using several past versions of the value network to reduce the magnitude of the TD-target's bias.

Moreover, Lan et al. (2020) introduced a sampling procedure for the critic's ensemble predictions to regulate underestimation in the TD-targets. Their work was later extended to the continuous setting by Chen et al. (2021), which showed that large ensembles combined with a high update-to-data ratio enable to outperform the sample efficiency of contemporary model-based methods. Ensembling has also been used to achieve better exploration following the principle of optimism in the face of uncertainty (Brafman & Tennenholtz, 2002) in both discrete (Osband et al., 2016; Chen et al., 2017) and continuous settings (Ciosek et al., 2019). Lee et al. (2021) further showed the effectiveness of combining several of these strategies in a unified framework.

In the same spirit as this work, multiple prior methods attempted to learn components and parameters of underlying RL algorithms. Several works have approached this problem by utilizing expensive meta-learning strategies to obtain new learning objectives based on the multi-task performance from low-computation environments (Bechtle et al., 2021; Oh et al., 2020; Xu et al., 2020; Co-Reyes et al., 2021). More related to our work, Moskovitz et al. (2021) recently proposed to use an adaptive binary controller that switches on or off a bias correction penalty for the TD-targets. In particular, they treat the controller optimization task as a multi-armed bandit problem performed throughout different training iterations to maximize immediate performance improvements. Instead, *GPL* makes effective use of dual gradient descent to minimize bias directly, similarly to how Haarnoja et al. (2018a;b) learns the exploration temperature parameter in the *Soft Actor-Critic (SAC)* algorithm.

3 PRELIMINARIES

In RL, we aim to autonomously recover optimal agent behavior for performing a particular task. Formally, we describe this problem setting as a Markov Decision Process (MDP), defined as the tuple $(S, A, P, p_0, r, \gamma)$. At each time-step of interaction, the agent observes some state in the state space, $s \in S$, and performs some action in the action space, $a \in A$. The transition dynamics function $P : S \times A \times S \to \mathbb{R}$ and the initial state distribution $p_0 : S \to \mathbb{R}$ describe the evolution of the environment as a consequence of the agent's behavior. The reward function $r : S \times A \to \mathbb{R}$ quantifies the effectiveness of each performed action, while the discount factor $\gamma \in [0, 1)$ represents the agent's preference for earlier rewards. A policy $\pi : S \times A \to \mathbb{R}$ maps each state to a probability distribution over actions and represents the agent's behavior. An episode of interactions between the agent and the environment yields some trajectory τ containing the transitions experienced, $\tau = (s_0, a_0, r_0, s_1, a_1, r_1, ...)$. The RL objective is then to find an optimal policy π^* that maximizes the expected sum of discounted future rewards:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{p_\pi(\tau)} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right], \tag{1}$$

where $p_\pi(\tau)$ represents the distribution of trajectories stemming from the agent's interaction with the environment. Off-policy RL algorithms commonly utilize some *critic* model to evaluate the effectiveness of the agent's behavior. A straightforward choice for the critic is to represent the policy's action-value function $Q^\pi : S \times A \to \mathbb{R}$. This function quantifies the expected sum of discounted future rewards after executing some particular action from a given state:

$$Q^\pi(s, a) = \mathbb{E}_{p_\pi(\tau | s_0 = s, a_0 = a)} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right]. \tag{2}$$

Most RL algorithms consider learning parameterized models for both the policy, π_θ, and the corresponding action-value function, Q^π_ϕ. In particular, after storing experience transitions (s, a, s', r) in a replay data buffer D, we learn Q^π_ϕ by iteratively minimizing a squared TD-loss of the form:

$$\begin{aligned} J_Q(\phi) &= \mathbb{E}_{(s,a,s',r) \sim D} \left[(Q^\pi_\phi(s, a) - y)^2 \right], \\ y &= r + \gamma \mathbb{E}_{a \sim \pi(s')} \left[\hat{Q}^\pi_{\phi'}(s', a) \right]. \end{aligned} \tag{3}$$

Here, the TD-targets y are obtained by computing a 1-step bootstrap with a *target action-value estimator* $\hat{Q}^\pi_{\phi'}$. Usually, $\hat{Q}^\pi_{\phi'}$ is a regularized function of action-value predictions from a target critic model using delayed parameters ϕ'. Following the policy gradient theorem (Sutton et al., 2000; Silver et al., 2014), we can then improve our policy by maximizing the expected returns as predicted by the critic, e.g., by minimizing the negated action-value estimates:

$$J_\pi(\theta) = -\mathbb{E}_{s \sim D, a \sim \pi_\theta(s)} \left[\hat{Q}^\pi_\phi(s, a) \right]. \tag{4}$$

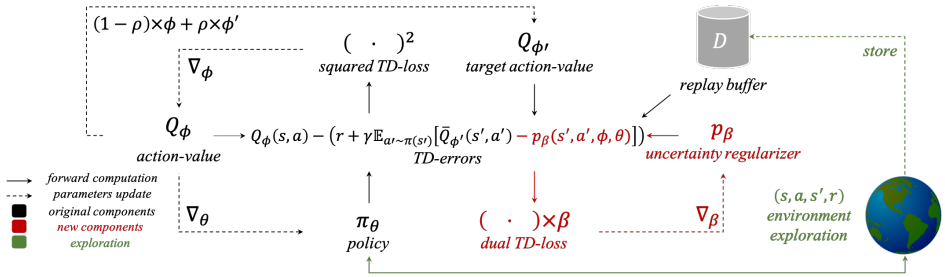

Figure 1: Schematic overview of the training and exploration processes involved in the proposed *GPL* framework. The TD-errors play a central role and are utilized both for updating the critic and for estimating the current bias to update the parameterized uncertainty regularizer.

4 Addressing Overestimation Bias

4.1 Bias in Q-learning

In off-policy RL, several works have identified an accumulation of overestimation bias in the action-value estimates as a consequence of TD-learning (Thrun & Schwartz, 1993; Pendrith et al., 1997; Mannor et al., 2007). Formally, we quantify the target action-value bias $B(s,a,s')$ as the difference between the actual and estimated TD-targets for a given transition:

$$\begin{aligned} B(s,a,s') &= r + \gamma \mathbb{E}_{a' \sim \pi(s')}\left[\hat{Q}^\pi_{\phi'}(s',a')\right] - \left(r + \gamma \mathbb{E}_{a' \sim \pi(s')}\left[Q^\pi(s',a')\right]\right) \\ &= \gamma \mathbb{E}_{a' \sim \pi(s')}\left[\hat{Q}^\pi_{\phi'}(s',a') - Q^\pi(s',a')\right]. \end{aligned} \tag{5}$$

As discussed by Fujimoto et al. (2018), positive bias arises when the target action-values are obtained directly from the outputs of a parameterized action-value function, i.e., $\hat{Q}^\pi_{\phi'} = Q^\pi_{\phi'}$. The reason for this phenomenon is that the policy is trained to locally maximize the action-value estimates from Eqn. 4. Hence, its actions will exploit potential model errors to obtain higher scores, implying that $\mathbb{E}_{s,a\sim\pi(s)}[Q^\pi_{\phi'}(s,a)] > \mathbb{E}_{s,a\sim\pi(s)}[Q^\pi(s,a)]$. Instabilities then arise as the errors can quickly propagate through the bootstrap operation, inherently causing the phenomenon of *positive bias accumulation*. To counteract this phenomenon, Fujimoto et al. (2018) proposed *clipped double Q-learning*. This technique consists in learning two separate action-value functions and computing the target action-values by taking the minimum over their outputs:

$$\hat{Q}^\pi_{\phi'_{min}}(s,a) = \min\left(Q^\pi_{\phi'_1}(s,a), Q^\pi_{\phi'_2}(s,a)\right). \tag{6}$$

The role of the minimization is to consistently produce overly pessimistic estimates of the target action-values, preventing positive bias accumulation. This approach is an empirically effective strategy for different benchmark tasks and has become standard practice.

4.2 The Uncertainty Regularizer

In this work, we take a more general approach for computing the target action-values. Particularly, we use a parameterized function, the *uncertainty regularizer* $p_\beta(s,a,\phi,\theta)$, for trying to approximate the bias in the critic's action-value predictions for on-policy actions. Thus, we specify an action-value estimator that penalizes the action-value estimates via the uncertainty regularizer:

$$\begin{aligned} \hat{Q}^\pi_{\phi'}(s,a) =& Q^\pi_\phi(s,a) - p_\beta(s,a,\phi,\theta), \\ \textit{where} \quad p_\beta(s,a,\phi,\theta) \approx& Q^\pi_\phi(s,a) - Q^\pi(s,a), \quad \textit{for} \quad a \sim \pi_\theta(s). \end{aligned} \tag{7}$$

A consequence of this formulation is that as long as p_β is unbiased for on-policy actions, so will the action-value estimator. Hence, the expected target action-value bias will be zero, preventing the positive bias accumulation phenomenon without requiring overly pessimistic action-value estimates. Based on these observations, in the next section, we specify a new method that learns an unbiased uncertainty regularizer and continuously adapts it to reflect changes in the critic and policy.

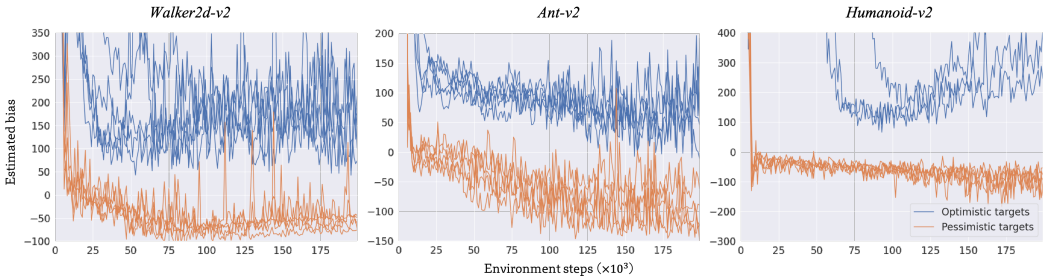

Figure 2: Recorded estimated bias for ten runs of two simple extensions of the *SAC* algorithm.

5 GENERALIZED PESSIMISM LEARNING

Generalized pessimism learning (GPL) entails learning a particular parameterized uncertainty regularizer p_β, defined in Eqn. 8. Our method makes p_β adapt to changes in θ and ϕ throughout the RL process to keep the target action-values unbiased. Hence, *GPL* allows preventing positive bias accumulation without overly pessimistic targets. With any fixed penalty, we argue that it would be infeasible to maintain the expected target action-value bias close to zero due to the number of affecting parameters and stochastic factors in different RL experiments.

5.1 UNCERTAINTY REGULARIZER PARAMETERIZATION

We strive for a parameterization of the uncertainty regularizer that ensures low bias and variance estimation of the target action-values. Similar to prior works (Ciosek et al., 2019; Moskovitz et al., 2021), *GPL* uses a linear model of some measure of the epistemic uncertainty in the critic. Epistemic uncertainty represents the uncertainty from the model's learned parameters towards its possible predictions. Hence, assuming enough model expressivity, the areas of the state and action spaces where the critic's epistemic uncertainty is elevated are the areas in which the agent did not yet observe enough data to reliably predict its returns and, for this reason, the magnitude of the critic's error is expectedly higher. Consequently, if a policy yields behavior with high epistemic uncertainty in the critic, it is likely exploiting positive errors and overestimating its expected returns. As we use the policy to compute the TD-targets, the higher the uncertainty, the higher the expected positive bias. Several prior works further discuss this relationship (Fujimoto et al., 2018; Ciosek et al., 2019).

We propose measuring epistemic uncertainty with the expected Wasserstein distance between the critic's predicted return distributions Z^π, as defined by Bellemare et al. (2017). In the usual case where we parameterize the critic with multiple action-value functions, we interpret each action-value estimate as a Dirac delta function approximation of the return distribution, $Z^\pi_\phi(s,a) = \delta_{Q^\pi_\phi(s,a)}$. Our uncertainty regularizer then consists of linearly scaling this measure via a learnable parameter β:

$$p_\beta(s,a,\phi,\theta) = \beta \times \mathbb{E}_{a\sim\pi_\theta(s),\phi_1,\phi_2}\left[W(Z^\pi_{\phi_1}(s,a), Z^\pi_{\phi_2}(s,a))\right]. \tag{8}$$

We estimate the expectation in Eqn. 8 by learning a critic ensemble of $N \geq 2$ independent models with parameters $\{\phi_i\}_{i=1}^N$, and averaging the distances between the corresponding predicted return distributions. Notably, the Wasserstein distance has easy-to-compute closed forms for many popular distributions. For Dirac delta functions, it is equivalent to the distance between the corresponding locations, hence, $W(\delta_{Q^\pi_{\phi_1}(s,a)}, \delta_{Q^\pi_{\phi_2}(s,a)}) = |Q^\pi_{\phi_1}(s,a) - Q^\pi_{\phi_2}(s,a)|$.

Our quantification of epistemic uncertainty is an interpretable measure for any distributional critic. Moreover, for some fixed β, increasing the number of critics decreases the estimation variance but leaves the expected magnitude of the uncertainty regularizer unchanged. This is because the sample mean of the Wasserstein distances is always an unbiased estimate of Eqn. 8 for $N \geq 2$. Under reasonable assumptions, it is proportional to the standard deviation of the distribution of action-value predictions. We can also restate clipped double Q-learning using our uncertainty regularizer with $N = 2$ and $\beta = 0.5$, allowing us to replicate its penalization effects for $N > 2$ by simply fixing β. In contrast, Ciosek et al. (2019) proposed the sample standard deviation of the action-value predictions to quantify epistemic uncertainty. However, the sample standard deviation does not have a clear generalization to arbitrary distributional critics and its expected magnitude is dependent on the number of models. We provide formal derivations for these claims in Appendix A.

Algorithm 1 *GPL-SAC*

procedure TRAININGLOOP(N, *UTD*, $\rho, \gamma, \hat{H}$)
  Initialize π_θ, $\{Q^\pi_{\phi_i}\}_{i=1}^N$, $\{Q^\pi_{\phi'_i}\}_{i=1}^N$, $\beta \leftarrow 0.5, \alpha \leftarrow 0.0, D \leftarrow \emptyset$
  loop
    Observe s, execute $a \sim \pi_\theta(s)$, collect s', r
    Store $D \leftarrow D \cup (s, a, s', r)$
    for $j = 1, 2, ...,$ *UTD* **do**
      Sample minibatch $\{(s, a, s', r)\} \in D$, $a' \sim \pi(s')$
      Compute $\hat{Q}^\pi_{\phi'}(s', a') \leftarrow$ REGULARIZEDVALUE$(s', a', \phi', \theta, \beta)$
      Compute the TD-targets $y \leftarrow r + \gamma \left(\hat{Q}^\pi_{\phi'}(s', a') - \alpha \log \pi(a'|s') \right)$
      for $i = 1, 2, ..., N$ **do**
        Compute the TD-errors for the i^{th} critic $e_i \leftarrow Q^\pi_{\phi_i} - y$
        Update ϕ_i to minimize $J(\phi) = e_i^2$ ▷ Learning the critic
        Update $\phi'_i \leftarrow \rho\phi'_i + (1 - \rho)\phi_i$ ▷ Updating the target critic
    Update β to minimize $J(\beta) = \beta \times \sum_{i=1}^N e_i$ ▷ Learning the bias (*dual TD-learning*)
    Sample $a \sim \pi(s)$, compute $\hat{Q}^\pi_\phi(s, a) \leftarrow$ REGULARIZEDVALUE$(s, a, \phi, \theta, \beta)$
    Update θ to minimize $J(\theta) = -\hat{Q}^\pi_{\phi_i}(s, a) + \alpha \log \pi(a|s)$ ▷ Learning the policy
    Update α to minimize $J(\alpha) = \alpha \times (-\log \pi(a|s) - \hat{H})$ ▷ Learning the entropy bonus
procedure REGULARIZEDVALUE$(s, a, \phi, \theta, \beta)$
  Compute $p_\beta(s, a, \phi, \theta) \leftarrow \frac{\beta}{N^2 - N} \sum_{i=1}^N \sum_{j \neq i}^N |Q^\pi_{\phi_i}(s, a) - Q^\pi_{\phi_j}(s, a)|$ ▷ Sample mean of Eqn. 8
  Compute $\bar{Q}^\pi_\phi(s, a) \leftarrow \frac{1}{N} \sum_{i=1}^N Q^\pi_{\phi_i}(s, a)$
  return $\bar{Q}^\pi_\phi(s, a) - p_\beta(s, a, \phi, \theta)$

5.2 DUAL TD-LEARNING

The expected bias present in the action-value targets is highly dependent on several stochastic factors stemming from aleatoric uncertainty and the learning process. We empirically show this by running multiple experiments with simple extensions of the *SAC* algorithm in different Gym environments and periodically recording estimates of the action-value bias by comparing the actual and estimated discounted returns (as described in Appendix B). As shown in Figure 2, the bias in the predicted action-values notably varies across environments, agents, training stages, and even across different random seeds. These results reinforce our thesis that there is no fixed penalty able to account for the many sources of stochasticity in the RL process, even for a single task. Hence, they show the necessity of learning p_β alongside the policy and critic to accurately counteract bias.

GPL optimizes β as a dual variable, enforcing the expected target action-value bias to be zero:

$$\arg\min_\beta -\beta \times \mathbb{E}_{s,a,s'} \left[B(s, a, s') \right]. \tag{9}$$

For arbitrary actions, the action-value estimates are not directly affected by the positive bias induced by the policy gradient optimization. Consequently, we can make the reasonable assumption of the critic itself providing an initially unbiased estimate of the expected returns, i.e., $\mathbb{E}_{s,a}[Q^\pi_\phi(s, a)] \approx \mathbb{E}_{s,a}[Q^\pi(s, a)]$. Thus, to approximate $B(s, a, s')$, we propose to use the differences between the current TD-targets and action-value predictions for off-policy actions:

$$B(s, a, s') \approx \hat{B}(s, a, s') = r + \gamma \mathbb{E}_{a' \sim \pi(s')} \left[\hat{Q}^\pi_{\phi'}(s', a') \right] - Q^\pi_\phi(s, a). \tag{10}$$

In practice, *GPL* alternates the optimizations of β for the current bias, and both ϕ and θ, with the corresponding updated RL objectives. This procedure is similar to the automatic exploration temperature optimization proposed by Haarnoja et al. (2018b), approximating dual gradient descent (Boyd & Vandenberghe, 2004). We can estimate the current bias according to Eqn. 10 at minimal extra cost by negating the errors from the TD-loss. Thus, we name this procedure *dual TD-learning*.

Unfortunately, the unbiasedness assumption of Eqn. 10 does not necessarily hold when using deep networks and approximate stochastic optimization. In particular, given initially biased targets, some of the bias might propagate to the critic model, influencing the approximation in Eqn. 10. Nonetheless, *GPL*'s performance and the optimization dynamics of dual TD-learning appear to be empirically robust to moderate levels of initial target bias. We show this in Appendix D.1 by analyzing the

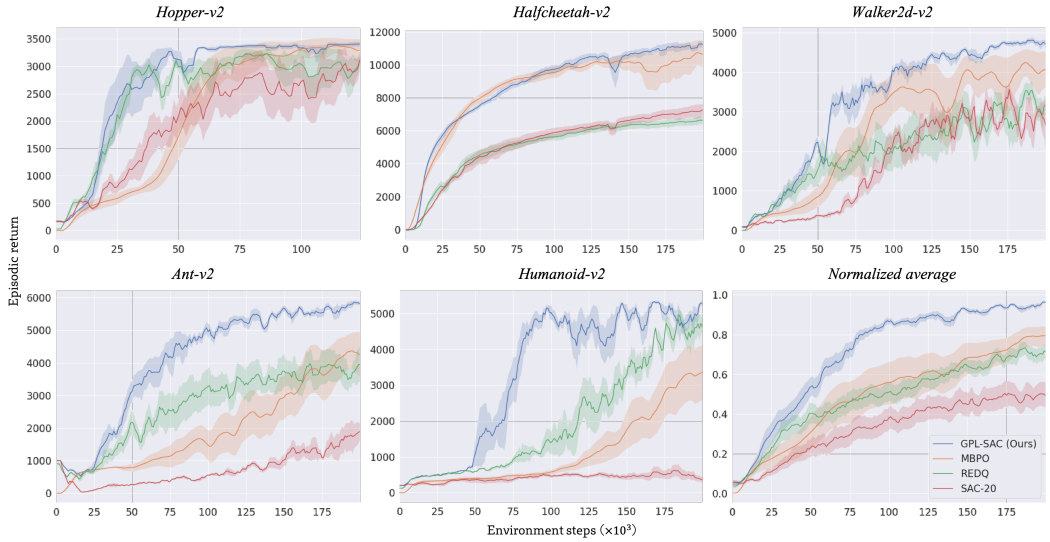

Figure 3: Performance curves for the considered five complex environments from the OpenAI Gym suite. We report both mean and standard deviation of the episodic returns over five random seeds.

behavior and performance of *GPL* with different initial values of β and, thus, different bias in the initial targets. An intuition for this robustness is that the relative difference between the off-policy and on-policy action-value predictions should always push β to counteract bias stemming from model errors in the policy gradient action maximization (Fujimoto et al., 2018). We further validate dual TD-learning in Appendix D.2 by comparing and discussing alternative optimization strategies. In particular, we evaluate optimizing β by minimizing the squared norm of the bias and by using the bandit-based optimization from *Tactical Optimism and Pessimism* (Moskovitz et al., 2021).

5.3 Pessimism annealing for directed exploration

As described in Section 3, the policy learns to maximize the unbiased action-values predicted by the online estimator $\hat{Q}^{\pi}_{\phi}$. Motivated by the principle of *optimism in the face of uncertainty* (Brafman & Tennenholtz, 2002), we also consider an alternative *optimistic* policy gradient objective:

$$J^{opt}_{\pi}(\theta) = -\mathbb{E}_{s,a\sim\pi(s)}[\hat{Q}^{\pi_{opt}}_{\phi}(s,a)], \quad \textit{where} \quad \hat{Q}^{\pi_{opt}}_{\phi}(s,a) = Q^{\pi}_{\phi}(s,a) - p_{\beta_{opt}}(s,\phi,\theta). \tag{11}$$

This objective utilizes an *optimistic shifted uncertainty regularizer*, $p_{\beta_{opt}}$, calculated with parameter $\beta_{opt} = \beta - \lambda_{opt}$, for some decaying *optimistic shift value*, $\lambda_{opt} \geq 0$. This new objective trades off the traditional exploitative behavior of the policy with directed exploration. In particular, as λ_{opt} is large, π will be incentivized to perform actions for which the outcome has high epistemic uncertainty. Therefore, the agent will experience transitions that are increasingly informative for the critic but expectedly sub-optimal. Hence, we name the process of decaying λ_{opt} *pessimism annealing*, striving for improved exploration early on without biasing the policy's final objective.

6 Experiments

To evaluate the effectiveness of *GPL*, we integrate it with two popular off-policy RL algorithms. *GPL* itself introduces trivial computational and memory costs as it optimizes a single additional weight, re-utilizing the errors in the TD-loss to estimate the bias. Moreover, we implement the critic's model ensemble as a single neural network, using linear non-fully-connected layers evenly splitting the nodes and dropping the weight connections between the splits. Practically, when evaluated under the same hardware, this results in our algorithm running more than two times faster than the implementation from Chen et al. (2021) while having a similar algorithmic complexity.

We show that *GPL* significantly improves the performance and robustness of off-policy RL, concretely surpassing prior algorithms and setting new state-of-the-art results. In our evaluation, we repeat each experiment with five random seeds and record both mean and standard deviation over

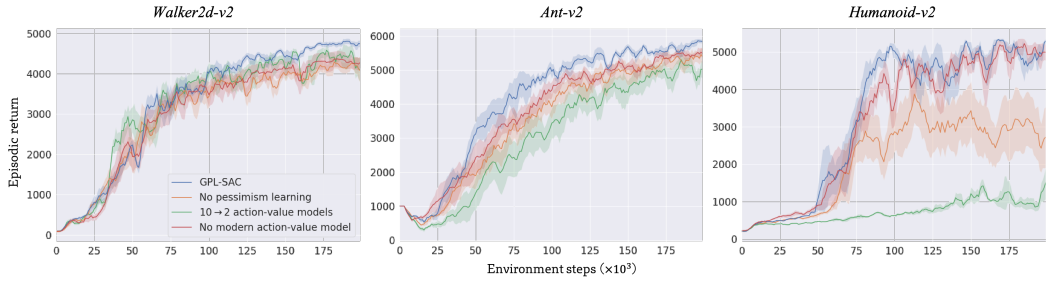

Figure 4: Ablation study of the main components differentiating *GPL-SAC* from *SAC-20*.

the episodic returns. We report additional details of our experimental settings and utilized hyperparameters in Appendix B. Furthermore, we provide comprehensive extended results analyzing different parameters, alternative implementations, and training times in Appendix D.

6.1 Continuous control from proprioceptive observations

GPL-SAC. First, we integrate *GPL* with *Soft Actor-Critic (SAC)* (Haarnoja et al., 2018a;b), a popular model-free off-policy algorithms that uses a weighted entropy term in its objective to incentivize exploration. Specifically, we substitute *SAC*'s clipped double Q-learning with our uncertainty regularizer, initialized with $\beta = 0.5$. We also adopt additional practices inspired by recent related works (Chen et al., 2021; Bjorck et al., 2021). In particular, we parameterize ten independent action-value functions as 'modern' residual neural networks with spectral normalization (Miyato et al., 2018) and increase the critic's update-to-data (UTD) ratio to twenty. We denote the resulting algorithm *GPL-SAC* and provide a summary of this integration in Algorithm 1, highlighting the novelties.

Baselines. We compare *GPL-SAC* with prior state-of-the-art model-free and model-based algorithms with similar or greater computational complexity, employing high UTD ratios:

- ***REDQ*** (Chen et al., 2021): State-of-the-art, model-free algorithm on the OpenAI Gym suite. This algorithm employs multiple parameterized action-value functions and utilizes clipped double Q-learning over a sampled pair of outputs to compute the critic's targets.
- ***MBPO*** (Janner et al., 2019): State-of-the-art, model-based algorithm on the OpenAI Gym suite. This algorithm learns a large ensemble of world models and employs a *Dyna*-style (Sutton, 1991) optimization procedure to train the policy with short, generated rollout data.
- ***SAC-20***: Simple *SAC* extension where we increase the UTD ratio to twenty.

Results. We evaluate *GPL-SAC* compared to the described baselines on five of the more challenging environments from the OpenAI Gym suite (Brockman et al., 2016), involving complex locomotion problems from proprioceptive observations. We evaluate the performance by collecting the returns over five evaluation episodes every 1000 environment steps. In Figure 3, we provide visualizations of the different performance curves. We extend this analysis in Appendix C.

GPL-SAC is consistently the best performing algorithm on all environments, setting new state-of-the-art results for this benchmark at the time of writing. Moreover, the performance gap is greater for tasks with larger state and action spaces. We motivate this by noting that all baselines use fixed strategies to counteract overestimation bias, thus, requiring overly pessimistic estimates of the returns to avoid instabilities. Hence, the resulting policies are likely overly conservative, hindering exploration and efficiency, with larger effects on the more complex tasks. For instance, on the *Humanoid* task, *GPL-SAC* remarkably surpasses a score of 5000 by the 100K experience threshold, more than nine times faster than *SAC* and more than 2.5 times faster than *REDQ*.

Ablations. Furthermore, we provide results from ablating the main components that differentiate *GPL-SAC* from *SAC-20* on a subset of environments. Namely, we evaluate the contributions from learning the uncertainty regularizer, increasing the number of action-value functions, and using the improved modern action-value network architecture. In Figure 4, we provide visualizations of the performance curves. Empirically, each component consistently improves performance, with higher contributions from learning the uncertainty regularizer and increasing the critic's ensemble size. We provide many further in-depth ablation studies as part of Appendix D.

Table 1: Results summary for the pixel observations experiments on the DeepMind Control Suite

Evaluation milestone	1.5M environment frames			3M environment frames		
Algorithm / Task	DrQv2	GPL-DrQ	GPL-DrQ-Expl+	DrQv2	GPL-DrQ	GPL-DrQ-Expl+
Acrobot swingup	$\mathbf{277 \pm 39}$	246 ± 31	$\mathbf{283 \pm 49}$	414 ± 34	393 ± 29	$\mathbf{446 \pm 33}$
Cartpole swingup sparse	475 ± 388	740 ± 123	$\mathbf{780 \pm 25}$	503 ± 411	$\mathbf{837 \pm 15}$	824 ± 33
Cheetah run	771 ± 24	$\mathbf{837 \pm 29}$	$\mathbf{830 \pm 25}$	873 ± 53	$\mathbf{903 \pm 1}$	$\mathbf{902 \pm 2}$
Finger turn easy	794 ± 159	$\mathbf{860 \pm 74}$	810 ± 84	$\mathbf{946 \pm 16}$	$\mathbf{945 \pm 18}$	$\mathbf{952 \pm 15}$
Finger turn hard	484 ± 156	$\mathbf{587 \pm 238}$	$\mathbf{615 \pm 233}$	$\mathbf{923 \pm 17}$	893 ± 59	$\mathbf{918 \pm 24}$
Hopper hop	198 ± 101	$\mathbf{242 \pm 56}$	$\mathbf{252 \pm 76}$	240 ± 123	296 ± 52	$\mathbf{349 \pm 90}$
Quadruped run	385 ± 214	$\mathbf{564 \pm 54}$	$\mathbf{589 \pm 71}$	504 ± 279	$\mathbf{712 \pm 51}$	$\mathbf{725 \pm 112}$
Quadruped walk	591 ± 270	$\mathbf{834 \pm 46}$	$\mathbf{826 \pm 35}$	897 ± 46	$\mathbf{918 \pm 16}$	$\mathbf{912 \pm 16}$
Reach duplo	$\mathbf{219 \pm 6}$	$\mathbf{217 \pm 6}$	$\mathbf{219 \pm 5}$	$\mathbf{227 \pm 2}$	$\mathbf{226 \pm 1}$	$\mathbf{225 \pm 3}$
Reacher easy	961 ± 13	951 ± 21	$\mathbf{968 \pm 8}$	952 ± 17	$\mathbf{957 \pm 12}$	$\mathbf{962 \pm 7}$
Reacher hard	$\mathbf{813 \pm 122}$	$\mathbf{790 \pm 103}$	$\mathbf{798 \pm 114}$	$\mathbf{957 \pm 13}$	946 ± 38	934 ± 18
Walker run	569 ± 273	574 ± 275	$\mathbf{713 \pm 5}$	617 ± 296	618 ± 298	$\mathbf{782 \pm 10}$
Top performance count	$3/12$	$8/12$	$\mathbf{11/12}$	$4/12$	$7/12$	$\mathbf{10/12}$

6.2 Continuous control from pixels

GPL-DrQ. We also integrate *GPL* with a recent version of *Data-regularized Q* (*DrQv2*) (Yarats et al., 2021a), an off-policy, model-free algorithm achieving state-of-the-art performance for pixel-based control problems. *DrQv2* combines image augmentation from *DrQ* (Yarats et al., 2021b) with several advances such as n-step returns (Sutton & Barto, 2018) and scheduled exploration noise (Amos et al., 2021). Again, we substitute *DrQv2*'s clipped double Q-learning with our uncertainty regularizer. To bolster exploration in pixel-based environments, we also integrate pessimism annealing from Section 5.3, with λ_{opt} linearly decayed from 0.5 to 0.0 together with the exploration noise in *DrQv2*. We leave the rest of the hyper-parameters and models unaltered to evaluate the generality of applying *GPL*. We name the resulting algorithms *GPL-DrQ* and *GPL-DrQ-Expl+*, respectively.

Results. We evaluate *GPL-DrQ* and *GPL-DrQ-Expl+* on the environments from the DeepMind Control Suite (Tassa et al., 2018) modified to yield pixel observations. We use the medium benchmark evaluation as described by Yarats et al. (2021a), consisting of 12 complex tasks involving control problems with hard exploration and sparse rewards. In Table 1 we report the returns obtained after experiencing 3M and 1.5M environment frames. For each run, we average the returns from 100 evaluation episodes collected in the 100K steps preceding each of these milestones. We highlight the top performances that are within half a standard deviation from the highest mean return. We provide visualizations of the relative performance curves in Appendix C.

Both *GPL-DrQ* and *GPL-DrQ-Expl+* significantly improve the performance of *DrQv2* in most tasks. In particular, *DrQv2* sporadically yields underperforming returns, likely due to a lack of exploration from its overly pessimistic critic[1]. *GPL* generally appears to resolve this issue, while pessimism annealing further aids precisely in the tasks where under-exploration is more frequent. Overall, these results show both the generality and effectiveness of *GPL* to improve the current state-of-the-art through simple integrations by providing a framework to better capture and exploit bias.

7 Discussion and future work

We propose *Generalized Pessimism Learning (GPL)*, a strategy that adaptively *learns* a penalty to recover an unbiased performance objective for off-policy RL. Unlike traditional methods, *GPL* achieves training stability without necessitating overly pessimistic estimates of the target returns, thus, improving convergence and exploration. We show that simple integrations of *GPL* with modern algorithms yield state-of-the-art results for both proprioceptive and pixel-based control tasks. Moreover, *GPL*'s penalty has a natural generalization to different distributional critics and variational representations of the weights posterior. Hence, our method has the potential to facilitate research going beyond action-value functions and model ensembles for continuous control, two exciting extensions we leave for future work.

[1]This instability was also observed by *DrQv2*'s authors after re-collecting their results.

ETHICS STATEMENT

This work aims to improve the efficiency and generality of reinforcement learning algorithms. Thus, it may contribute to future effective deployments of autonomous agents for real-world applications. Furthering automation has the potential to provide many economic and social benefits to humanity. However, if unregulated, such advancements could accentuate societal inequalities, worsen the consequences of harmful misuse, and have a tangible environmental impact.

REPRODUCIBILITY STATEMENT

In the supplementary material, we provide a compact version of our source code that enables the reproduction of all experiments in this work. After review, we will open-source a comprehensive, documented repository for this project to facilitate future extensions.

REFERENCES

Brandon Amos, Samuel Stanton, Denis Yarats, and Andrew Gordon Wilson. On the model-based stochastic value gradient for continuous reinforcement learning. In *Learning for Dynamics and Control*, pp. 6–20. PMLR, 2021.

Oron Anschel, Nir Baram, and Nahum Shimkin. Averaged-dqn: Variance reduction and stabilization for deep reinforcement learning. In *International conference on machine learning*, pp. 176–185. PMLR, 2017.

Christopher G Atkeson and Juan Carlos Santamaria. A comparison of direct and model-based reinforcement learning. In *Proceedings of international conference on robotics and automation*, volume 4, pp. 3557–3564. IEEE, 1997.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*, pp. 30–37. Elsevier, 1995.

Sarah Bechtle, Artem Molchanov, Yevgen Chebotar, Edward Grefenstette, Ludovic Righetti, Gaurav Sukhatme, and Franziska Meier. Meta learning via learned loss. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 4161–4168. IEEE, 2021.

Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, pp. 449–458. PMLR, 2017.

Johan Björck, Xiangyu Chen, Christopher De Sa, Carla P Gomes, and Kilian Weinberger. Low-precision reinforcement learning: Running soft actor-critic in half precision. In *International Conference on Machine Learning*, pp. 980–991. PMLR, 2021.

Johan Bjorck, Carla P Gomes, and Kilian Q Weinberger. Towards deeper deep reinforcement learning. *arXiv preprint arXiv:2106.01151*, 2021.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Ronen I Brafman and Moshe Tennenholtz. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3(Oct):213–231, 2002.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Richard Y Chen, Szymon Sidor, Pieter Abbeel, and John Schulman. Ucb exploration via q-ensembles. *arXiv preprint arXiv:1706.01502*, 2017.

Xinyue Chen, Che Wang, Zijian Zhou, and Keith Ross. Randomized ensembled double q-learning: Learning fast without a model. *arXiv preprint arXiv:2101.05982*, 2021.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *arXiv preprint arXiv:1805.12114*, 2018.

Kamil Ciosek, Quan Vuong, Robert Loftin, and Katja Hofmann. Better exploration with optimistic actor-critic. *arXiv preprint arXiv:1910.12807*, 2019.

John D Co-Reyes, Yingjie Miao, Daiyi Peng, Esteban Real, Sergey Levine, Quoc V Le, Honglak Lee, and Aleksandra Faust. Evolving reinforcement learning algorithms. *arXiv preprint arXiv:2101.03958*, 2021.

Will Dabney, Mark Rowland, Marc G Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Bo Dai, Albert Shaw, Lihong Li, Lin Xiao, Niao He, Zhen Liu, Jianshu Chen, and Le Song. Sbeed: Convergent reinforcement learning with nonlinear function approximation. In *International Conference on Machine Learning*, pp. 1125–1134. PMLR, 2018.

Catherine Forbes, Merran Evans, Nicholas Hastings, and Brian Peacock. *Statistical distributions*. John Wiley & Sons, 2011.

Scott Fujimoto, Herke Van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. *arXiv preprint arXiv:1802.09477*, 2018.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018a.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018b.

Hado Hasselt. Double q-learning. *Advances in neural information processing systems*, 23:2613–2621, 2010.

Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *arXiv preprint arXiv:1906.08253*, 2019.

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

Roger Koenker and Kevin F Hallock. Quantile regression. *Journal of economic perspectives*, 15(4): 143–156, 2001.

Aviral Kumar, Abhishek Gupta, and Sergey Levine. Discor: Corrective feedback in reinforcement learning via distribution correction. *arXiv preprint arXiv:2003.07305*, 2020.

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

Qingfeng Lan, Yangchen Pan, Alona Fyshe, and Martha White. Maxmin q-learning: Controlling the estimation bias of q-learning. *arXiv preprint arXiv:2002.06487*, 2020.

Donghun Lee, Boris Defourny, and Warren B Powell. Bias-corrected q-learning to control max-operator bias in q-learning. In *2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pp. 93–99. IEEE, 2013.

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

Fred C Leone, Lloyd S Nelson, and RB Nottingham. The folded normal distribution. *Technometrics*, 3(4):543–550, 1961.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

A Rupam Mahmood, Dmytro Korenkevych, Gautham Vasan, William Ma, and James Bergstra. Benchmarking reinforcement learning algorithms on real-world robots. *arXiv preprint arXiv:1809.07731*, 2018.

Shie Mannor, Duncan Simester, Peng Sun, and John N Tsitsiklis. Bias and variance approximation in value function estimates. *Management Science*, 53(2):308–322, 2007.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Ted Moskovitz, Jack Parker-Holder, Aldo Pacchiano, Michael Arbel, and Michael I. Jordan. Tactical optimism and pessimism for deep reinforcement learning, 2021.

Junhyuk Oh, Matteo Hessel, Wojciech M Czarnecki, Zhongwen Xu, Hado van Hasselt, Satinder Singh, and David Silver. Discovering reinforcement learning algorithms. *arXiv preprint arXiv:2007.08794*, 2020.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29:4026–4034, 2016.

Mark D Pendrith, Malcolm RK Ryan, et al. *Estimator variance in reinforcement learning: Theoretical problems and practical solutions*. University of New South Wales, School of Computer Science and Engineering, 1997.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. 2014.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

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

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pp. 1057–1063, 2000.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Sebastian Thrun and Anton Schwartz. Issues in using function approximation for reinforcement learning. In *Proceedings of the Fourth Connectionist Models Summer School*, pp. 255–263. Hillsdale, NJ, 1993.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Thirtieth AAAI conference on artificial intelligence*, 2016.

Tingwu Wang and Jimmy Ba. Exploring model-based planning with policy networks. *arXiv preprint arXiv:1906.08649*, 2019.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pp. 10524–10533. PMLR, 2020.

Zhongwen Xu, Hado van Hasselt, Matteo Hessel, Junhyuk Oh, Satinder Singh, and David Silver. Meta-gradient reinforcement learning with an objective discovered online. *arXiv preprint arXiv:2007.08433*, 2020.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. *arXiv preprint arXiv:2107.09645*, 2021a.

Denis Yarats, Ilya Kostrikov, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=GY6-6sTvGaf.

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

APPENDIX

A PENALTIES FOR BIAS COUNTERACTION

Generalized Pessimism Learning makes use of the uncertainty regularizer penalty to counteract biases arising from the off-policy RL optimization process. Here, we further analyze some of the properties and relationships of our penalty with alternatives from prior works. We will consider the common case where the critic is parameterized as an ensemble of $N \geq 2$ action-value functions $\{Q^{\pi}_{\phi_i}\}^{N}_{i=1}$. To simplify our analysis, we assume that the different action-value predictions for a given input are independently drawn from a Gaussian distribution with some variance $\sigma^2_{Q(s,a)}$. We motivate this latter assumption by the *Central Limit Theorem*, given the many sources of stochasticity affecting the training process of each action-value model in the ensemble. To simplify our notation, we will drop input dependencies when doing so will not compromise clarity, e.g., $\sigma^2_{Q(s,a)} \Rightarrow \sigma^2_Q$.

A.1 EXPECTED PENALIZATION OF THE UNCERTAINTY REGULARIZER

The uncertainty regularizer penalty $p_\beta(s, a, \phi, \theta)$ is parameterized as a weighted model of epistemic uncertainty. Particularly, we propose to measure epistemic uncertainty with the expected Wasserstein distance between the return distributions predicted by the critic. This is formally described in Eqn. 8 from Section 5. In the common case where we parameterize the critic with an ensemble of action-value functions, we treat each as a Dirac delta function approximation of the return distribution. Consequently, we can estimate the expected Wasserstein distance by averaging over all the $N^2 - N$ absolute differences between different Dirac locations predicted by the action-value functions:

$$\begin{aligned} p_\beta(s, a, \phi, \theta) &= \beta \times \mathbb{E}_{a\sim\pi_\theta(s),\phi_1,\phi_2}\left[W(Z^{\pi}_{\phi_1}(s, a), Z^{\pi}_{\phi_2}(s, a))\right] \\ &= \beta \times \mathbb{E}_{a\sim\pi_\theta(s),\phi_1,\phi_2}\left[|Q^{\pi}_{\phi_1}(s, a) - Q^{\pi}_{\phi_2}(s, a)|\right] \\ &\approx \frac{\beta}{N^2 - N}\sum_{i=1}^{N}\sum_{j\neq i}^{N}|Q^{\pi}_{\phi_i}(s, a) - Q^{\pi}_{\phi_j}(s, a)|. \end{aligned} \tag{12}$$

From the independence assumption of the ensemble models, the differences in the action-value predictions will also follow a Gaussian distribution:

$$D_{ij} = Q^{\pi}_{\phi_i}(s, a) - Q^{\pi}_{\phi_j}(s, a) \sim N(0, 2\sigma^2_Q). \tag{13}$$

Each absolute difference $W_{ij} = |D_{ij}|$ will then follow a *folded normal* distribution (Leone et al., 1961), with moments given by:

$$\mathbb{E}[W] = \mu_W = \frac{\sqrt{2}}{\pi}\sigma_D = \frac{2}{\pi}\sigma_Q, \quad var(W) = \sigma_W = \sigma^2_D - \mu^2_W = \left(2 - \frac{4}{\pi^2}\right)\sigma^2_Q. \tag{14}$$

Consequently, the expected penalization of the uncertainty regularizer immediately follows:

$$\mathbb{E}[p_\beta] = \beta\mu_W = \frac{2\beta}{\pi}\sigma_Q. \tag{15}$$

A.2 EXPECTED PENALIZATION OF THE POPULATION STANDARD DEVIATION

Ciosek et al. (2019) and Moskovitz et al. (2021) make use of a weighted model of an alternative epistemic uncertainty measure to define a penalty as $p^s_\beta(s, a, \phi, \theta) = \beta \times s(s, a, \phi, \theta)$. In particular, they make use of the population standard deviation $s(s, a, \phi, \theta)$, treating the critic's ensemble predictions as independently sampled action-values:

$$s(s, a, \phi, \theta) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(Q^{\pi}_{\phi_i}(s, a) - \mu_{Q_\phi(s,a)}\right)^2}, \quad \textit{where} \quad \mu_{Q_\phi} = \frac{1}{N}\sum_{i=1}^{N}Q^{\pi}_{\phi_i}. \tag{16}$$

We can rewrite the population standard deviation measure in terms of the square root of the sum of N squared difference terms D_i:

$$s = \sqrt{\sum_{i=1}^{N} D_i^2}, \quad \textit{where} \quad D_i = \frac{1}{\sqrt{N}} \left(Q_{\phi_i}^{\pi} - \mu_{Q_\phi} \right). \tag{17}$$

Moreover, we can rewrite each difference term as a weighted sum of the N different ensemble predictions:

$$D_i = \frac{1}{\sqrt{N}} \left(Q_{\phi_i}^{\pi} - \frac{1}{N} \sum_{i=1}^{N} Q_{\phi_i}^{\pi} \right) = \frac{1}{\sqrt{N}} \left(\frac{N-1}{N} Q_{\phi_i}^{\pi} - \sum_{j \neq i} \frac{Q_{\phi_j}}{N} \right). \tag{18}$$

From the independence assumption of the ensemble models, it follows that each D_i is also a Gaussian random variable with moments given by the basic properties for the sums of independent random variables:

$$\mathbb{E}[D] = \mu_D = 0, \quad var(D) = \sigma_D^2 = \frac{1}{N} \left(\frac{(N-1)^2}{N^2} \sigma_Q^2 + \frac{N-1}{N^2} \sigma_Q^2 \right) = \frac{N-1}{N^2} \sigma_Q^2. \tag{19}$$

Hence, by simply scaling the population standard deviation, we can express it as a sum of standard normal random variables:

$$\begin{aligned} \frac{N}{\sigma_Q \sqrt{N-1}} s &= \sqrt{\sum_{i=1}^{N} \left(\frac{N}{\sigma_Q \sqrt{N-1}} D_i \right)^2} \\ &= \sqrt{\sum_{i=1}^{N} Z_i^2}, \quad \textit{where} \quad Z \sim N(0,1). \end{aligned} \tag{20}$$

This enables to relate s to a *Chi* distribution (Forbes et al., 2011) with parameter N:

$$\frac{N}{\sigma_Q \sqrt{N-1}} s \sim \chi_N.$$

Consequently, the expected value of p_β^s can be obtained from scaling the expected value of a *Chi* distribution:

$$\mathbb{E}[p_\beta^s] = \beta \mathbb{E}[s] = \frac{\beta \sqrt{N-1}}{N} \mathbb{E}[\chi_N] \sigma_Q = \frac{\beta \sqrt{N-1}}{N} \times \frac{\sqrt{2} \Gamma\left(\frac{N+1}{2}\right)}{\Gamma\left(\frac{N}{2}\right)} \sigma_Q. \tag{21}$$

Comparing the expected value of this penalty with the expected value of the uncertainty regularizer penalty from Eqn. 15, we note a few key facts. Both quantities are linearly proportional to the standard deviation of the action-value predictions σ_Q. However, for the population standard deviation penalty, the scale of the proportionality is dependent on the number of action-value models used to parameterize the critic. Hence, unlike for the uncertainty regularizer penalty, the critic's parameterization directly affects the expected penalization magnitude and should influence design choices regarding the parameter β.

A.3 Relationship with clipped double Q-learning

We can rewrite the ubiquitous clipped double Q-learning penalization practice (Fujimoto et al., 2018) using the uncertainty regularizer penalty. In particular, we can specify the clipped double Q-learning action-value targets from Eqn. 6 in terms of the difference between the mean action-value prediction and a penalty:

$$\begin{aligned} \hat{Q}_{\phi_{min}}(s,a)^{\pi} &= \min\left(Q_{\phi_1}^{\pi}(s,a), Q_{\phi_2}^{\pi}(s,a) \right) = \bar{Q}_{\phi}^{\pi}(s,a) - p_{min}(s,a,\phi,\theta), \\ \textit{where} \quad \bar{Q}_{\phi}^{\pi} &= \frac{1}{N} \sum_{i=1}^{N} Q_{\phi_i}^{\pi}, \quad p_{min} = \min\left(Q_{\phi_1}^{\pi}, Q_{\phi_2}^{\pi} \right) - \bar{Q}_{\phi}^{\pi} = \frac{1}{2} |Q_{\phi_1}^{\pi} - Q_{\phi_2}^{\pi}|. \end{aligned} \tag{22}$$

Similarly, for $N = 2$ our uncertainty regularizer reduces to:

$$p_\beta = \frac{\beta}{2}\left(|Q^\pi_{\phi_1} - Q^\pi_{\phi_2}| + |Q^\pi_{\phi_2} - Q^\pi_{\phi_1}|\right) = \beta|Q^\pi_{\phi_1} - Q^\pi_{\phi_2}|. \tag{23}$$

Thus, $p_{min} = p_\beta$ for $\beta = 0.5$. As shown earlier in this section, the expected magnitude of the uncertainty regularizer penalty is not dependent on the number of action-value functions. Hence, our penalty allows us to extend the expected regularization induced by clipped double Q-learning for $N > 2$, simply fixing $\beta = 0.5$. This would not be possible using the population standard deviation penalty since its expected magnitude is dependent on the number of action-value functions employed.

Table 2: Hyper-parameters used for *GPL-SAC*

SAC hyper-parameters	
Replay data buffer size	1000000
Batch size	256
Minimum data before training	5000
Random exploration steps	5000
Optimizer	*Adam* (Kingma & Ba, 2014)
Policy/critic learning rate	0.0003
Policy/critic β_1	0.9
Critic UTD ratio	20
Policy UTD ratio	1
Discount γ	0.99
Polyak coefficient ρ	0.995
Hidden dimensionality	256
Nonlinearity	ReLU
Initial entropy coefficient α	1
Entropy coefficient learning rate	0.0001
Entropy coefficient β_1	0.5
Policy target entropy $\hat{H}$	*Hopper* : -1, *HalfCheetah*: -3, *Walker2d*: -3, *Ant*: -4, *Humanoid*: -2
GPL hyper-parameters	
Initial uncertainty regularizer weight β	0.5
Uncertainty regularizer learning rate	0.1
Uncertainty regularizer β_1	0.5

B ALGORITHMIC AND EXPERIMENTAL SPECIFICATIONS

In this section, we provide details regarding the experimental results from the main text. For further information about our efficient implementation, please, refer to the shared code.

B.1 EMPIRICAL BIAS ESTIMATION

In Section 5, we record the evolution of the target action-value bias throughout the RL process by running experiments with two different extensions to the *SAC* algorithm on three OpenAI Gym environments. The first extension makes use of unpenalized optimistic target action-values, while the second extension makes use of a penalty with a magnitude equivalent to the one induced by the clipped double Q-learning targets. Both are implemented through the uncertainty regularizer with a fixed β, as derived in Appendix A. Moreover, both extensions use an update-to-data ratio of twenty and $N = 4$ action-value models to parameterize the critic. The rest of the hyper-parameters follow our implementation of *GPL-SAC*. We record estimates of the action-value bias by collecting transitions from ten evaluation episodes every 1000 environment steps. We obtain each estimate by taking the average difference between the observed discounted returns in the evaluation episodes and the discounted returns from the critic's predictions. In particular, we correct the critic's raw predictions by subtracting the discounted log probabilities of the performed actions, accounting for *SAC*'s modified action-value objective.

B.2 PROPRIOCEPTIVE OBSERVATIONS EXPERIMENTS

Hyper-parameters. We provide a list of the utilized hyper-parameters for *GPL-SAC* in Table 2. The majority of the chosen hyper-parameters follow closely the seminal *SAC* papers (Haarnoja et al., 2018a;b), with some exceptions to improve performance and stability. For instance, consistently with Chen et al. (2021), we employ an ensemble of ten action-value models and increase the update-to-data ratio to twenty. Additionally, we employ a simplified version of the modern architecture from Bjorck et al. (2021) to parameterize the action-value models, as depicted in Figure 5. This

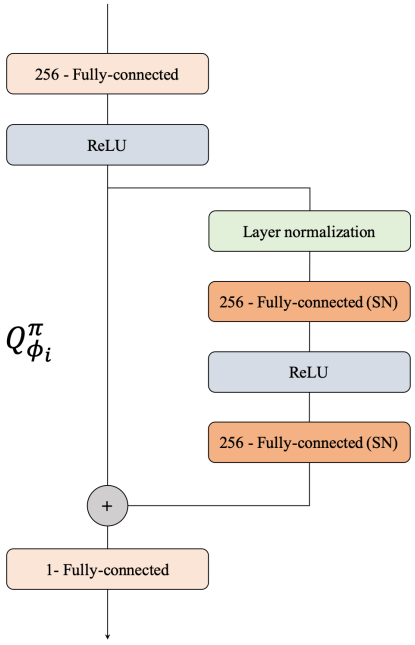

Figure 5: Schematic representation of the modern model architecture used to parameterize the action-value functions in our implementation of *GPL-SAC*. We make use of a single hidden residual block where the fully-connected layers are regularized via spectral normalization.

architecture follows many of the practices introduced by recent work to stabilize transformer training (Xiong et al., 2020). Specifically, we employ a single hidden residual block with layer normalization (Ba et al., 2016) followed by two fully-connected layers regularized with spectral normalization (Miyato et al., 2018). Throughout all architectures, we keep the hidden dimensionality fixed to 256. We initialize the uncertainty regularizer with $\beta = 0.5$ to reflect the penalization magnitude of clipped double Q-learning. Following Haarnoja et al. (2018b), the entropy coefficient α is adaptively updated at each training iteration based on a policy target entropy $\hat{H}$. In particular, this follows a dual optimization procedure to keep the estimated average policy entropy close to $\hat{H}$, as described in Line 17 of Algorithm 1. We utilize increased values of $\hat{H}$ to optimize α, following the choices of Janner et al. (2019). We learn β using dual TD-learning with the same optimizer used for adjusting the value of α.

Baseline results. For the continuous control experiments from proprioceptive states, we employ different baselines to ground our results and provide a comparison with current state-of-the-art algorithms. The reported results for *REDQ* come from re-running Chen et al. (2021)'s original implementation with the provided hyper-parameters. The reported results for *MBPO* come instead from the original paper, as publicly shared by Janner et al. (2019). The reported results for *SAC-20* come from running the same base implementation as *GPL-SAC*, with a few differences in the listed hyper-parameters. Namely, *SAC-20* uses an ensemble of $N = 2$ action-value models with the classical 3-layer fully-connected architecture from Haarnoja et al. (2018b) and uses an uncertainty regularizer penalty with a fixed parameter $\beta = 0.5$.

B.3 Pixel observations experiments

Hyper-parameters. We provide a list of the utilized hyper-parameters for *GPL-DrQ* in Table 3. All the hyper-parameters shared with *DrQv2* follow the values provided by Yarats et al. (2021a), while the uncertainty regularizer and optimizer for β follow the same specifications as in *GPL-SAC*. Additionally, *GPL-DRQ-Expl+* linearly decays the optimistic shift value λ_{opt} from 0.5 down to 0.0 with the same frequency as the exploration noise's standard deviation.

Table 3: Hyper-parameters used for *GPL-DrQ*

DrQv2 hyper-parameters	
Replay data buffer size	1000000 (100000 for *Quadruped run*)
Batch size	256 (512 for *Walker run*)
Minimum data before training	4000
Random exploration steps	2000
Optimizer	*Adam* (Kingma & Ba, 2014)
Policy/critic learning rate	0.0001
Policy/critic β_1	0.9
Critic UTD ratio	0.5
Policy UTD ratio	0.5
Discount γ	0.99
Polyak coefficient ρ	0.99
N-step returns	3 (1 for *Walker run*)
Hidden dimensionality	1024
Feature dimensionality	50
Nonlinearity	ReLU
Initial entropy coefficient α	1
Exploration stddev. clip	0.3
Exploration stddev. schedule	linear: $1 \rightarrow 0.1$ in 500000 steps
GPL hyper-parameters	
Initial uncertainty regularizer weight β	0.5
Uncertainty regularizer learning rate	0.1
Uncertainty regularizer β_1	0.5
Pessimism annealing hyper-parameters	
Optimistic shift value λ_{opt} schedule	linear: $0.5 \rightarrow 0$ in 500000 steps

Baseline results. For the experiments on the DeepMind Control Suite from pixel observations, we compare our extensions with the base *DrQv2* algorithm. Since the provided results for *DrQv2* had inconsistent numbers of repetitions, we recollected the results by running the experiments with Yarats et al. (2021a)'s original implementation. However, in our evaluation, we observed higher variances in the performance for some of the considered tasks than what Yarats et al. (2021a) reported. We shared this inconsistency with *DrQv2*'s authors, and they confirmed the validity of our empirical findings after recollecting the results themselves.

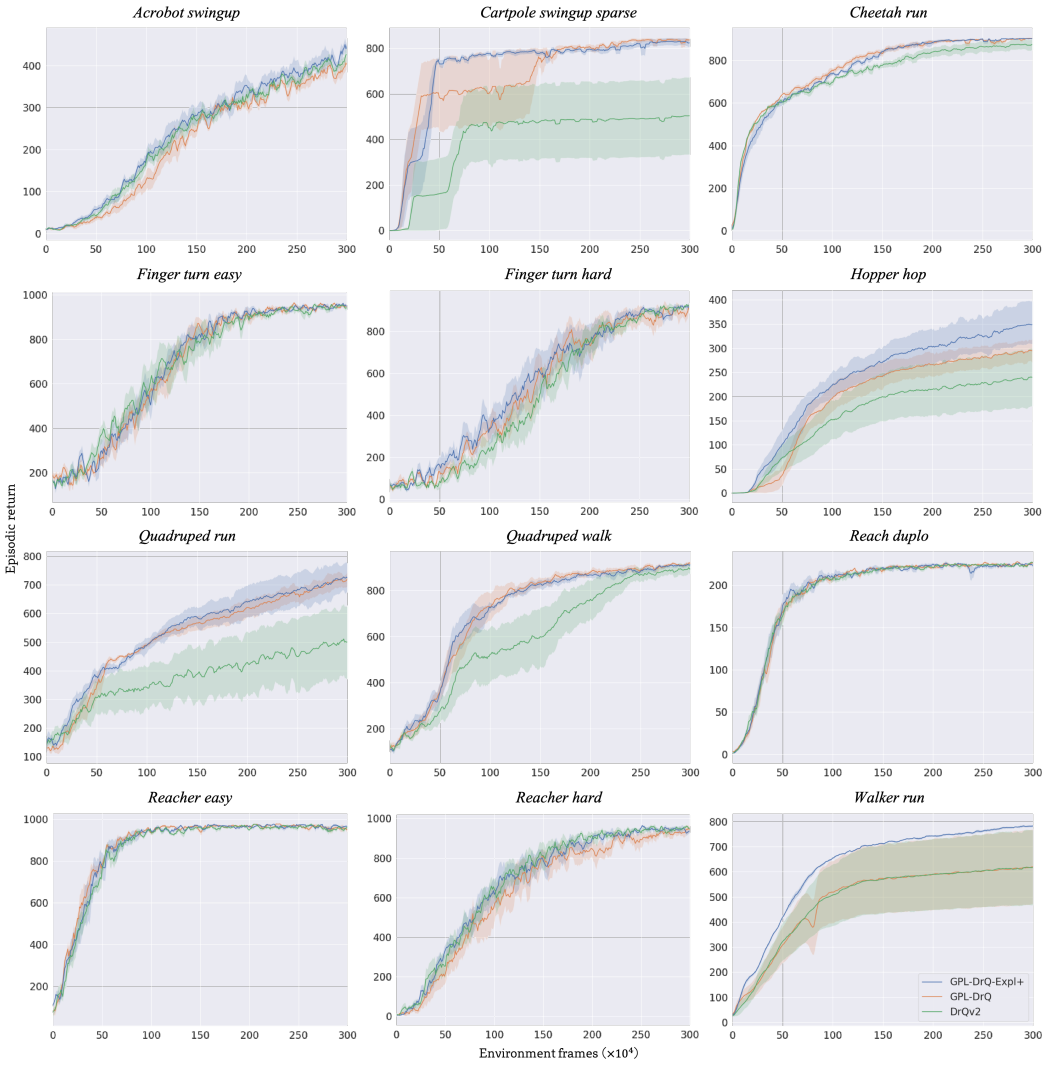

Figure 6: Performance curves for the environments of the DeepMind Control Suite. We report both mean and standard deviation of the episodic returns over five random seeds.

C Main performance results

C.1 OpenAI gym results after 100k experience steps

In Table 4 we compare the performances of the examined algorithms after the milestone of collecting 100K environment steps. We believe this to be an appropriate comparison point for recent algorithms, given their relative sample-efficiency improvements in the OpenAI Gym tasks. For each run, we record the average returns from 25 evaluation episodes collected in the preceding 5000 environment steps. These results highlight once again the superior sample efficiency of *GPL-SAC*, pushing the performance boundary of off-policy methods. *GPL-SAC* is also more consistent than the considered baselines in most tasks, as shown by the lower relative standard deviations. As mentioned in Section 6 of the main text, we attribute the performance gap mainly to the ability of *GPL* to prevent the accumulation of overestimation bias without requiring overly pessimistic targets. In particular, the principal identified downsides of overly pessimistic targets are two-fold. Firstly, they slow down reward propagation and introduce further errors in the TD-targets, potentially leading to suboptimal convergence of the relative action-value functions (Kumar et al., 2020). Secondly, they

Table 4: Results for the proprioceptive observations experiments on the OpenAI Gym suite after collecting 100K experience steps

Algorithm / Task	SAC-20	REDQ	MBPO	GPL-SAC (Ours)
Hopper-v2	2694 ± 902	3007 ± 471	3262 ± 197	$\mathbf{3386 \pm 92}$
Halfcheetah-v2	5822 ± 728	5625 ± 431	9501 ± 331	$\mathbf{9685 \pm 658}$
Walker2d-v2	2101 ± 876	1937 ± 968	3377 ± 529	$\mathbf{3662 \pm 360}$
Ant-v2	482 ± 101	3160 ± 1213	1624 ± 447	$\mathbf{4933 \pm 333}$
Humanoid-v2	493 ± 94	1460 ± 689	555 ± 61	$\mathbf{5155 \pm 191}$
Normalized average	0.35 ± 0.28	0.48 ± 0.24	0.52 ± 0.30	$\mathbf{0.81 \pm 0.12}$

induce overly conservative policies that prefer low-uncertainty behavior, hindering exploration in stochastic environments (Ciosek et al., 2019).

C.2 DEEPMIND CONTROL SUITE PERFORMANCE CURVES

In Figure 6 we provide the performance curves for the pixel-based medium benchmark tasks from the DeepMind Control Suite, as specified by Yarats et al. (2021a). These visualizations complement the results summary provided in Section 6 and further highlight the effectiveness of *GPL* and the proposed pessimism annealing procedure. In accordance with our earlier analysis, our methods yield improved performance and robustness in the harder exploration environments, where overly pessimistic action-value targets are expectedly more detrimental.

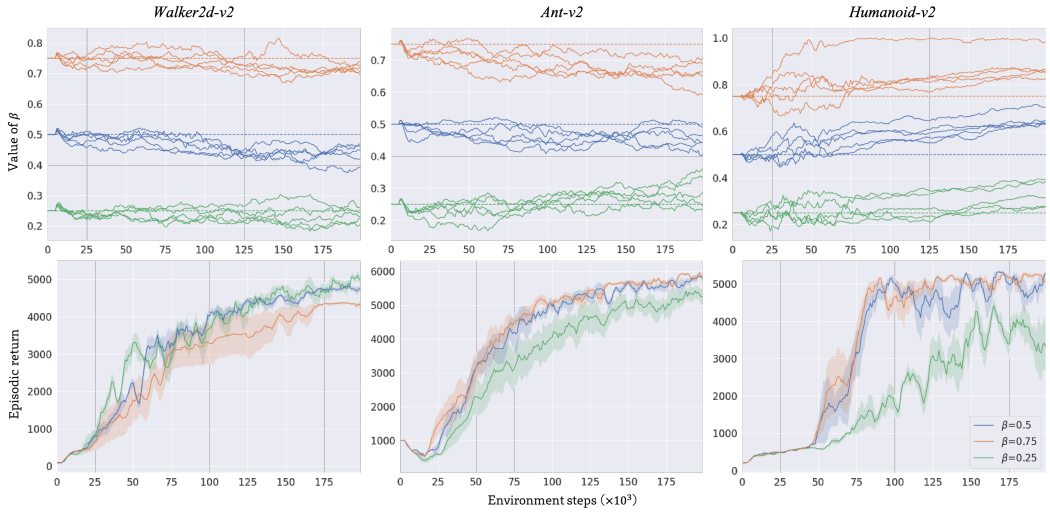

Figure 7: Analysis of the uncertainty regularizer parameter. We consider three versions of *GPL-SAC* with different initial values for β. We show the evolution of β throughout optimization in each experiment (Top). We also provide the relative performance curves for each initial setting (Bottom).

D EXTENDED EMPIRICAL ANALYSIS

We provide an extended empirical analysis of *Generalized Pessimism Learning*. In particular, we study the effects of different components and parameters on the stability and efficiency of our *GPL-SAC* algorithm. We focus our analysis on a representative subset of the considered OpenAI Gym tasks. We follow the same experimental protocol as described in Section 6 of the main text.

D.1 UNCERTAINTY REGULARIZER ANALYSIS

In all main experiments, we initialize the uncertainty regularizer with $\beta = 0.5$. This setting yields the same initial expected regularization magnitude as the ubiquitous clipped double Q-learning method. Hence, to better understand the optimization properties of *GPL*, we consider instantiating *GPL-SAC* with three different initial values for β, namely, $\beta = 0.25$, $\beta = 0.5$, and $\beta = 0.75$. For these settings, we record both the evolution of β throughout learning and the performance of the resulting algorithms. This analysis aims to evaluate the sensitivity of *GPL* to the initial value of β and the effectiveness of the dual TD-learning procedure. In particular, this simple experimental setting should elucidate some of the properties of the actual implementation of *GPL*, where the initial unbiasedness assumption used to motivate dual TD-learning does not necessarily hold. Moreover, the learned values of β should provide insights regarding the bias arising from interactions of off-policy reinforcement learning methods and the different examined tasks.

Parameter evolution. In Figure 7 (Top), we show the value of β collected throughout training in each experiment with the examined initial settings. For each task, β appears to follow a recognizable trend of convergence towards some particular distinct range of values. This range appears to be influenced by the environment's complexity, with β converging to lower values for *Walker2d* and increasingly higher values for *Ant* and *Humanoid*, respectively. However, β appears to adapt rather slowly, seldom reaching stability for distant initialization values in the examined experience regime. Our intuition is that this phenomenon is mainly due to two characteristics of pessimism learning. The first characteristic is that bias in the target action-value predictions can arise simply due to the stochasticity of the RL process dynamics for any value of β. Hence, there will always be a stochastic component from the dual-TD learning signal introducing noise to β's optimization. The second characteristic is that, in practice, dual TD-learning occurs at a slower rate than TD-learning itself. Hence, as discussed in Section 5, given an imperfect initialization of β, part of the target bias might leak into the online action-value models in the first iterations of training. While the iterative

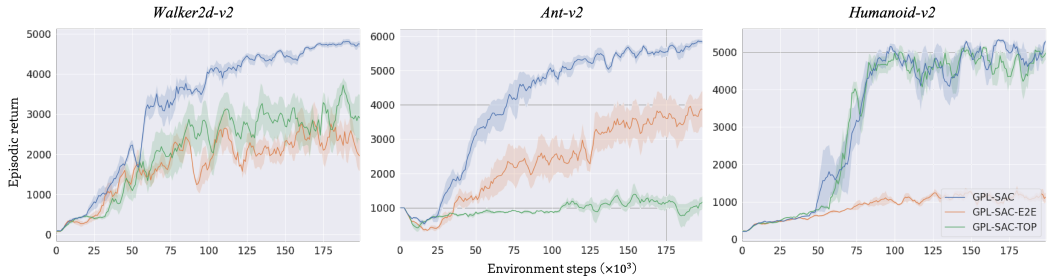

Figure 8: Performance curves showing the effects of optimizing the uncertainty regularizer with alternative strategies in *GPL-SAC*. We consider learning β to minimize the estimated bias end-to-end and maximize immediate returns improvements via a bandit-based optimization from prior work.

interactions between dual TD-learning and TD-learning still pushes the value of β to mitigate bias, consistently with our inituition, training takes longer to reach expectedly unbiased targets from a dimmed-down signal in the 'unbiasing' learning direction.

Performance results. In Figure 7 (Bottom), we show the performance curves for the examined initial settings. While *GPL-SAC* recovers strong performance for most tested initializations, we can see robustness improve when we initialize β closer to its convergence range. Specifically, $\beta = 0.75$ appears to work best for *Humanoid*, $\beta = 0.5$ for *Ant*, and $\beta = 0.25$ for *Walker*. These results provide further evidence that one of the main factors determining optimal pessimism is task complexity, with harder tasks requiring more conservative targets for better stabilization. Thus, these results further highlight the importance of an adaptive strategy for off-policy RL to achieve proper bias counteraction in arbitrary environments.

D.2 Alternative uncertainty regularizer optimizations

To evaluate the relative empirical effectiveness of dual TD-learning, we implement and test two alternative optimization strategies for the uncertainty regularizer p_β. In particular, we replace the dual TD-learning procedure in *GPL-SAC* with each strategy and compare the resulting performances.

GPL-SAC-E2E. First, we consider learning β to minimize end-to-end the squared norm of the expected target action-value bias approximation from Eqn. 10. This optimization strategy could be seen as a more direct approach than dual TD-learning, resulting in the following optimization objective:

$$\begin{gathered} \arg\min_{\beta} \mathbb{E}_{s,a,s}\left[\hat{B}(s,a,s')^2\right], \quad \textit{where} \\ \hat{B}(s,a,s') = r + \gamma \mathbb{E}_{a'\sim\pi(s')}\left[\bar{Q}^{\pi}_{\phi'}(s',a') - p_\beta(s',a',\phi,\theta)\right] - Q^{\pi}_{\phi}(s,a). \end{gathered} \tag{24}$$

In practice, we optimize this alternative objective in place of dual TD-learning with standard gradient descent. We name the resulting algorithm *GPL-SAC-E2E*.

GPL-SAC-TOP. We also consider the alternative optimization procedure from the *Tactical Optimism and Pessimism (TOP)* algorithm (Moskovitz et al., 2021). In particular, this optimization involves learning an adaptive binary controller that switches on or off a bias correction penalty based on the sample standard deviation of the critic's action-value estimates. The *TOP* algorithm learns this controller as a multi-armed bandit problem, using the difference in consecutive episodic returns as feedback. We transpose this framework to *GPL* by optimizing β over a discrete choice of two possible values, $\beta \in \{0, \sqrt{2}\}$. With these values, the uncertainty regularizer yields the same expected penalization as the optimistic and pessimistic modes of *TOP*, respectively. *TOP*'s authors selected these penalization levels from a vast choice of settings evaluated on the OpenAI Gym tasks. We name the resulting algorithm *GPL-SAC-TOP*.

Results. In Figure 8 we show the performance curves for the evaluated alternative optimization procedures as compared to the original dual TD-learning in *GPL-SAC*:

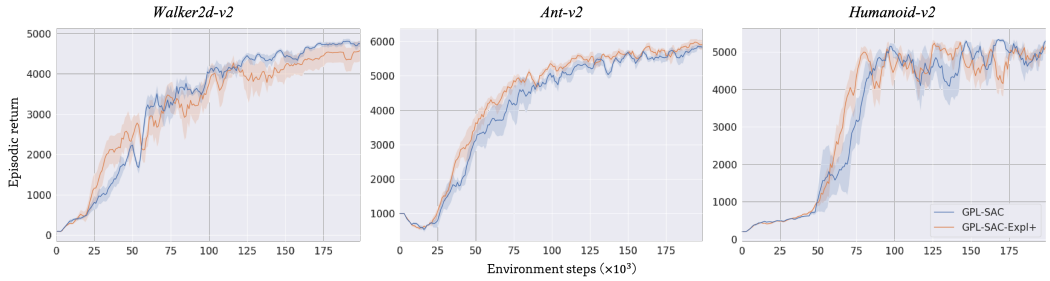

Figure 9: Performance curves showing the effects of applying pessimism annealing to *GPL-SAC* for the considered OpenAI Gym tasks.

GPL-SAC-E2E optimizes the uncertainty regularizer too aggressively, leading to instabilities and suboptimal performance across all tasks. We motivate these results based on two related inconsistencies of end-to-end bias minimization from Eqn. 24. Firstly, this alternative optimization does not consider the effects of the target action-value bias on the changes in the action distribution used for bootstrapping. In particular, modifying the uncertainty regularizer will affect the objective of the policy's optimization from Eqn. 4 and the corresponding distribution of on-policy actions. Secondly, even considering a fixed policy, this end-to-end strategy assumes that β linearly affects the bias. However, this is not the case due to the non-linear compounding effects from the bootstrapping in the TD-recursions. Hence, as empirically confirmed by the superior performance, framing bias minimization as a dual problem is a more general and appropriate formulation.

On the other hand, *GPL-SAC-TOP* matches the performance of dual TD-learning on the *Humanoid* task but severely underperforms on the other two tasks. For *Humanoid*, *TOP*'s optimization strategy quickly converges to sampling $\beta = \sqrt{2}$ consistently, as $\beta = 0$ causes considerable instabilities. As observed in Appendix D.1, this task benefits from highly penalized targets for stabilization, making this early convergence to a high β effective. However, both *Walker* and *Ant* tasks benefit from moderate levels of pessimism, which the bandit optimization strategy fails to recover in the examined experience regime. This inefficiency stems from the fact that immediate episodic return improvement does not appear to correlate with overall learning efficiency for these tasks. Moreover, the slow update frequency and noisy nature of the controller feedback make *GPL-SAC-TOP* unable to dynamically address arising biases. These results further show the effectiveness of dual TD-learning to directly minimize a source of learning instability and enable consistently efficient learning.

D.3 Pessimism annealing for OpenAI Gym

We experiment with applying pessimism annealing to *GPL-SAC* for the OpenAI Gym tasks. In the same spirit as our previously described application for pixel observations tasks, we linearly decay λ_{opt} from 0.5 to 0.0 in the first 50000 environment steps. We leave the rest of the hyper-parameters unaltered. We name the resulting algorithm *GPL-SAC-Expl+*.

Results. In Figure 9 we provide the performance curves comparing *GPL-SAC* with *GPL-SAC-Expl+*. The shifted uncertainty regularizer marginally improves performance in only two out of three environments. These results indicate that the undirected Gaussian exploration from *SAC* combined with the unbiased targets from *GPL* are already sufficient to ensure effective exploration in the OpenAI Gym tasks. In contrast, pessimism annealing appears to have a more significant effect in our results for the DeepMind Control Suite from Section 6 of the main text, highlighting the harder exploration challenge introduced by these pixel-based tasks.

D.4 Ensemble size

In the ablation study from Section 6 of the main text, the increased number of action-value models in the critic's ensemble appeared to have the greatest overall effect on *GPL-SAC*'s performance. This parameter directly influences the ability to capture the critic's epistemic uncertainty, thus, affecting both the accuracy of the action-value predictions and the variance in the Wasserstein distance

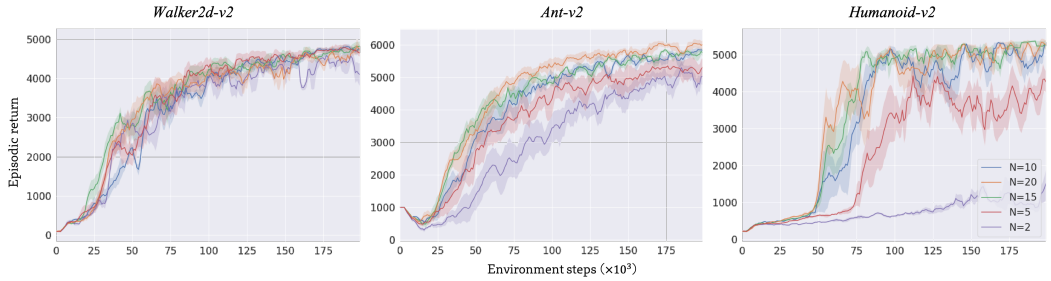

Figure 10: Performance curves showing the effects of varying the ensemble size of the critic in *GPL-SAC*.

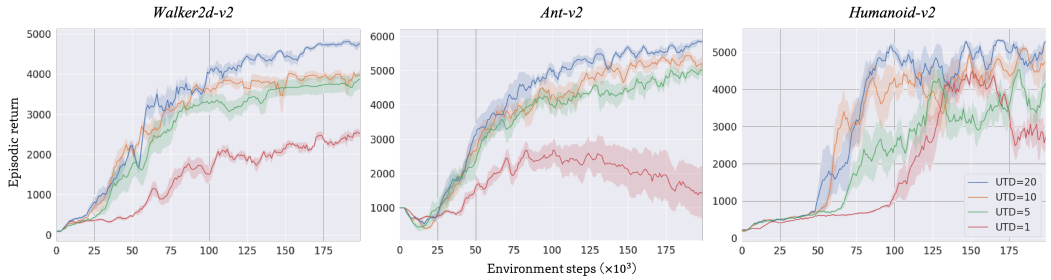

Figure 11: Performance curves showing the effects of lowering the critic's update-to-data ratio in *GPL-SAC*.

estimation for the uncertainty regularizer penalty. Therefore, we extend our empirical analysis by testing *GPL-SAC* with broader numbers of action-value models to understand the relevance of this parameter and the scalability of our algorithm. Namely, we compare instances of *GPL-SAC* with $N = 2$, $N = 5$, $N = 10$, $N = 15$, and $N = 20$ action-value models.

Results. In Figure 10 we provide the performance curves for the different ensemble sizes. Predictably, both learning efficiency and final performance monotonically improve with the ensemble size. Yet, these improvements appear to saturate at different values of N based on the complexity of the underlying environment, showing how harder tasks increasingly rely on accurately representing epistemic uncertainty. In particular, for *Walker2d* training with only $N = 2$ action-value models yields very similar results to training with $N = 20$ action-value models. However, the performance differences are increasingly noticeable for the other tasks that involve significantly larger state and action spaces. For instance, for *Humanoid* the agent is not able to recover meaningful behavior with $N = 2$ and performance saturates only around training with $N = 10$.

D.5 Update-to-data ratio

Following the examined prior state-of-the-art algorithms, *GPL-SAC* uses a critic UTD ratio of 20 across the different OpenAI Gym tasks. This aggressive optimization frequency ensures that the critic encapsulates most information from experience collected at any given point in the RL process. However, since the UTD ratio affects training time almost linearly, we examine its impact on performance. In particular, we compare instances of *GPL-SAC* with UTD ratios of 1, 5, 10, and 20.

Results. In Figure 11 we provide the performance curves for the different examined UTD ratios. Larger UTD ratios yield clear sample efficiency improvements for all tasks. Moreover, using a UTD ratio of 1 in the *Ant* and *Humanoid* environments appears to cause some learning instabilities towards the end of the examined experience regime. These instabilities are likely from the inability of the learning process to quickly incorporate new information from recently collected data into the critic, given the growing size of the replay buffer. Consequently, a slow learning process might produce

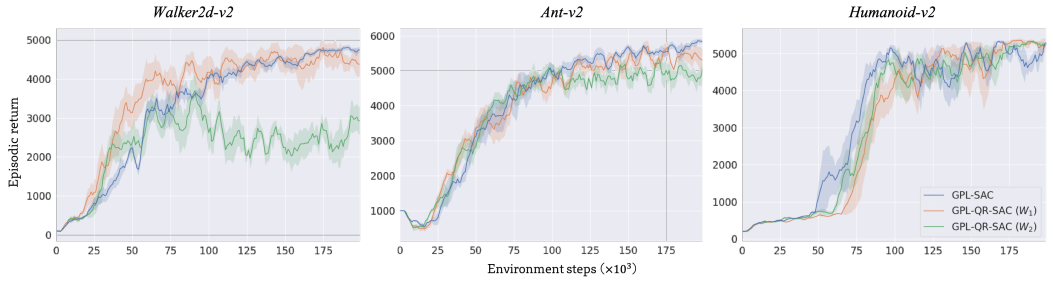

Figure 12: Performance curves showing the effects of employing a more expressive distributional critic based on quantile regression in *GPL-SAC*.

inaccurate return predictions for on-policy behavior, leading to a noisy policy gradient signal and, therefore, a degradation of the policy itself.

D.6 Distributional critic

The uncertainty regularizer penalty is a function of the expected Wasserstein distance between the return distributions predicted by the critic. Hence, it has a natural generalization to more expressive distributional critics beyond ensembles of action-value functions. Therefore, we evaluate the effectiveness of extending *GPL-SAC* by parameterizing the critic using an ensemble of N distributional models $\{Z^\pi_{\phi_i}\}_{i=1}^N$. Following Dabney et al. (2018), we let each model output M quantile locations values parameterizing a quantile distribution. We consider both 1-Wasserstein and 2-Wasserstein distances for the uncertainty regularizer (which for Dirac delta functions are equivalent). For quantile distributions, these distance measures are calculated as follows:

$$\begin{aligned} W_1(Z^\pi_{\phi_i}(s,a), Z^\pi_{\phi_j}(s,a)) &= \sum_{k=1}^{M} \frac{1}{M} |Z^\pi_{\phi_i}(s,a)_k - Z^\pi_{\phi_j}(s,a)_k|, \\ W_2(Z^\pi_{\phi_i}(s,a), Z^\pi_{\phi_j}(s,a)) &= \sqrt{\sum_{k=1}^{M} \frac{1}{M} \left(Z^\pi_{\phi_i}(s,a)_k - Z^\pi_{\phi_j}(s,a)_k\right)^2}. \end{aligned} \tag{25}$$

We propose applying the uncertainty regularizer to each quantile location to compute the target return distribution, $\hat{Z}^\pi_\phi(s,a) \in \mathbb{R}^M$, as:

$$\hat{Z}^\pi_\phi(s,a)_j = Z^\pi_{\phi'}(s,a)_j - p_\beta(s', a', \phi, \theta), \quad \textit{for all} \quad j \in \{1, 2 \dots M\}. \tag{26}$$

We optimize the critic's models with a distributional version of TD-learning (Bellemare et al., 2017) making use of Huber quantile regression (Koenker & Hallock, 2001). We still optimize the policy to maximize the expected returns by averaging over the quantile locations of the target return distribution. Moreover, we still perform dual TD-learning with minimal overheads by averaging all quantile errors computed for quantile regression during the distributional TD-learning updates. In our implementation, we approximate the return distribution with $M = 10$ quantile locations and set the Huber quantile regression coefficient to $\kappa = 1$. We keep all other hyper-parameters and model architectures consistent with *GPL-SAC*. We name the resulting algorithms *GPL-QR-SAC* (W_1/W_2).

Results. In Figure 12 we provide the performance curves comparing both versions of *GPL-QR-SAC* with the original *GPL-SAC* algorithm. *GPL-QR-SAC* (W_1) appears to outperform *GPL-QR-SAC* (W_2), indicating that the 1-Wasserstein distance is a more effective and stable measure of epistemic uncertainty for p_β. However, in the examined tasks, *GPL-QR-SAC* (W_1) performs similarly to *GPL-SAC*, meaning there are no major benefits in parameterizing the critic with more expressive approximations to the return distribution. Yet, these results could also indicate that the returns for the considered OpenAI Gym environments have low stochasticity or that the action-value models already encapsulate all learnable information in the considered training regime. We leave further experimentation and analysis of using *GPL* with distributional critics for future work.

Table 5: Average training times for the tested algorithms and ablations

Proprioceptive observations tasks	Training time (seconds/1000 env. steps)
GPL-SAC ($N = 10$, $UTD = 20$)	76.1
No pessimism learning	74.2
No modern action-value model	51.6
GPL-SAC-Expl+	77.0
GPL-QR-SAC (W_1)	86.6
GPL-QR-SAC (W_2)	88.1
$N = 20$	85.0
$N = 15$	81.2
$N = 5$	62.5
$N = 2$	38.8
$UTD = 10$	41.4
$UTD = 5$	20.1
$UTD = 1$	6.3
REDQ (Original implementation)	183.8
Pixel observations tasks	Training time (seconds/10000 env. steps)
GPL-DrQ-Expl+	112.2
GPL-DrQ	111.3
DrQv2	111.2

D.7 COMPUTATIONAL SCALING

We analyze the computational scaling of our implementations of *Generalized Pessimism Learning*-based algorithms. Particularly, we record the average training time of executing the different considered algorithms for either 1000 environment steps for OpenAI Gym tasks or 10000 environment steps for DeepMind Control Suite tasks. We run each algorithm on an *NVIDIA RTX 3090* GPU and an *AMD Ryzen Threadripper 3970x* CPU. As described in Section 6 of the main text, our implementation groups the parameters of the different models in the critic's ensemble into a single network to achieve better scaling for inference and backpropagation with distributed hardware.

Training times. We report all average training times in Table 5. Comparing the training times with and without dual TD-learning and pessimism annealing clearly shows that both procedures introduce trivial computational overheads. Since the critic's updates represent the bulk of the computation in off-policy RL, modifying the associated hyper-parameters and parameterizations has relevant effects on training times. In particular, using a classical action-value model architecture speeds up training time by approximately 30%, while using a distributional critic slows down training by approximately 20%. Thanks to our implementation and the efficiency of large tensor operations, increasing the number of action-value models affects training times sub-linearly. For instance, doubling the critic's ensemble size to 20 results in less than 12% additional training overhead. On the other hand, increasing the UTD ratio increases the total training time almost linearly. As compared to the original *REDQ* implementation from Chen et al. (2021), *GPL-SAC* trains in less than half the time while having similar algorithmic complexity.

Considerations. The results from this Section can provide intuitions on selecting components and hyper-parameters when applying our implementation of *Generalized Pessimism Learning* for different problems. For instance, increasing ensemble size beyond $N = 10$ appears to affect training times only marginally. Thus, indicating that a viable criterion for selecting N in practical scenarios could be based on how many models fit into GPU memory. Multiple directions can be explored to further improve the time-efficiency and scalability of *GPL*. Since our uncertainty regularizer is compatible with variational representations of the critic's parameters posterior, Bayesian parameterizations could be a viable option to capture epistemic uncertainty more efficiently than model ensembles. Moreover, recent work by Björck et al. (2021) shows that it is possible to train modern reinforcement learning algorithms in half precision, with non-trivial computation and memory benefits.