Dissecting Deep RL with High Update Ratios: Combatting Value Divergence

Marcel Hussing†
University of Pennsylvania
mhussing@seas.upenn.edu

Claas Voelcker†
University of Toronto
Vector Institute, Toronto
cvoelcker@cs.toronto.edu

Igor Gilitschenski
University of Toronto
Vector Institute, Toronto

Amir-massoud Farahmand
University of Toronto

Eric Eaton
University of Pennsylvania

Abstract

We show that deep reinforcement learning algorithms can retain their ability to learn without resetting network parameters in settings where the number of gradient updates greatly exceeds the number of environment samples by combatting value function divergence. Under large update-to-data ratios, a recent study by Nikishin et al. (2022) suggested the emergence of a *primacy bias*, in which agents overfit early interactions and downplay later experience, impairing their ability to learn. In this work, we investigate the phenomena leading to the primacy bias. We inspect the early stages of training that were conjectured to cause the failure to learn and find that one fundamental challenge is a long-standing acquaintance: value function divergence. Overinflated Q-values are found not only on out-of-distribution but also in-distribution data and can be linked to overestimation on unseen action prediction propelled by optimizer momentum. We employ a simple unit-ball normalization that enables learning under large update ratios, show its efficacy on the widely used dm_control suite, and obtain strong performance on the challenging dog tasks, competitive with model-based approaches. Our results question, in parts, the prior explanation for sub-optimal learning due to overfitting early data.

1 Introduction

To improve sample efficiency, contemporary work in off-policy deep reinforcement learning (RL) has begun increasing the number of gradient updates per collected environment step (Janner et al., 2019; Fedus et al., 2020; Chen et al., 2021; Hiraoka et al., 2022; Nikishin et al., 2022; D'Oro et al., 2023; Schwarzer et al., 2023; Kim et al., 2023). As this update-to-data (UTD) ratio increases, various novel challenges arise. Notably, a recent study proposed the emergence of a *primacy bias* in deep actor critic algorithms, defined as "a tendency to overfit initial experiences that damages the rest of the learning process" (Nikishin et al., 2022). This is a fairly broad explanation of the phenomenon, leaving room for investigation into how fitting early experiences causes suboptimal learning behavior.

First approaches to tackle the learning failure challenges have been suggested, such as completely resetting networks periodically during the training process and then retraining them using the contents of the replay buffer (Nikishin et al., 2022; D'Oro et al., 2023). Resetting network parameters is a useful technique in that, in some sense, it can circumvent any previous optimization failures without prior specification. Yet it seems likely that a more nuanced treatment of the various optimization challenges in deep RL might lead to more efficient training down the line. Especially if the

† The two first authors contributed equally to this work.

objective is efficiency, throwing away all learned parameters and starting from scratch periodically is counter-productive, for instance in scenarios where, keeping all previous experience is infeasible. As such, we set out to study the components of early training that impair learning more closely and examine whether high-UTD learning without resetting is possible.

To motivate our study, we repeat the priming experiment of Nikishin et al. (2022), in which a network is updated for a large number of gradient steps on limited data. We show that during priming stages of training, value estimates diverge so far—and become so extreme—that it takes very long to unlearn them using new, counter-factual data. However, contrary to prior work, we find that it is not *impossible* to learn even after priming, it merely takes a long time and many samples. This sparks hope for our endeavor of smooth learning in high-UTD regimes. We show that compensating for the value function divergence allows learning to proceed. This suggests that the failure to learn does not stem from overfitting early data, which would result in correct value function on seen data, but rather from improperly fitting Q-values. We demonstrate that this divergence is most likely caused by prediction of out-of-distribution (OOD) actions that trigger large gradient updates, compounded by the momentum terms in the Adam optimizer (Kingma & Ba, 2015).

The identified behavior, although triggered by OOD action prediction, seems to be more than the well-known overestimation due to statistical bias (Thrun & Schwartz, 1993). Instead, we hypothesize that the problem is an optimization failure and focus on limiting the exploding gradients from the optimizer via architectural changes. The main evidence for this hypothesis is that standard RL approaches to mitigating bias, such as minimization over two independent critic estimates (Fujimoto et al., 2018), are insufficient. In addition, using pessimistic updates (Fujimoto et al., 2019; Fujimoto & Gu, 2021) or regularization (Krogh & Hertz, 1991; Srivastava et al., 2014) to treat the value divergence can potentially lead to suboptimal learning behavior, which is why architectural improvements are preferable in many cases.

We use a simple feature normalization method (Zhang & Sennrich, 2019; Wang et al., 2020; Bjorck et al., 2022) that projects features onto the unit sphere. This decouples learning the scale of the values from the first layers of the network and moves it to the last linear layer. Empirically, this approach fully mitigates diverging Q-values in the priming experiment. Even after a large amount of priming steps, the agent immediately starts to learn. In a set of experiments on the dm_control MuJoCo benchmarks (Tunyasuvunakool et al., 2020), we show that accounting for value divergence can achieve significant across-task performance improvements when using high update ratios. Moreover, we achieve non-trivial performance on the challenging dog tasks that are often only tackled using model-based approaches. We demonstrate comparable performance with the recently developed TD-MPC2 (Hansen et al., 2024), without using models or advanced policy search methods. Lastly, we isolate more independent failure modes, giving pointers towards their origins. In Appendix E we list open problems whose solutions might illuminate other RL optimization issues.

2 Preliminaries

Reinforcement learning We phrase the RL problem (Sutton & Barto, 2018) via the common framework of solving a discounted Markov decision process (MDP) (Puterman, 1994) $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, P, r, \gamma\}$. Here, $\mathcal{S}$ denotes the state space, $\mathcal{A}$ the action space, $P(s'|s,a)$ the transition probabilities when executing action a in state s, $r(s,a)$ the reward function, and γ the discount factor. A policy π encodes a behavioral plan in an MDP via a mapping from states to a distribution over actions $\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$. The goal is to find an optimal policy π^* that maximizes the sum of discounted return $J_t = \sum_{k=t+1}^{\infty} \gamma^{k-t-1} r_k(s,a)$. The value function $V_\pi(s) = \mathbb{E}_{\pi,P}[J_t \mid s_t = s]$ and the Q-value function $Q_\pi(s,a) = \mathbb{E}_{\pi,P}[J_t \mid s_t = s, a_t = a]$ define the expected, discounted cumulative return given that an agent starts in state s_t or starts in state s_t with action a_t respectively.

Deep actor-critic methods We focus on the setting of deep RL with off-policy actor-critic frameworks for continuous control (Lillicrap et al., 2016; Fujimoto et al., 2018; Haarnoja et al., 2018). Our analysis uses the soft-actor critic (SAC) algorithm (Haarnoja et al., 2018), but our findings extend to other methods such as TD3 (Fujimoto et al., 2018). Commonly used off-policy actor

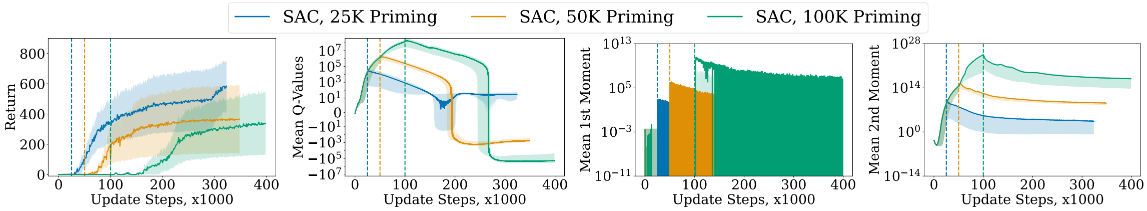

Figure 1: Return, in-distribution Q-valueslmao and Adam optimizer moments during priming for different lengths. Dotted lines correspond to end of priming. More priming leads to lower return and larger Q-value and optimizer divergence.

critic algorithms like SAC have four main components: a policy $\pi_\psi(a|s)$, a critic network $Q_\theta(s,a)$, a delayed target network $\bar{Q}_{\bar{\theta}}(s,a)$, and a replay buffer $\mathcal{D} = \{s_i, a_i, r_i, s_{i+1}\}_{i=1}^N$ that stores past interaction data. All functions are parameterized as neural networks (by ψ, θ, and $\bar{\theta}$, respectively) and, except for the target network, updated via gradient descent. The target network is updated using Polyak averaging (Polyak & Juditsky, 1992) at every time-step, formulated as $\bar{\theta} \leftarrow (1-\tau)\bar{\theta} + \tau\theta$, where τ modulates the update amount. Actor and critic are updated using the objectives

$$\max_{\psi} \mathbb{E}_{\substack{s\sim\mathcal{D} \\ a\sim\pi_\psi(\cdot|s_i)}} \left[\min_{j\in\{1,2\}} Q_{\theta_j}(s,a) \right], \tag{1}$$

$$\min_{\theta} \left(Q_\theta(s,a) - \left(r + \gamma \mathbb{E}_{a'\sim\pi_\psi(\cdot|s_{i+1})} \left[\min_{j\in\{1,2\}} \bar{Q}_{\bar{\theta}_j}(s',a') \right] \right) \right)^2, \tag{2}$$

respectively. In SAC, the update rules additionally contain a regularization term that maximizes the entropy of the actor $H(\pi_\psi(\cdot)|s)$. The differentiability of the expectation in Equation (1) is ensured by choosing the policy from a reparameterizable class of density functions (Haarnoja et al., 2018). We assume that all Q-functions consist of a multi-layer perceptron (MLP) encoder ϕ and a linear mapping w such that $Q_\theta(s,a) = \phi(s,a)w$, where we omit parametrization of the encoder for brevity.

3 Investigating the effects of large update-to-data ratios during priming

As mentioned, the definition of the primacy bias is broad. To obtain a more nuanced understanding, we set out to re-investigate the early stages of high-UTD training. To do so, we repeat the priming experiment conducted by Nikishin et al. (2022).[1] We first collect a small amount of random samples. Then, using the SAC algorithm, we perform a priming step, training the agent for a large number of updates without additional data. After priming, training continues as usual. Prior results reported by Nikishin et al. (2022) suggest that once the priming step has happened, agents lose their ability to learn completely. We use the simple Finger-spin task (Tunyasuvunakool et al., 2020) to study the root causes for this systematic failure and to examine if there are ways to recover without resets. In this section, we report means over five random seeds with standard error in shaded regions. Hyperparameters are kept consistent with previous work for ease of comparison.

3.1 An old acquaintance: Q-value overestimation

We first ask whether there is a barrier as to how many steps an agent can be primed for before it becomes unable to learn. We test this by collecting 1,000 samples and varying the number of updates during priming from 25,000 to 50,000 and 100,000. The results are presented in Figure 1.

We make two key observations. First, lower amounts of priming are correlated with higher early performance. More precisely, it seems that many runs simply take longer before they start learning as the number of priming steps increases. Second, during priming the scale of the average Q-value estimates on observed state-action pairs increases drastically. We find that the Q-values start out at

[1] For consistency with later sections, we use the ReLU activation here which can lead to unstable learning of other components. We repeat all the experiments with ELUs in Appendix B to provide even stronger support of our findings.

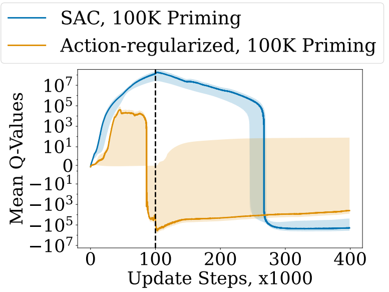

Figure 2: Priming with SAC and action regularization during priming. The latter lowers divergence.

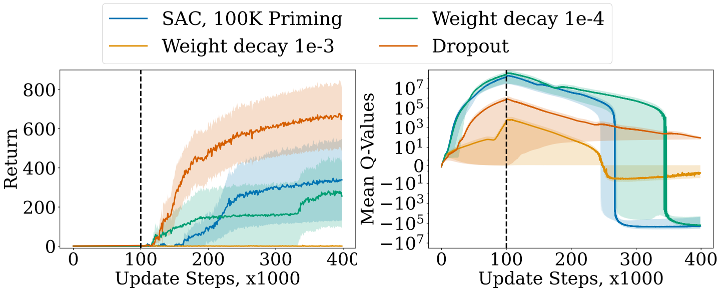

Figure 3: Return and Q-values of priming runs with weight decay and dropout. Results indicate that both regularizations mitigate priming to some extent but not sufficiently.

a reasonable level, but as priming goes on they eventually start to diverge drastically. Once the agent estimates very large Q-values, the final performance in terms of average returns deteriorates. We also observe that the second moment of the Adam optimizer (Kingma & Ba, 2015) is correlated with the divergence effect. Optimizer divergence has been observed before as a cause of plasticity loss under non-stationarity (Lyle et al., 2023), but in our experiments the data is stationary during priming. We conjecture that the momentum terms lead to much quicker propagation of poor Q-values and ultimately to prediction of incorrect Q-values, even on in-distribution data.

After priming, there exist two cases: 1) either the Q-values need to be unlearned before the agent can make progress or 2) there is a large drop from very high to very low Q-values that is strongly correlated with loss in effective dimension of the embedding, as defined by Yang et al. (2020) (see Appendix B.3). In the second case, rank can sometimes be recovered upon seeing new, counter-factual data and the network continues to learn. Yet, sometimes the agent gets stuck at low effective dimension; a possible explanation for the failure to learn observed in the priming experiments of Nikishin et al. (2022). This is orthogonal to a previously studied phenomenon where target network-based updates lead to perpetually reduced effective rank (Kumar et al., 2021).

3.2 On the potential causes of divergence

We conjecture that Q-value divergence starts with overestimated values of OOD actions. This overestimation could cause the optimizer to continually increase Q-values via its momentum leading to divergence. To test this hypothesis, we add a conservative behavioral cloning (Pomerleau, 1988; Atkeson & Schaal, 1997) loss term to our actor that forces the policy to be close to replay buffer actions. Prior work employed this technique in offline RL to mitigate value overestimation (Fujimoto & Gu, 2021). More formally, our actor update is extended by the loss $\mathcal{L}_{c,\psi} = \min_{\psi} \mathbb{E}_{a\sim\mathcal{D},\hat{a}\sim\pi_{\psi}(s)}[||a - \hat{a}||_2]$. The results in Figure 2 indicate that the basis of the conjecture is corroborated as divergence is much smaller—but not mitigated completely—when values are learned on actions similar to seen ones. However, in practice we do not know when divergence sets in, which limits the applicability of this technique in realistic scenarios. Using it throughout all of training, rather than just during priming, impairs the learner's ability to explore. We investigate the effects of the optimizer in more detail and provide preliminary evidence that the second-order term may be at fault in Appendix B.2.

3.3 Applying common regularization techniques

Regularization is a common way to mitigate gradient explosion and is often used to address overestimation (Farebrother et al., 2018; Chen et al., 2021; Liu et al., 2021; Hiraoka et al., 2022; Li et al., 2023). We investigate the priming experiments under techniques such as using L^2 weight decay (Krogh & Hertz, 1991) or adding dropout (Srivastava et al., 2014) to our networks in Figure 3.

Both L^2 weight decay as well as dropout can somewhat reduce the divergence during priming, however not to a sufficient degree. While L^2 regularization fails to attain very high performance,

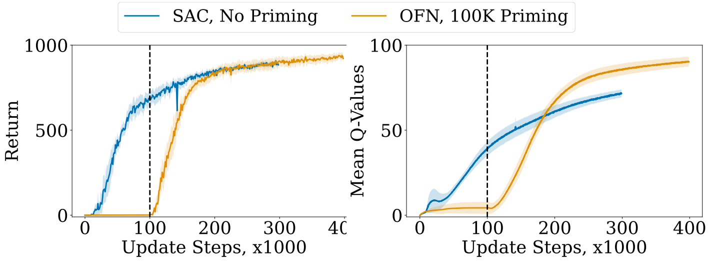

Figure 5: (Left) Return and (Right) Q-values comparing SGD result and OFN when priming for 100K steps. OFN obtains returns close to that of the well-trained SGD agent and learns an appropriate Q-value scale correctly.

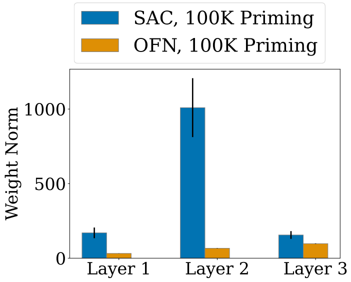

Figure 6: L_2 norm of network weights per layer after priming for default and OFN architectures. OFN leads to smaller weights and significant mass in the last layer.

dropout is able to recover a good amount of final return. However, both methods require tuning of a hyperparameter that trades off the regularization term with the main loss. This hyperparameter is environment-dependent and tuning it becomes infeasible for large UTD-ratios due to computational resource limitations. Still, the results imply that it is in fact possible to overcome the divergence in priming and continue to learn good policies.

3.4 Divergence in practice

One question that remains is whether we can find these divergence effects outside of the priming setup. We find that, while priming is an artificially constructed worst case, similar phenomena happen in regular training on standard benchmarks when increasing update ratios (see Figure 4). Further, the divergence is not limited to early stages of training as it happens at arbitrary points in time. We therefore conjecture that divergence is not a function of amount of experience but rather one of state-action space coverage. Note that the reported Q-values have been measured on the observed training data, not on any out-of-distribution state-action pairs. The respective critic losses become very large. All this points toward a failure to fit Q-values. This behavior does not align with our common understanding of overfitting (Bishop, 2006), challenging the hypothesis that high-UTD learning fails merely due to large validation error (Li et al., 2023).

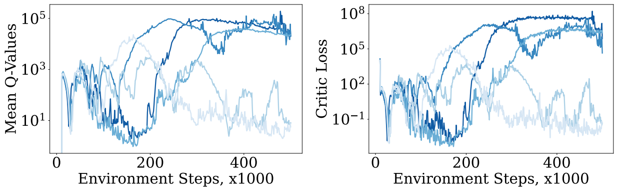

Figure 4: In-distribution Q-values and critic loss of five SAC seeds on the humanoid-run task using UTD $= 32$. Values diverge at arbitrary time-points, not only during the beginning. Loss mirrors Q-value divergence.

4 Towards high-UTD optimization without resetting

Regularization techniques such as those in Section 3.3 can fail to alleviate divergence as they tend to operate across the whole network and lower the weights everywhere even if higher values are actually indicated by the data. They also require costly hyperparameter tuning. Thus, we turn towards network architecture changes to the commonly used MLPs that have proven useful in overcoming issues such as exploding gradients in other domains (Ba et al., 2016; Xu et al., 2019).

4.1 Limiting gradient explosion via unit ball normalization

As discussed previously, the prediction of an unknown action might trigger the propagation of a large, harmful gradient. Further, the Q-values of our network ought to grow over time as they more closely approximate those of a good policy. If we predict incorrectly on one of these Q-values, a potentially very large loss is propagated. Gradients are magnified by multiplicative backpropagation via ReLU activations (Glorot et al., 2011) as well as momentum from Adam (Kingma & Ba, 2015). Note that all resulting issues arise in the early network layers. We hypothesize that we can address many of these problems by separating the scaling of the Q-values to the appropriate size from the earlier non-linear layers of the network and moving the Q-value scaling to the final linear layer.

One contender to achieve the value decoupling described in the previous paragraph is layer normalization (Ba et al., 2016), but one would have to disable scaling factors used in common implementations. Still, standard layer normalization would not guarantee small features everywhere. Instead, we use a stronger constraint and project the output features of the critic encoder onto the unit ball using the function $f(\mathbf{x}) = \frac{\mathbf{x}}{\|\mathbf{x}\|_2}$ (Zhang & Sennrich, 2019), where $\|\cdot\|_2$ denotes the L^2 vector norm and $\mathbf{x}$ is the output of our encoder $\phi(s, a)$. This ensures that all values are strictly between 0 and 1 and the gradients will be tangent to the unit sphere. Note that this function's gradient is not necessarily bounded to ensure low gradient propagation (see Appendix D), but we argue that if large values are never created in the early layers, gradient explosion will not occur. The unit ball has previously been used to mitigate large action prediction in the actor (Wang et al., 2020) or to stabilize RL training in general (Bjorck et al., 2022). For brevity, we will refer to this method as *output feature normalization* (OFN). We solely apply OFN to the critic, unlike Wang et al. (2020), since our goal is to mitigate value divergence. OFN is very simple and requires only a one-line change in implementation.

4.2 Evaluating feature output normalization during priming

To test the efficacy of the OFN-based approach, we repeat the priming experiment in Figure 5. We find that OFN achieves high reward and most distinctly, Q-value divergence during priming is fully mitigated. Note also that we are using a discount factor of $\gamma = 0.99$, returns are collected over 1,000 timesteps and rewards are in $[0, 1]$. We therefore expect the average Q-values to be roughly at 10% of the undiscounted return which seems correct for the OFN network. However, more importantly, as shown in Figure 6, most of the Q-value scaling happens in the last layer.

5 Experimental evaluation

We evaluate our findings on the commonly used dm_control suite (Tunyasuvunakool et al., 2020). All results are averaged over ten random seeds.[2] We report evaluation returns similar to Nikishin et al. (2022), which we record every 10,000 environment steps. We compare a standard two-layer MLP with ReLU (Nair & Hinton, 2010) activations, both with and without resetting, to the same MLP with OFN. The architecture is standard in many reference implementations. Architecture and the resetting protocol are taken from D'Oro et al. (2023) and hyperparameters are kept without new tuning to ensure comparability of the results. More details can be found in Appendix A.

To understand the efficacy of output normalization on real environments under high UTD ratios, we set out to answer multiple questions that will illuminate RL optimization failures:
Q1: Can we maintain learning without resetting neural networks?
Q2: Are there other failure modes beside Q-value divergence under high UTD ratios?
Q3: When resets alone fall short, can architectural changes enable better high-UTD training?

5.1 Feature normalization stabilizes high-UTD training

To answer **Q1**, we compare OFN and SAC with resets on the DMC15-500k benchmark with large update ratios of 8 and 32 as proposed by Nikishin et al. (2022) and Schwarzer et al. (2023). We report mean, interquartile mean (IQM) and median as well as 95% bootstrapped confidence intervals aggregated over seeds and tasks, following Agarwal et al. (2021). The results are shown in Figure 7.

[2] For comparison with TD-MPC2 (Hansen et al., 2024) we use data provided by their implementation, which only contains three seeds. As the goal is not to rank algorithmic performance but to give intuition about the relative strengths of adapting the network architecture, we believe that this is sufficient in this case.

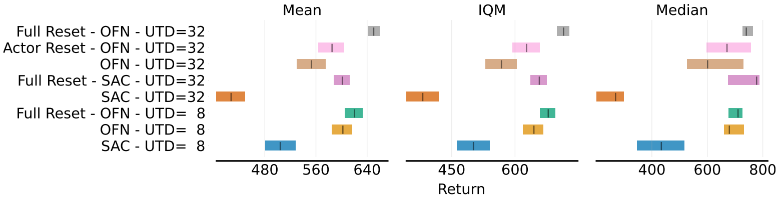

Figure 7: Mean, interquartile mean (IQM), and median with 95% bootstrapped confidence intervals of standard SAC and OFN on the DMC15-500k Suite. OFN can maintain high performance even under large UTD. OFN with UTD = 8 achieves comparable performance to standard resetting with UTD = 32 across metrics.

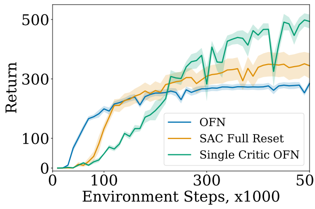

Figure 8: Mean return of single-critic OFN, standard OFN and resetting; UTD = 32 on hopper-hop. Shaded regions are standard error.

First, we observe that in both cases, UTD = 8 and UTD = 32, OFN can significantly improve over the non-resetting MLP baseline across all metrics. The value estimates that diverge seem to have been handeled properly (see Appendix C.2); learning is maintained. We note that our approach with UTD = 8 achieves mean and IQM performance comparable to that of standard resetting with UTD = 32. In median and quantile performance, all UTD = 32 overlap, highlighting that outliers contribute to the performance measurement. Note that the overall performance drops slightly for the OFN-based approach when going from UTD = 8 to UTD = 32. We conjecture that there are other learning problems such as exploration that have not been treated by alleviating value divergence. However, these do not lead to complete failure to learn but rather slightly slower convergence.

5.2 Other failure modes: Exploration limitations

To validate the hypothesis of other failures and answer **Q2**, we run two additional experiments. First, our current focus is on failures of the critic; our proposed mitigation does not address any further failures that might stem from the actor. We defer a more detailed analysis of actor failure cases to future work. Instead, we test the OFN-based architecture again and, for now, simply reset the actor to shed light on the existence of potential additional challenges. For comparison, we also run a second experiment in which we reset all learned parameters, including the critic.

The results in Figure 7 indicate that actor resetting can account for a meaningful portion of OFN's performance decline when going from UTD = 8 to UTD = 32. The actor-reset results are within variance of the full-resetting standard MLP baseline. Further, we observe that there is still some additional benefit to resetting the critic as well. This does not invalidate the premise of our hypothesis, value divergence might not be the *only* cause of problems in the high UTD case. We have provided significant evidence that it is a *major* contributor. Resetting both networks of OFN with UTD = 32 outperforms all other baselines on mean and IQM comparisons.

To explain the remaining efficacy of critic resets, we examine the hopper-hop environment where standard SAC with resets outperforms OFN. In RL with function approximation, one might not only encounter over- but also under-estimation (Wu et al., 2020; Lan et al., 2020; Saglam et al., 2021). We believe that hopper is sensitive to pessimism, and periodically resetting the networks might partially and temporarily counteract the inherent pessimism of the dual critic setup.

To obtain evidence for this conjecture, we repeated some experiments with a single critic. As OFN handles divergence it might not require a minimization over two critics (Fujimoto et al., 2018). We compare OFN using a single critic and 32 updates per environment step to standard SAC and OFN in Figure 8. With a single critic, OFN does not get stuck in a local minimum and outperforms full resetting. Note that this is only true in few environments, leading us to believe that the effects of high-update training are MDP-dependent. In some environments we observe unstable learning with a single critic, which highlights that the bias countered by the double critic optimization and the overestimation from optimization we study are likely orthogonal phenomena that both need to be

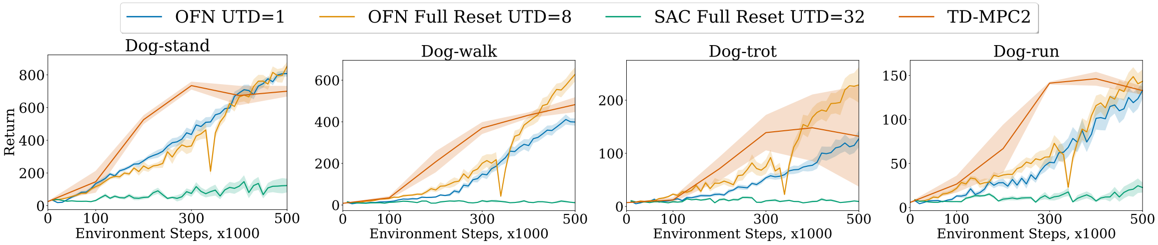

Figure 9: Mean return on the dog DMC tasks, comparing OFN to SAC with resets and the model-based TD-MPC2. Shaded regions indicate standard error. OFN outperforms SAC with resets, which is unable to learn and OFN with UTD = 8 and resetting is competitive with TD-MPC2.

addressed. Most likely, there is a difficult trade-off between optimization stability and encouraging sufficient exploration, which is an exciting avenue for future research.

5.3 Limit-testing feature normalization

To answer **Q3**, we move to a set of training environments that is considered exceptionally hard for model-free approaches, namely the dog tasks of the DMC suite. Standard SAC can generally not obtain any reasonable reward and, due to their complex dynamics, these tasks are often tackled using model-based approaches such as TD-MPC2 (Hansen et al., 2024) with complicated update procedures and carefully tuned network architectures. We evaluate SAC and OFN on the dog tasks and compare against TD-MPC2 in Figure 9.

First, observe that resetting without OFN obtains no improvement over a random policy. However, OFN with UTD = 1 can already obtain very good performance across all tasks, indicating that a major problem for SAC in these high-dimensional tasks is value divergence. When increasing the update ratio to 8 and adding resetting, we improve the performance of the OFN agent even further and can match the reported results of the strong model-based TD-MPC2 baseline.

We have already seen that resetting can take care of multiple optimization failures. However, these experiments also indicate that resetting is not a panacea as it is only effective when the initially learned policy can obtain some reward before being overwritten. This seems intuitive since resetting to a policy that cannot gather any useful data should not help. These results highlight that the early training dynamics of RL are highly important when it comes to training on complex environments and fitting early data correctly and quickly is crucial for success.

This also opens up the question why resetting in the humanoid environments in the previous sections can yield success even though very little reward is observed. Besides greater divergence due to larger observation spaces in the dog MDPs, we suspect that this might be related to the complexity of exploration. The ability of a random policy to obtain non-trivial reward and information about the environment has been shown to be a crucial factor in explaining the success of DRL methods in discrete environments (Laidlaw et al., 2023), and similar phenomena might be in effect here.

6 Related work

Our work closely examines previous work on the primacy bias and the related resetting technique (Anderson, 1992; Nikishin et al., 2022; D'Oro et al., 2023; Schwarzer et al., 2023). Going beyond, overestimation and feature learning challenges are a widely studied phenomenon in the literature.

Combatting overestimation Overestimation in off-policy value function learning is a well-established problem in the RL literature that dates back far before the prime times of deep learning (Thrun & Schwartz, 1993; Precup et al., 2001). The effects of function approximation error and the effect on variance and bias have been studied (Pendrith & Ryan, 1997; Mannor et al., 2007) as well. With the rise of deep learning, researchers have tried to address the overestimation bias via algorithmic interventions such as combining multiple Q-learning predictors to achieve underestimation (Hasselt, 2010; Hasselt et al., 2016; Zhang et al., 2017; Lan et al., 2020), using averages over previous Q-values for variance reduction (Anschel et al., 2017), or general error term correction (Lee et al., 2013; Fox et al., 2016). In the context of actor-critic methods, the twinned critic minimization approach of Fujimoto et al. (2018) has become a de-facto standard. Most of these approaches are not applicable or break down under very high update ratios. To regulate the overestimation-pessimism balance more carefully, several authors have attempted to use larger ensembles of independent Q-value estimates (Lee et al., 2021; Peer et al., 2021; Chen et al., 2021; Hiraoka et al., 2022). Ensembling ideas were also combined with ideas from distributional RL (Bellemare et al., 2017) to combat overestimation (Kuznetsov et al., 2020). Instead of addressing the statistical bias in deep RL, our study focuses on the problems inherent to neural networks and gradient based optimization for value function estimation. Work from offline-to-online RL has demonstrated that standard layer normalization can bound value estimates during offline training and mitigate extrapolation while still allowing for exploration afterwards (Ball et al., 2023). Layer normalization has subsequently been used to achieve generally strong results in offline RL (Tarasov et al., 2023). Our work is also related to a recent contribution using batch-normalization for increased computational efficiency by Bhatt et al. (2024) who focus on decreasing update ratios rather than increasing them. A concurrent work by Nauman et al. (2024) provides a large scale study on different regularization techniques to combat overestimation. This work also demonstrates the efficacy of SAC on the dog tasks when properly regularized but it does not highlight the effects of Q-value divergence from exploding gradients as a key challenge for this set of environments.

Combating plasticity loss Another aspect of the primacy bias is the tendency of neural networks to lose their capacity for learning over time (Igl et al., 2021), sometimes termed *plasticity loss* (Lyle et al., 2021; Abbas et al., 2023). Recent work mitigates plasticity loss using feature rank maximization (Kumar et al., 2021), regularization (Lyle et al., 2023), or learning a second copied network (Nikishin et al., 2024). Some of the loss stems from neurons falling dormant over time (Sokar et al., 2023). A concurrent, closely related work by Lyle et al. (2024) disentangles the causes for plasticity loss further. They use layer normalization to prevent some of these causes, which is closely related to our unit ball normalization. Our work differs in that we focus on the setting of high update ratios and use stronger constraints to mitigate value divergence rather than plasticity loss.

7 Conclusion and future work

By dissecting the effects underlying the primacy bias, we have identified a crucial challenge: *value divergence*. While the main focus in studying increased Q-value has been on the statistical bias inherent in off-policy sampling, we show that Q-value divergence can arise due to problems inherent to neural network optimization. This optimization-caused divergence can be mitigated using the unit-ball normalization approach, which shines on the dm_control benchmark with its simplicity and efficacy. With this result, we challenge the assumption that failure to learn in high-UTD settings primarily stems from *overfitting* early data by showing that combating value divergence is competitive with resetting networks. This offers a starting point towards explaining the challenges of high-UTD training in more detail and opens the path towards even more performant and sample efficient RL in the future.

However, as our other experiments show, mitigating value overestimation through optimization is not the only problem that plagues high-UTD learning. To clearly highlight these possible directions for future work, we provide an extensive discussion of open problems in Appendix E. Additional problems, such as *exploration failures* or *suboptimal feature learning*, can still exist and need to be resolved to unlock the full potential of high-UTD RL.

Acknowledgements

EE and MH's research was partially supported by the Army Research Office under MURI award W911NF20-1-0080, the DARPA Triage Challenge under award HR001123S0011, and by the University of Pennsylvania ASSET center. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of DARPA, the Army, or the US government. AMF acknowledges the funding from the Canada CIFAR AI Chairs program, as well as the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) through the Discovery Grant program (2021-03701). Resources used by CV, AMF, and IG in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute.

The authors thank the members of the GRASP lab at UPenn, and the members of the AdAge and TISL labs at UofT, as well as the anonymous reviewers for their valuable feedback.

References

Zaheer Abbas, Rosie Zhao, Joseph Modayil, Adam White, and Marlos C. Machado. Loss of plasticity in continual deep reinforcement learning, 2023.

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021.

Charles Anderson. Q-learning with hidden-unit restarting. In S. Hanson, J. Cowan, and C. Giles (eds.), *Advances in Neural Information Processing Systems*, volume 5. Morgan-Kaufmann, 1992.

Oron Anschel, Nir Baram, and Nahum Shimkin. Averaged-DQN: Variance reduction and stabilization for deep reinforcement learning. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 176–185. PMLR, 06–11 Aug 2017.

Kavosh Asadi, Rasool Fakoor, and Shoham Sabach. Resetting the optimizer in deep RL: An empirical study. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Christopher G. Atkeson and Stefan Schaal. Robot learning from demonstration. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pp. 12–20, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc. ISBN 1558604863.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.

Philip J. Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 449–458. PMLR, 06–11 Aug 2017.

Aditya Bhatt, Daniel Palenicek, Boris Belousov, Max Argus, Artemij Amiranashvili, Thomas Brox, and Jan Peters. Cross*q*: Batch normalization in deep reinforcement learning for greater sample efficiency and simplicity. In *The Twelfth International Conference on Learning Representations*, 2024.

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

Johan Bjorck, Carla P Gomes, and Kilian Q Weinberger. Is high variance unavoidable in RL? a case study in continuous control. In *International Conference on Learning Representations*, 2022.

Xinyue Chen, Che Wang, Zijian Zhou, and Keith W. Ross. Randomized ensembled double q-learning: Learning fast without a model. In *International Conference on Learning Representations*, 2021.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In Yoshua Bengio and Yann LeCun (eds.), *International Conference on Learning Representations*, 2016.

William C Dabney. Adaptive step-sizes for reinforcement learning. 2014.

Pierluca D'Oro, Max Schwarzer, Evgenii Nikishin, Pierre-Luc Bacon, Marc G Bellemare, and Aaron Courville. Sample-efficient reinforcement learning by breaking the replay ratio barrier. In *The Eleventh International Conference on Learning Representations*, 2023.

Amir-massoud Farahmand and Mohammad Ghavamzadeh. Pid accelerated value iteration algorithm. In *International Conference on Machine Learning*. PMLR, 2021.

Jesse Farebrother, Marlos C. Machado, and Michael Bowling. Generalization and regularization in DQN. *CoRR*, abs/1810.00123, 2018.

William Fedus, Prajit Ramachandran, Rishabh Agarwal, Yoshua Bengio, Hugo Larochelle, Mark Rowland, and Will Dabney. Revisiting fundamentals of experience replay. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Roy Fox, Ari Pakman, and Naftali Tishby. Taming the noise in reinforcement learning via soft updates. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, pp. 202–211, Arlington, Virginia, USA, 2016. AUAI Press. ISBN 9780996643115.

Scott Fujimoto and Shixiang Gu. A minimalist approach to offline reinforcement learning. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1587–1596. PMLR, 10–15 Jul 2018.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pp. 2052–2062, 2019.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In Geoffrey Gordon, David Dunson, and Miroslav Dudík (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 315–323, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870. PMLR, 10–15 Jul 2018.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *The Twelfth International Conference on Learning Representations*, 2024.

Hado van Hasselt. Double q-learning. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta (eds.), *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pp. 2094–2100. AAAI Press, 2016.

Takuya Hiraoka, Takahisa Imagawa, Taisei Hashimoto, Takashi Onishi, and Yoshimasa Tsuruoka. Dropout q-functions for doubly efficient reinforcement learning. In *International Conference on Learning Representations*, 2022.

Maximilian Igl, Gregory Farquhar, Jelena Luketina, Wendelin Boehmer, and Shimon Whiteson. Transient non-stationarity and generalisation in deep reinforcement learning. In *International Conference on Learning Representations*, 2021.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, 2019.

Woojun Kim, Yongjae Shin, Jongeui Park, and Youngchul Sung. Sample-efficient and safe deep reinforcement learning via reset deep ensemble agents. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 53239–53260. Curran Associates, Inc., 2023.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA, 2015.

Anders Krogh and John Hertz. A simple weight decay can improve generalization. In J. Moody, S. Hanson, and R.P. Lippmann (eds.), *Advances in Neural Information Processing Systems*, volume 4. Morgan-Kaufmann, 1991.

Aviral Kumar, Rishabh Agarwal, Dibya Ghosh, and Sergey Levine. Implicit under-parameterization inhibits data-efficient deep reinforcement learning. In *International Conference on Learning Representations*, 2021.

Arsenii Kuznetsov, Pavel Shvechikov, Alexander Grishin, and Dmitry Vetrov. Controlling overestimation bias with truncated mixture of continuous distributional quantile critics. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5556–5566. PMLR, 13–18 Jul 2020.

Cassidy Laidlaw, Stuart Russell, and Anca Dragan. Bridging RL theory and practice with the effective horizon. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Qingfeng Lan, Yangchen Pan, Alona Fyshe, and Martha White. Maxmin q-learning: Controlling the estimation bias of q-learning. In *International Conference on Learning Representations*, 2020.

Donghun Lee, Boris Defourny, and Warren Buckler Powell. Bias-corrected q-learning to control max-operator bias in q-learning. In *Proceedings of the 2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning, ADPRL 2013 - 2013 IEEE Symposium Series on Computational Intelligence, SSCI 2013*, IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning, ADPRL, pp. 93–99, 2013. ISBN 9781467359252. doi: 10.1109/ADPRL.2013.6614994.

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*. PMLR, 2021.

Qiyang Li, Aviral Kumar, Ilya Kostrikov, and Sergey Levine. Efficient deep reinforcement learning requires regulating overfitting. In *The Eleventh International Conference on Learning Representations*, 2023.

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. In Yoshua Bengio and Yann LeCun (eds.), *International Conference on Learning Representations*, 2016.

Zhuang Liu, Xuanlin Li, Bingyi Kang, and Trevor Darrell. Regularization matters in policy optimization - an empirical study on continuous control. In *International Conference on Learning Representations*, 2021.

Clare Lyle, Mark Rowland, and Will Dabney. Understanding and preventing capacity loss in reinforcement learning. In *International Conference on Learning Representations*, 2021.

Clare Lyle, Zeyu Zheng, Evgenii Nikishin, Bernardo Avila Pires, Razvan Pascanu, and Will Dabney. Understanding plasticity in neural networks. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 23190–23211. PMLR, 23–29 Jul 2023.

Clare Lyle, Zeyu Zheng, Khimya Khetarpal, Hado van Hasselt, Razvan Pascanu, James Martens, and Will Dabney. Disentangling the causes of plasticity loss in neural networks, 2024.

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

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. In *NeurIPS Deep Learning Workshop*. 2013.

Ted Moskovitz, Jack Parker-Holder, Aldo Pacchiano, Michael Arbel, and Michael Jordan. Tactical optimism and pessimism for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 2021.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML 2010*, pp. 807–814, 2010.

Michal Nauman, Michał Bortkiewicz, Piotr Miłoś, Tomasz Trzcinski, Mateusz Ostaszewski, and Marek Cygan. Overestimation, overfitting, and plasticity in actor-critic: the bitter lesson of reinforcement learning. In *Forty-first International Conference on Machine Learning*, 2024.

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron Courville. The primacy bias in deep reinforcement learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16828–16847. PMLR, 17–23 Jul 2022.

Evgenii Nikishin, Junhyuk Oh, Georg Ostrovski, Clare Lyle, Razvan Pascanu, Will Dabney, and André Barreto. Deep reinforcement learning with plasticity injection. *Advances in Neural Information Processing Systems*, 36, 2024.

Oren Peer, Chen Tessler, Nadav Merlis, and Ron Meir. Ensemble bootstrapping for q-learning. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8454–8463. PMLR, 18–24 Jul 2021.

Mark Pendrith and Malcolm Ryan. Estimator variance in reinforcement learning: Theoretical problems and practical solutions. 1997.

B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992. doi: 10.1137/0330046.

Dean A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In D. Touretzky (ed.), *Advances in Neural Information Processing Systems*, volume 1. Morgan-Kaufmann, 1988.

Doina Precup, Richard S. Sutton, and Sanjoy Dasgupta. Off-policy temporal difference learning with function approximation. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pp. 417–424, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558607781.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., USA, 1st edition, 1994. ISBN 0471619779.

Herbert Robbins and Sutton Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400 – 407, 1951. doi: 10.1214/aoms/1177729586.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning internal representations by error propagation*, pp. 318–362. MIT Press, Cambridge, MA, USA, 1986. ISBN 026268053X.

Baturay Saglam, Enes Duran, Dogan C. Cicek, Furkan B. Mutlu, and Suleyman S. Kozat. Estimation error correction in deep reinforcement learning for deterministic actor-critic methods. In *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 137–144, 2021. doi: 10.1109/ICTAI52525.2021.00027.

Tom Schaul, Andre Barreto, John Quan, and Georg Ostrovski. The phenomenon of policy churn. In *Advances in Neural Information Processing Systems*, 2022.

Max Schwarzer, Johan Samir Obando Ceron, Aaron Courville, Marc G Bellemare, Rishabh Agarwal, and Pablo Samuel Castro. Bigger, better, faster: Human-level Atari with human-level efficiency. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 30365–30380. PMLR, 23–29 Jul 2023.

Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. The dormant neuron phenomenon in deep reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In Sanjoy Dasgupta and David McAllester (eds.), *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 1139–1147, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

Denis Tarasov, Vladislav Kurenkov, Alexander Nikulin, and Sergey Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 11592–11620. Curran Associates, Inc., 2023.

Sebastian Thrun and Anton Schwartz. Issues in using function approximation for reinforcement learning. In Michael Mozer, Paul Smolensky, David Touretzky, Jeffrey Elman, and Andreas Weigend (eds.), *Proceedings of the 1993 Connectionist Models Summer School*, pp. 255–263. Lawrence Erlbaum, 1993.

Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, Nicolas Heess, and Yuval Tassa. dm_control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020. ISSN 2665-9638. doi: https://doi.org/10.1016/j.simpa.2020.100022.

Nino Vieillard, Bruno Scherrer, Olivier Pietquin, and Matthieu Geist. Momentum in reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, 2020.

Che Wang, Yanqiu Wu, Quan Vuong, and Keith Ross. Striving for simplicity and performance in off-policy DRL: Output normalization and non-uniform sampling. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10070–10080. PMLR, 13–18 Jul 2020.

Dongming Wu, Xingping Dong, Jianbing Shen, and Steven C. H. Hoi. Reducing estimation bias via triplet-average deep deterministic policy gradient. *IEEE Transactions on Neural Networks and Learning Systems*, 31(11):4933–4945, 2020. doi: 10.1109/TNNLS.2019.2959129.

Jingjing Xu, Xu Sun, Zhiyuan Zhang, Guangxiang Zhao, and Junyang Lin. Understanding and improving layer normalization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Yuzhe Yang, Guo Zhang, Zhi Xu, and Dina Katabi. Harnessing structures for value-based planning and reinforcement learning. In *International Conference on Learning Representations*, 2020.

Biao Zhang and Rico Sennrich. Root Mean Square Layer Normalization. In *Advances in Neural Information Processing Systems 32*, Vancouver, Canada, 2019.

Zongzhang Zhang, Zhiyuan Pan, and Mykel J. Kochenderfer. Weighted double q-learning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 3455–3461, 2017. doi: 10.24963/ijcai.2017/483.

A Implementation details and hyperparameters

We employ two commonly used implementations, one for fast iterations on priming experiments (https://github.com/denisyarats/pytorch_sac) and one for scaling up our experiments to high update ratios (https://github.com/proceduralia/high_replay_ratio_continuous_control). All experiments in the main sections use default hyperparameters of the high update ratio codebase unless otherwise specified with minor exceptions.

Table 1: Shared hyperparameters between priming and high-UTD implementations

Optimizer	Adam
Adam β_1	0.9
Adam β_2	0.999
Adam ε	$1e-8$
Actor Learning Rate	$4e-3$
Critic Learning Rate	$4e-3$
Temp. Learning Rate	$3e-4$
Batch Size	256
γ	0.99
τ	0.005
# critics	2
# critic layers	2
# actor layers	2
critic hidden dim	256
actor hidden dim	256

Table 2: Differing hyperparameters between priming and high-UTD implementations

	Priming	High UTD
Initial temperature	0.1	1.0
Target entropy	-action_dim	-action_dim / 2
actor log std bounds	[-5, 2]	[-10, 2]

B Additional priming experiments

B.1 Activation functions

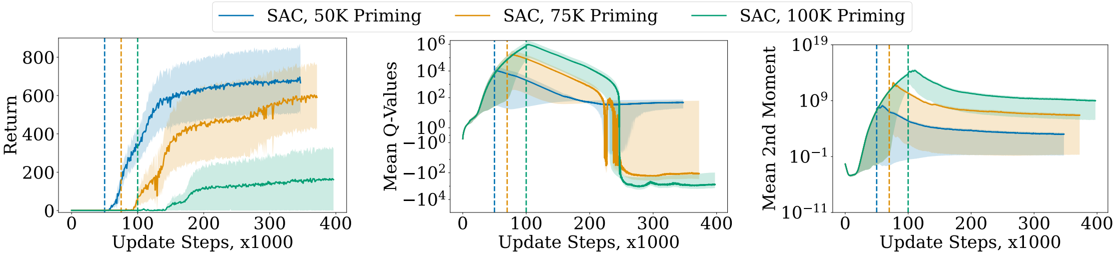

Figure 10: ELU activations. Return, in-distribution Q-values and Adam optimizer moments during priming for different lengths. Dotted lines correspond to end of priming. More priming leads to lower return and larger Q-value and optimizer divergence.

During our experiments, we found that the ReLU activation can sometimes lead to destabilization of other parts of the SAC agent during priming. We found that using ELU (Clevert et al., 2016) activations instead remedies some of these issues. We repeat various experiments from Section 3 again but with more stable activations. First, we show in Figure 10 that divergence happens similar to before and that it is correlated with the amount of priming.

Furthermore, we discussed that the divergence is most likely triggered by out of distribution action prediction (see Figure 11) and that regularization can help. When using ELUs, the effect of regularization is much more stable and as expected but still not as good as our OFN approach from

Section 4.2 (compare with Figure 12). Dropout leads to significantly worse performance and L^2 regularization learns Q-values too small for the obtained return which we suspect correlates with decreased exploration.

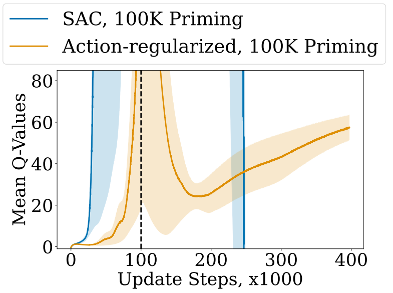

Figure 11: ELU activations. Priming with SAC and action regularization during priming. The latter lowers divergence.

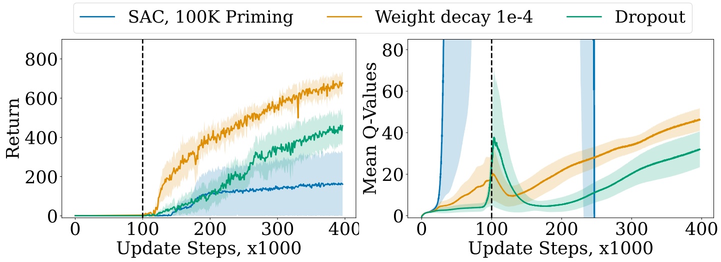

Figure 12: ELU activations. Return and Q-values of priming runs with weight decay and dropout. Results indicate that both regularizations mitigate priming more than with ReLUs.

B.2 Optimizer divergence

With more stable effects from the ELU activation, we introduce a second intervention to the priming stage. We hypothesize that most of the divergence stems from the second optimizer term that will propell the gradients to increase more and more over time. To test this, we run an additional experiment in which we use standard stochastic gradient descent (SGD) (Robbins & Monro, 1951) with first-order momentum (Rumelhart et al., 1986; Sutskever et al., 2013) during priming to isolate the effect of the second-order momentum term. We compare this against RMSProp which is equivalent to Adam but without the first optimizer term instead. The results are shown in Figure B.2. As we can see, the divergence is almost completely gone when using SGD with momentum but is even larger in RMSProp. Note that running the same experiment with ReLU activations leads to divergence in the actor when using SGD only. We suspect this might have to do with divergence in the actor entropy.

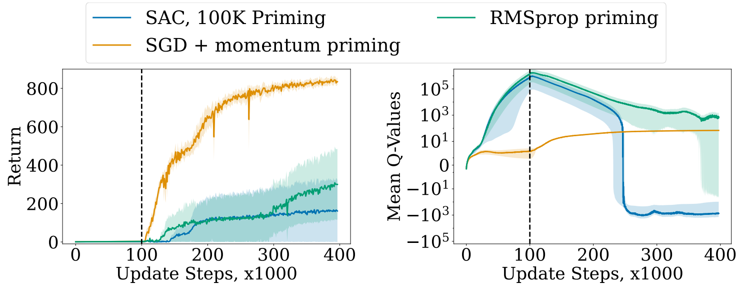

Figure 13: Comparing standard SAC priming to priming when using either SGD+momentum or RMSProp during the priming updates. SGD+momentum does not diverge with ELU activations, indicating that the second-order momentum term is the problematic one.

B.3 Effective dimension

Let $\Phi \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times d}$ be a feature matrix (in our case produced by ϕ). The effective dimension of a feature matrix has previously been defined as

$$\text{srank}_\delta = \min \left\{ k : \frac{\sum_{i=1}^{k} \sigma_i(\Phi)}{\sum_{i=1}^{d} \sigma_i(\Phi)} \geq 1 - \delta \right\} \quad ,$$

where δ is a threshold parameters and $\{\sigma_i(\Phi)\}$ are the singular values of Φ in decreasing order (Yang et al., 2020; Kumar et al., 2021).

An additional finding of ours is that divergence of Q-values is correlated with this effective rank srank_δ. We plot three different random seeds that have been subjected to 75,000 steps of priming in Figure 14; the effective rank is approximated over a sample 10 times the size of the embedding dimension. We observe, that divergence correlates with a decrease in effective dimension and that when divergence is exceptionally strong, the effective dimension drops so low that the agent has trouble to continue learning. This might explain the failure to learn observed by Nikishin et al. (2022). However, as long as the effective dimension does not drop too far, the agent can recover and regain capacity by observing new data. Previous work on effective rank loss has often assumed that it is mostly irreversible, yet we find that this is not always the case. We suspect that in complete failure cases, the policy has collapse and rarely any new data is seen.

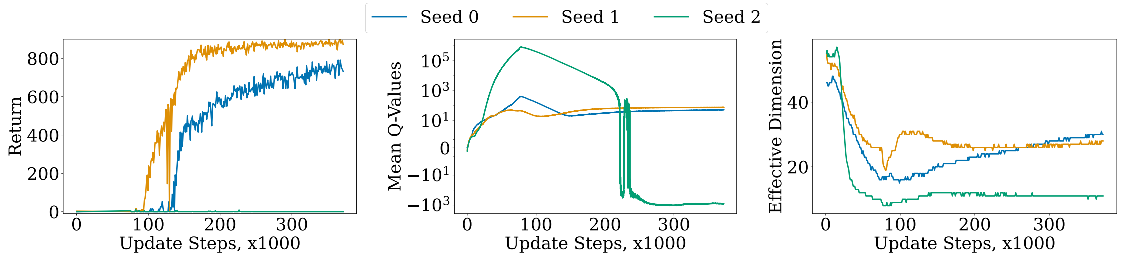

Figure 14: Returns, Mean Q-values and effective dimension for 3 seeds of standard priming for 75,000 steps. When divergence happens, effective dimension is lost. If the effective dimension drops too far, the agent has difficulties to recover.

C Additional experimental results

C.1 Returns on all environments

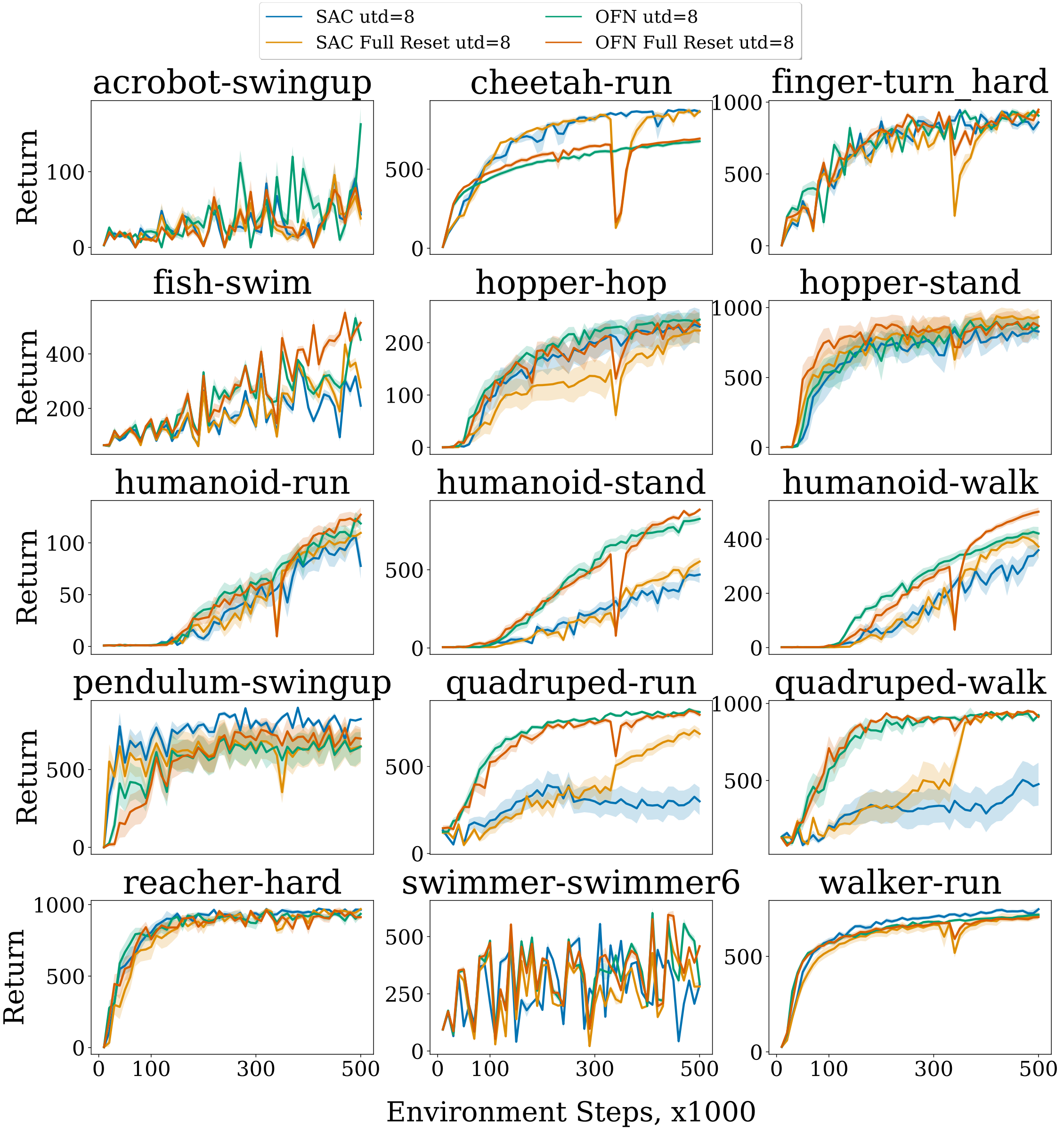

Figure 15: UTD8 Returns on Full DMC15-500K.

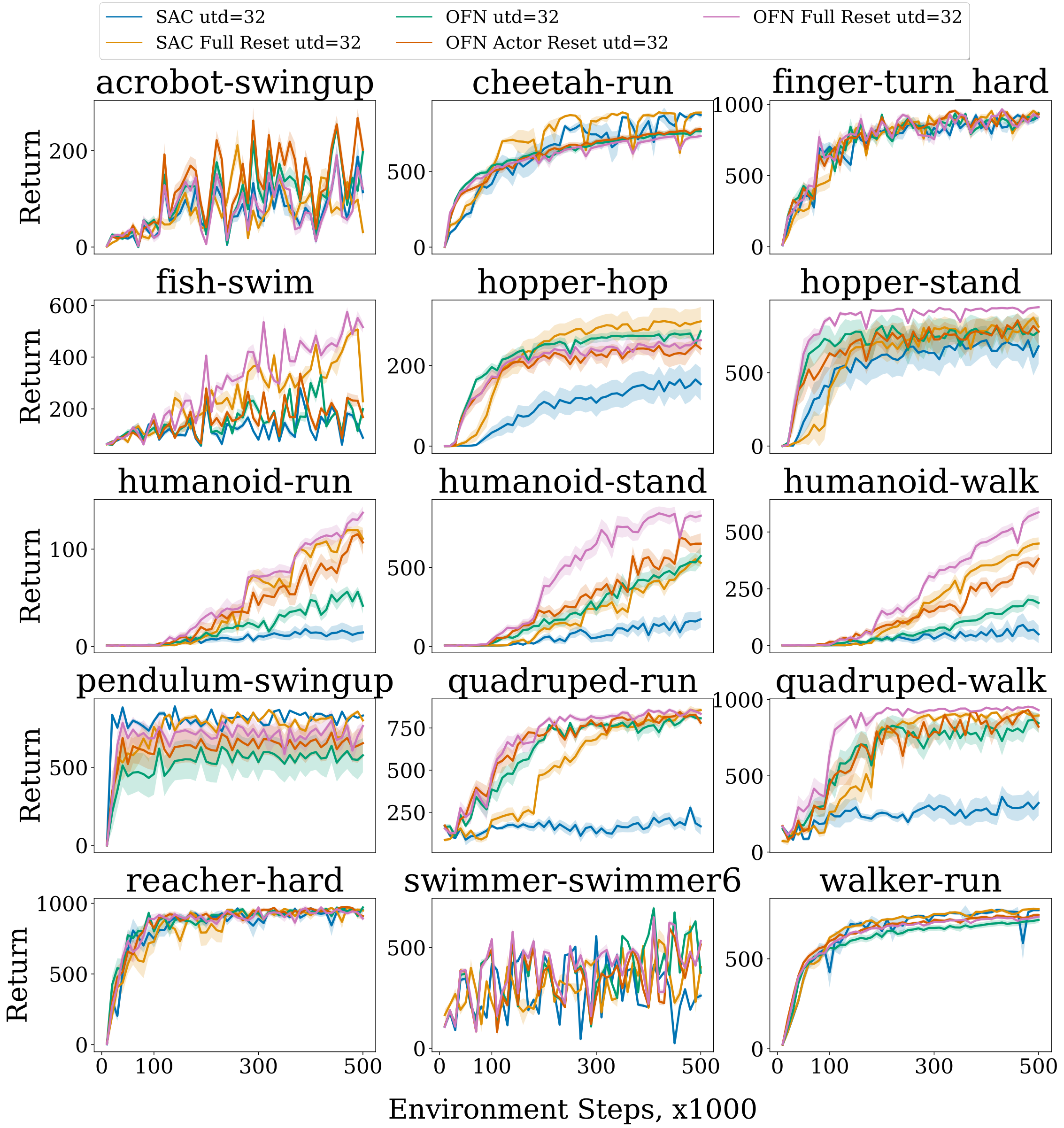

Figure 16: UTD32 Returns on Full DMC15-500K.

C.2 Q-values on all environments

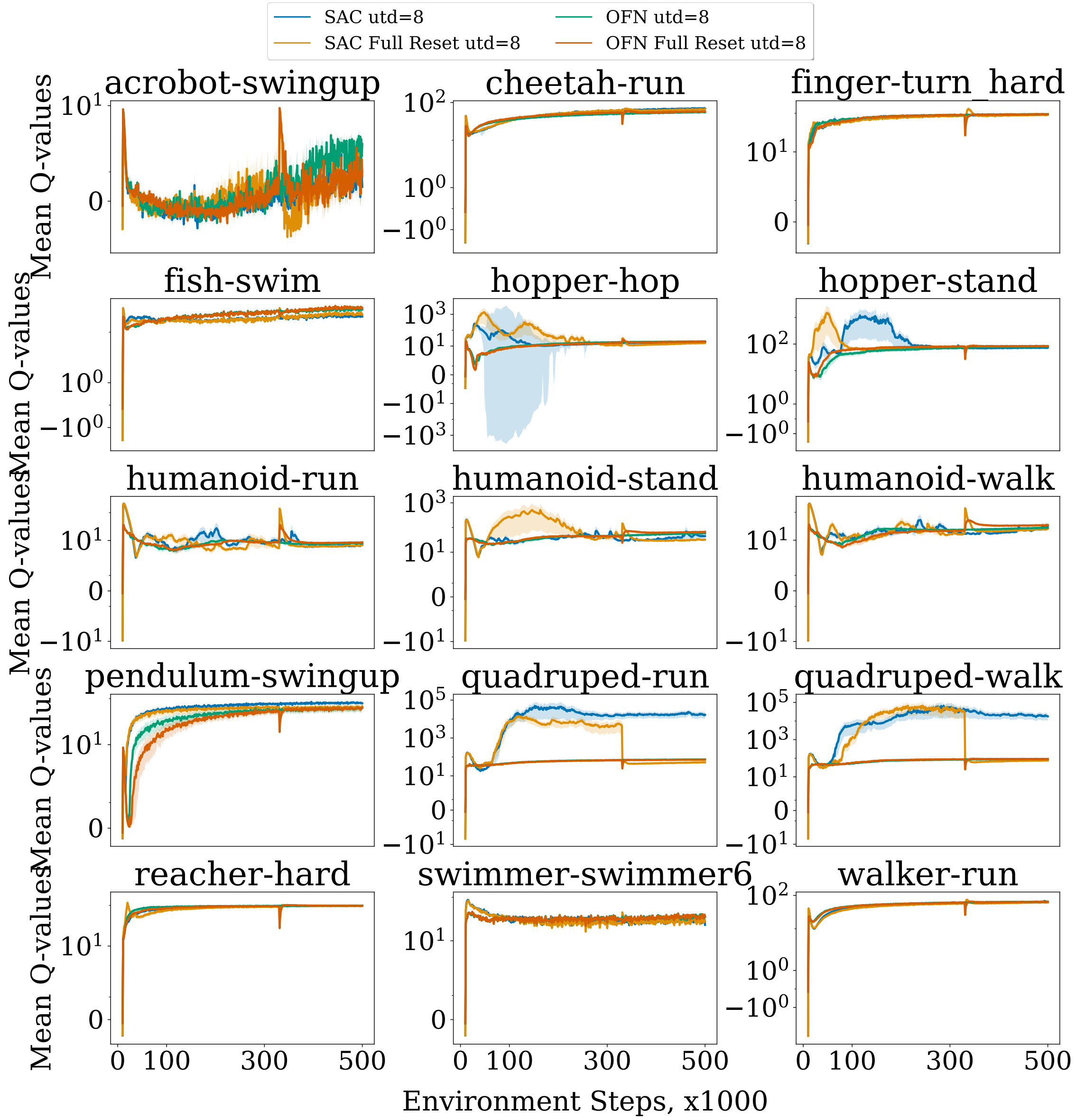

Figure 17: UTD8 Q-values on Full DMC15-500K. Resetting often works when Q-values diverge. ONF mitigates divergence.

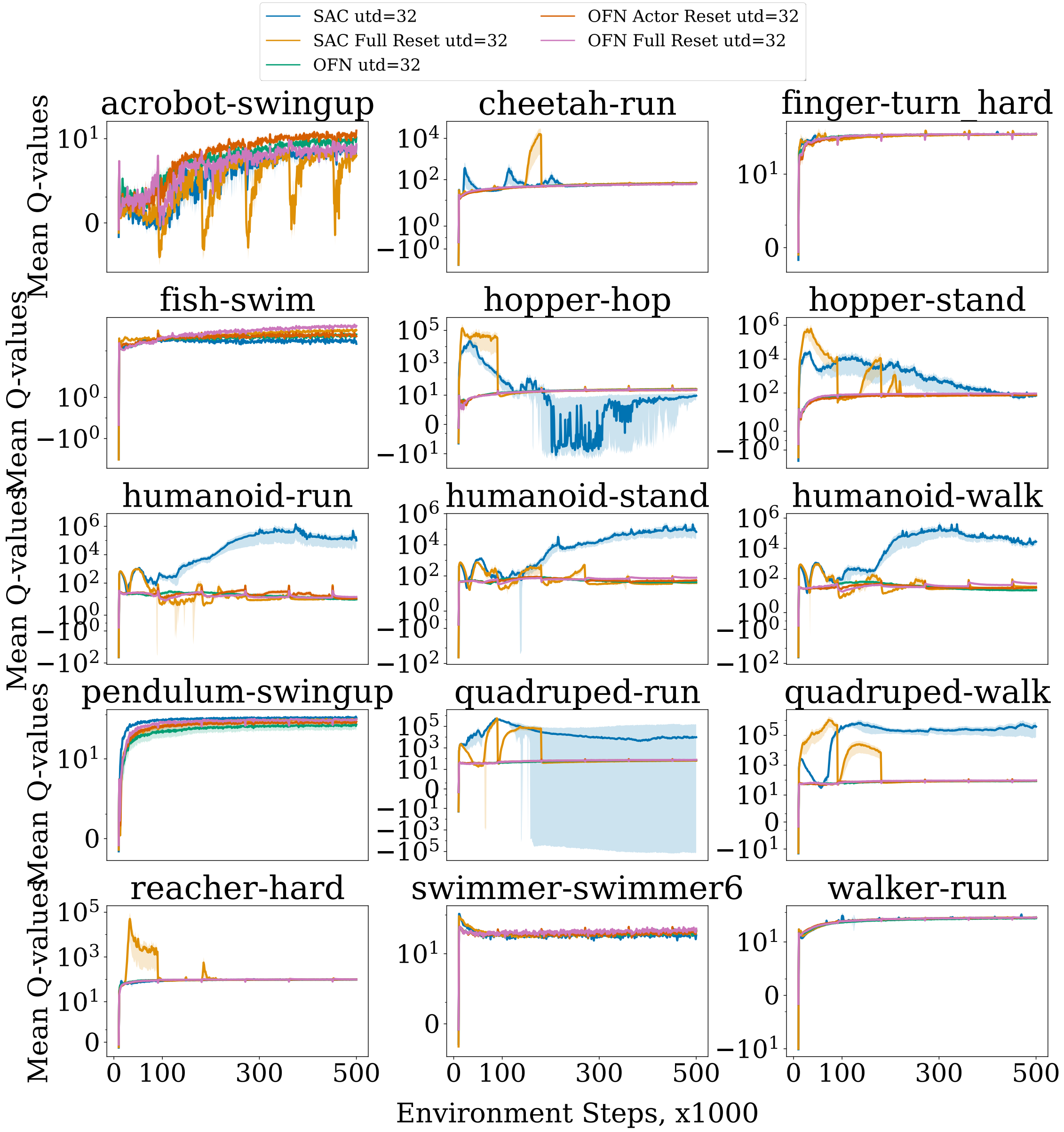

Figure 18: UTD32 Q-values on Full DMC15-500K. Resetting often works when Q-values diverge. ONF mitigates divergence.

D Unit norm gradient derivation

Here, we take a look at the gradient of the unit norm projection.

Let $i \in 1, ..., N$, for all $\mathbf{x} = (x_1, ..., x_n) \in \mathbb{R}^n \setminus \{0\}$. Suppose $f(\mathbf{x}) = \dfrac{\mathbf{x}}{\|\mathbf{x}\|}$.

Then,

$$\begin{aligned}
\partial_i f(\mathbf{x}) &= \frac{\|\mathbf{x}\| e_i - \mathbf{x}\partial_i \| \cdot \|(\mathbf{x})}{\|\mathbf{x}\|^2} \\
&= \frac{\|\mathbf{x}\| e_i - \dfrac{x_i}{\|\mathbf{x}\|}\mathbf{x}}{\|\mathbf{x}\|^2} \\
&= \frac{1}{\|\mathbf{x}\|} e_i - \frac{x_i}{\|\mathbf{x}\|^3}\mathbf{x}
\end{aligned}$$

Note that the second term can grow quite large if the norm of $\mathbf{x}$ is relatively small. Despite this fact, we are able to remedy the exploding gradients using unit norm projection, likely because gradients are small when the norm is small.

E Open Problems and Limitations

Feature divergence without regularization is an important problem that contributes substantially to the issues facing high-UTD learning However, as our experiments show, there are many additional open problems that introducing normalization does not address.

Understanding actor issues The resetting experiments in Figure 7 highlight that a part of the performance impact of high UTD comes from the actor optimization, not the critic optimization, as resetting the actor can boost performance without changing the critic. Our work does not address this issue, and to the best of our knowledge there are no specific attempts to investigate the actor optimization process in deep actor-critic reinforcement learning.

RL Optimizer As the priming experiments show (Figure 13), the update dynamics introduced by momentum terms in modern optimizers can exacerbate existing overestimation problems. Dabney (2014) derives adaptive step-sizes for reinforcement learning from a theoretical perspective, but the resulting optimization rules have not been adapted to Deep Reinforcement Learning to the best of our knowledge. A recent study by Asadi et al. (2023) shows that resetting the optimizer can have some benefit in the DQN setting, where it can be tied to the hard updates of the target Q network. In addition, Lyle et al. (2023) show that optimizers like Adam can lead to reduced plasticity of neural networks. However, our experiments also highlight that without the accelerated optimization of modern optimizers, convergence of the Q value can be prohibitively slow, highlighting the urgent need for stable and fast optimization in RL.

Conservative Learning for Online RL Most current actor-critic methods use some form of pessimistic value estimate to combat the overestimation bias inherent in off-policy Q learning. i.e. via the use of a twinned Q network (Fujimoto et al., 2018). However, this can lead to pessimistic under-exploration (Lan et al., 2020). To address this, Moskovitz et al. (2021) propose to tune the relative impact of pessimistic and optimistic exploration for the environments, while Lee et al. (2021) show that by combining independent critic estimates from ensembles, a UBC like exploration bound can be computed. These changes could be combined with the mitigation strategies for the feature layer divergence in future work to mitigate the harmful effects of underexploration further.

As our work shows, some of the previous problems with overestimation might not emerge from the bias introduced by off-policy actions, but from the learning dynamics of neural network updates. This suggests that more work on the exact causes of overestimation might allow us to move beyond the overly pessimistic twinned network minimization trick without needing costly solutions like ensemble methods.

Tau The rate of the target network updates is an important hyperparameter in online RL, either through periodic hard copies (Mnih et al., 2013) or the use of a Polyak averaging scheme (Lillicrap et al., 2016). Updating the network too fast can exacerbate the impact of value divergence, while updating too slowly can delay learning. Preliminary experiments show a relationship between value divergence and target update speed that requires further investigation.

There have also been attempts to accelerate optimization not via the neural network optimization, but through adapting the updates of the target networks (Vieillard et al., 2020; Farahmand & Ghavamzadeh, 2021). This is an orthogonal direction to the one presented here, and the interplay between target network updates and neural network optimization steps are an important topic for future work.

Reward Shaping Impact In several environments, we observe almost no detrimental effects due to high update ratios, while in others the Q-values diverge even without moving beyond one update per sample collected. A closer inspection suggests that environments in which the initial reward is small and uninformative are much more prone to lead to catastrophic divergence, suggesting a close connection between reward shaping and divergence. While sparse reward problems have received much attention in the context of exploration, our findings suggests that they also present a challenge for efficient optimization. Beyond this phenomenon, the interactions between optimization and explorations have been hypothesized to be a strong contributing factor to the good performance of some algorithms (Schaul et al., 2022), but the role diverging Q-values play in this phenomenon is to the best of our knowledge mostly unexplored.