Optimistic critics can empower small actors

Olya Mastikhina, Dhruv Sreenivas, Pablo Samuel Castro

Keywords: Deep reinforcement learning, actor-critic, asymmetric actor-critics, exploration, value underestimation, data collection

Summary

Actor-critic methods have been central to many of the recent advances in deep reinforcement learning. The most common approach is to use *symmetric* architectures, whereby both actor and critic have the same network topology and number of parameters. However, recent works have argued for the advantages of *asymmetric* setups, specifically with the use of smaller actors. We perform broad empirical investigations and analyses to better understand the implications of this.

Contribution(s)

1. We show that reducing the size of the actor in actor-critic methods can lead to degraded performance and increased overfitting in the critic.
   Context: Prior work suggests that actors require less capacity than critics in actor-critic algorithms (Mysore et al., 2021), and that asymmetric training with smaller actors can be beneficial for real-world applications (Degrave et al., 2022).
2. We demonstrate that performance degradation and critic overfitting is largely due to poorer data collection, and this arises due to value underestimation.
   Context: This is somewhat surprising, as it stands in contrast to the *over-estimation* that's commonly addressed in many popular algorithms (Hasselt, 2010; Hasselt et al., 2016; Fujimoto et al., 2018). However, other papers have shown that underestimation can be an issue with the actor-critic algorithms that address overestimation (Ciosek et al., 2019; Li et al., 2023b; He & Hou, 2020).
3. We explore a number of approaches for mitigating the value underestimation and find the most effective one to be replacing the $\min$ term with an average or $\max$ term when combining the value estimates of two critics (as done in SAC).
   Context: Taking the minimum of two estimated Q-values will, by definition, be conservative; indeed, the idea was originally proposed to deal with over-estimation (Hasselt, 2010). Prior work has shown that resetting or regularizing the critic in particular improves plasticity (Ma et al., 2023; Nikishin et al., 2022; Liu et al., 2021) and can help mitigate value-estimation issues, particularly in the case of layer normalization (Nauman et al., 2024).

Optimistic critics can empower small actors

Olya Mastikhina [†, 1], **Dhruv Sreenivas**[†, 1], **Pablo Samuel Castro**[1,2]
{olya.mastikhina,dhruv.sreenivas,pablo-samuel.castro}@mila.quebec

[1]**Mila, Université de Montréal**
[2]**Google DeepMind**

[†] Equal contribution.

Abstract

Actor-critic methods have been central to many of the recent advances in deep reinforcement learning. The most common approach is to use *symmetric* architectures, whereby both actor and critic have the same network topology and number of parameters. However, recent works have argued for the advantages of *asymmetric* setups, specifically with the use of smaller actors. We perform broad empirical investigations and analyses to better understand the implications of this and find that, in general, smaller actors result in performance degradation and overfit critics. Our analyses suggest *poor data collection*, due to value underestimation, as one of the main causes for this behavior, and further highlight the crucial role the critic can play in alleviating this pathology. We explore techniques to mitigate the observed value underestimation, which enables further research in asymmetric actor-critic methods.

1 Introduction

Actor-critic (AC) algorithms are a fundamental part of deep reinforcement learning (RL), with various AC methods achieving state-of-the-art performance in complex discrete control (Espeholt et al., 2018) and continuous control (Haarnoja et al., 2018a) tasks. In these approaches, the actor interacts with the environment to collect data and to optimize a mapping of states to actions with the guidance of the critic, while the critic learns a value function with the collected data to guide the actor's learning. These symbiotic, but differing, roles have been traditionally implemented with either coupled or matching ("symmetric") neural network architectures (Haarnoja et al., 2018b; Yarats et al., 2021); however, recent work suggests that the actor requires less capacity and can be significantly reduced relative to the critic (Mysore et al., 2021).

As only the actor is used during inference, reducing the size of the actor while keeping a bigger critic offers several advantages for real-world applications. A smaller actor reduces inference costs, which is beneficial for resource-constrained applications such as robotics, where fast computations are essential for real-time performance (Hu et al., 2024; Schmied et al., 2025), and inference time is a bottleneck for deployment (Firoozi et al., 2024). Decoupling the size of the actor from the critic allows for bigger critics that can fully leverage data available in simulators for learning complex tasks without then affecting inference costs. This approach has recently been successfully applied to training an RL agent for the magnetic control of tokamak plasmas for nuclear fusion - an application that requires particularly fast computation speeds (Degrave et al., 2022).

Beyond computational constraints, another barrier to real-world deployment is interpretability and the incorporation of safety constraints, which are particularly important for safety-critical applications like autonomous driving (Tang et al., 2024; Xu et al., 2023; Xiao et al., 2022). Smaller actors tend to generate simpler policies which are easier to interpret (Fan et al., 2021; Li et al., 2022). While distillation is another promising approach for generating compact policies for real-world deployment

(Hinton et al., 2015; Rusu et al., 2016; Liu et al., 2024), direct training makes the incorporation of safety and functional constraints simpler and more reliable.

Despite their apparent advantages, there has been little work in developing an understanding of how to properly train asymmetric AC methods with smaller actors, as well as how the actor-critic relationship is affected by this asymmetry. In this paper, we address this gap by performing a broad empirical investigation with the Soft Actor-Critic (SAC; Haarnoja et al., 2018b) and Data-Regularized Q (DrQ; Yarats et al., 2021) agents in the physics-simulated DeepMind Control suite (DMC; Tassa et al., 2018; Tunyasuvunakool et al., 2020) environments. We reduce the number of parameters in the actor (sometimes as far down as 1% of its original size) and observe increased overfitting in the critics as actor size decreases. However, rather than this being a hard limitation due to capacity loss, our analyses suggest that this performance drop can mostly be attributed to poorer data collection by the actor, which may be caused by pessimistic under-exploration problems with algorithms like SAC and DrQ that compute the minimum of Q value estimates (Ciosek et al., 2019; Haarnoja et al., 2018b; Yarats et al., 2021). Notably, we find that simply alleviating value under-estimation in the critics can drastically improve performance. We show a similar mitigation effect for a drop in performance caused by the actor receiving limited information, suggesting assisting constrained actors with optimism may be a general strategy for conservative AC methods.

The paper is organized as follows: in section 2, we lay the groundwork and explain our experimental setup. In section 3, we show the performance effects when naively reducing a smaller actor across a variety of state-based and image-based continuous control tasks, and analyze what could be the cause of performance differences. In section 4, we focus on interventions that can gain back performance, specifically focusing on bias correction and value function underestimation. Finally, we conclude with discussions and avenues for future work in section 5.

2 Preliminaries

Reinforcement learning (RL) agents learn by interacting with an environment, which is typically formulated as a Markov decision process (MDP) $\langle \mathcal{X}, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle$ (Puterman, 1994). Here, $\mathcal{X}$ denotes the agent state space; $\mathcal{A}$ is the set of actions available to the agent; $\mathcal{P} : \mathcal{X} \times \mathcal{A} \to \Delta(\mathcal{X})$ are the transition dynamics with $\mathcal{P}(x' \mid x, a)$ indicating the probability of transitioning to state $x' \in \mathcal{X}$ after selecting action $a \in \mathcal{A}$ from state $x \in \mathcal{X}$; $\mathcal{R} : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ is the reward function, where $\mathcal{R}(x, a)$ denotes the reward received after performing action a from state x. An agent's behavior is quantified by a *policy* $\pi : \mathcal{X} \to \Delta(\mathcal{A})$, where $\pi(a \mid x)$ denotes the probability of selecting action a when in state x. The estimated returns of a policy π from state x are quantified via the (recursive) *value function* $V^\pi(x) := \mathbb{E}_{a \sim \pi(\cdot|x)} \left[\mathcal{R}(x, a) + \gamma \mathbb{E}_{x' \sim \mathcal{P}(\cdot|x,a)} V^\pi(x') \right]$, where $\gamma \in [0, 1)$ is a discount factor that discourages waiting too long before obtaining rewards. We can define the *state-action value function* $Q^\pi(x, a)$, which quantifies the value of taking an arbitrary action a from state x, and then following π afterwards: $Q^\pi(x, a) := \mathcal{R}(x, a) + \gamma \mathbb{E}_{x' \sim \mathcal{P}(\cdot|x,a)} V^\pi(x')$. One can easily see that $V^\pi(x) = \mathbb{E}_{a \sim \pi(\cdot|x)} Q^\pi(x, a)$. The goal of RL is to find an *optimal* policy π^* which maximizes returns, in the sense that $V^{\pi^*} \geq V^\pi$ for all π. There are a number of techniques for learning optimal policies, most of which alternate between *policy evaluation* and *policy improvement*. Policy evaluation seeks to estimate the value function of a policy π, which is primarily done by minimizing temporal-difference (TD) errors:

$$TD^\pi(x, a, x') = |Q^\pi(x, a) - (\mathcal{R}(x, a) + \gamma V^\pi(x'))|. \tag{1}$$

Policy improvement then follows by directly maximizing this Q function, either via a standard arg-max over actions or gradient ascent. Actor-critic methods operate by maintaining separate estimates of π_θ (the actor) and Q_ϕ (the critic), which are used in each of the learning objectives; in deep RL, these functions are approximated by neural networks, parameterized by θ and ϕ, respectively. Given a dataset $\mathcal{D}$ of transitions (often stored in a replay buffer), Soft Actor-Critic (**SAC**; Haarnoja et al.,

2018a;b) optimizes the actor and critic by minimizing the following losses:

$$J_Q(\phi) = \mathbb{E}_{x,a,x'\sim\mathcal{D}}\left[\frac{1}{2}\left(Q_\phi(x,a) - \left(\mathcal{R}(x,a) + \gamma V_{\bar{\phi}}(x')\right)\right)^2\right] \tag{2}$$

$$J_\pi(\theta) = \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{a\sim\pi_\theta(\cdot|x)}\left[\alpha \log \pi_\theta(a \mid x) - Q_\phi(x,a)\right]\right] \tag{3}$$

In eq. (2), $V_{\bar{\phi}}$ is the value function computed from Q_ϕ via $V_{\bar{\phi}}(x) = \mathbb{E}_{a\sim\pi_\theta(\cdot|x)}\left[Q_{\bar{\phi}}(x,a) - \alpha \log \pi_\theta(a \mid x)\right]$, where $\bar{\phi}$ are delayed target parameters (Mnih et al., 2015), and α is a learned Lagrange multiplier (we exclude its parameterization for simplicity of exposition). In their practical implementation, Haarnoja et al. (2018b) use two Q value estimates with parameters ϕ_1 and ϕ_2, trained independently, and take their minimum in the update terms in equations 2 and 3, resulting in the following updated losses, with $V_{\bar{\phi}}(x) = \mathbb{E}_{a\sim\pi_\theta(\cdot|x)}\left[\min_{i\in\{1,2\}} Q_{\bar{\phi}_i}(x,a) - \alpha \log \pi_\theta(a \mid x)\right]$:

$$J_Q(\phi_i) = \mathbb{E}_{x,a,x'\sim\mathcal{D}}\left[\frac{1}{2}\left(Q_{\phi_i}(x,a) - \left(\mathcal{R}(x,a) + \gamma V_{\bar{\phi}}(x')\right)\right)^2\right] \tag{4}$$

$$J_\pi(\theta) = \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{a\sim\pi_\theta(\cdot|x)}\left[\alpha \log \pi_\theta(a \mid x) - \min_{i\in\{1,2\}} Q_{\phi_i}(x,a)\right]\right] \tag{5}$$

It is important to note the interconnectedness of these losses: the actor influences the critic via the (soft) value function $V_{\bar{\phi}}$ used in eqs. (2) and (4), while the critic influences the actor via Q_ϕ in eqs. (3) and (5). Additionally, the actor influences the training dynamics of both given that it is in charge of data collection. Finally, note the use of the TD-error term in eqs. (2) and (4).

2.1 Experimental setup

We run our experiments on the DeepMind Control suite (DMC; Tassa et al., 2018; Tunyasuvunakool et al., 2020), a suite of continuous control tasks that have been a staple of continuous-action reinforcement learning research. For any of the tasks, DMC can provide either low-dimensional features or pixel observations to the agents, while keeping the underlying transition and reward dynamics unchanged. Pixel-based observations are generally more challenging, as the MDP is partially observed (Kaelbling et al., 1998; Yarats et al., 2020), but investigating both provides richer insights into the dynamics of the examined learning algorithms.

Due to computational limitations, the bulk of our analyses will be on feature-based tasks. For these, we use as baseline the default set up and parameters for DMC (Haarnoja et al., 2018b). This consists of one actor network, two critic networks, and two critic target networks. The critic and target networks consist of two hidden layers of size 256 and output a one dimensional Q value estimate. By default, the actor consists of hidden layers of size 256, with two output layers that parameterize the mean and standard deviation of a Gaussian distribution squashed by a tanh function. The critic and actor networks are decoupled, in the sense that they share no parameters.

For pixel observations we use **DrQ**, which enhances SAC's performance via data augmentation (Yarats et al., 2021). We replace the standard DrQ architecture of Yarats et al. (2021) with a larger one recommended for faster learning (Nikishin et al., 2022; Kostrikov, 2021), which consists of an encoder followed by two MLPs for the actor and two critics. The encoder consists of four convolutional layers with output feature maps $\{32, 64, 128, 256\}$ and strides $\{2, 2, 2, 2\}$, respectively, followed by a linear projection to a 50-dimensional output, layer normalization (Ba et al., 2016), and then a tanh activation; the MLPs consist of two dense 256-dimensional layers, with output layers defined exactly as is done above with SAC. As in SAC, we use decoupled architectures for both the actor and the critics, unlike the original baseline, in which the encoder is shared.

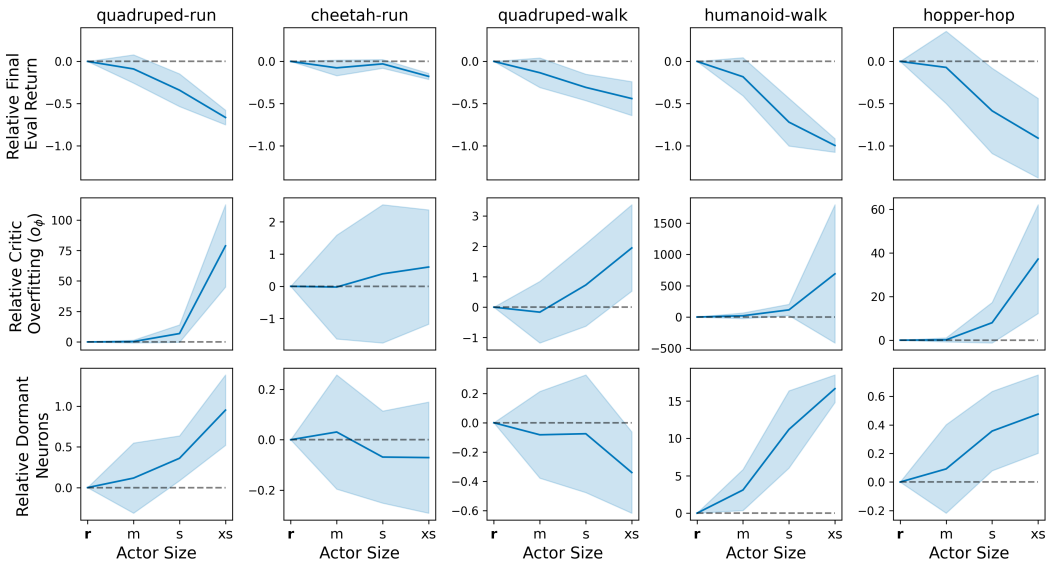

Figure 1: **Decreasing the size of the actor in SAC decreases performance (top row) and increases overfitting in the critics**, as measured by o_ϕ (Nauman et al., 2024, middle row) and dormant neurons (Sokar et al., 2023, bottom row). In the top row, the y-axis is kept fixed to show the relative performance impacts across environments; this becomes impractical for the metrics in the middle and bottom row. We report the final performance, where the solid lines indicate the mean, while the shaded area represents the 95% confidence interval, as computed from 10 seeds. In all rows we report values relative to the default baseline.

3 The impact of small actors

We begin by evaluating the impact on performance resulting from reduced actors. We use default hyperparameters (Haarnoja et al., 2018b) and keep the critic architecture fixed, but explore reducing the dimensionality of the actor. We denote by **r** (for **r**egular) the default dimensionality discussed above and use the following labels to indicate the dimensionality of the two dense hidden layers in SAC: **m**: 128; **s**: 32; **xs**: 8. The latter correspond to network weight numbers that are 32%, 5%, and 1% that of the default actor, respectively. In DrQ, we follow the same procedure as with SAC when modifying the projection MLPs, which leads to a corresponding parameter reduction as mentioned earlier in the MLPs. However, to further reduce expressivity, we also reduce the number of convolutional layers as follows: **m**: $\{32, 64, 128\}$; **s**: $\{32, 64\}$; **xs**: $\{32\}$. This results in more overall parameters for the DrQ actors, but this increase is at the encoder representation level, not at the direct policy level. To quantify the impact of the reduced actors, we report values relative to the baseline values. For instance, for a measure X_s obtained with the **s** actor, we report $\frac{X_s - X_r}{X_r}$, where X_r is the value obtained with the default actor.

In the top row of fig. 1 we evaluate the impact on performance when reducing the size of these layers and can see a clear degradation in performance across all environments. We additionally measure o_ϕ on the critics, introduced by Nauman et al. (2024) as a measure of overfitting, defined as $o_\phi := \frac{\mathbb{E}_{\mathcal{D}_V} TD_\phi}{\mathbb{E}_{\mathcal{D}} TD_\phi}$. Here, $\mathcal{D}_V$ is a validation dataset of size $11,000$, containing data sampled from a training run with a regular unmodified SAC agent, trained with a different random seed, and TD_ϕ is the temporal difference error. Higher values of o_ϕ are indicative of overfitting which, as seen in the middle row of fig. 1, are inversely correlated with the size of the actor. Finally, we report the fraction of dormant neurons, defined as the proportion of neurons that are 0 for every data point in the validation buffer, where higher levels of dormancy is associated with a loss of plasticity (Sokar et al., 2023; Lyle et al., 2024; Klein et al., 2024). In the bottom row of fig. 1 we see that the fraction

of dormant neurons tends to be inversely correlated with actor size and performance, particularly for the environments where the performance loss is greatest, although to a lesser extent than o_ϕ.

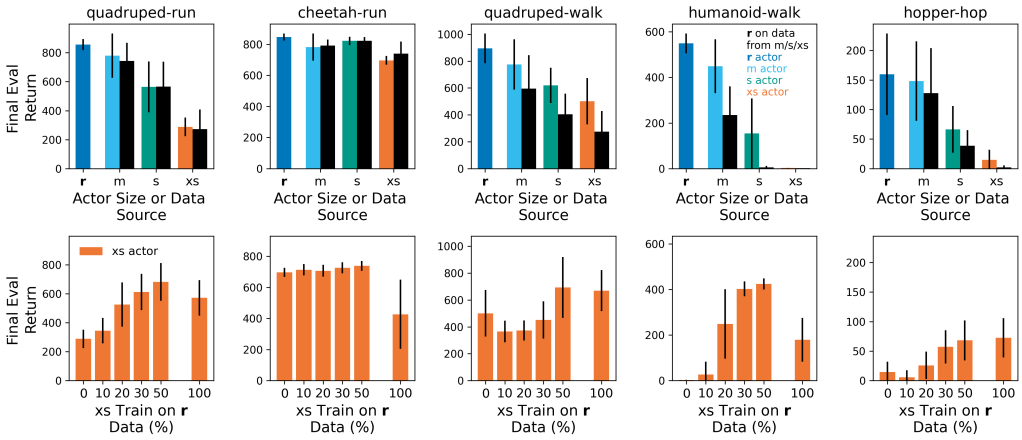

Figure 2: **Evaluating the impact of data quality collecting by actors of varying sizes.** Top row: the black bars denote training regularly-sized π_r on data collected by one of the smaller actors, while the colored bars indicate each actor trained on its own data. Bottom row: the smallest (**xs**) actor trained on data from the largest (**r**) actor, for varying fractions of training length. In both rows we report the final performance, where the bars indicate the mean, while the error bars represent the 95% confidence intervals, as computed from 10 independent seeds.

Figure 8 illustrates the impact of actor reduction in DrQ, where in the pixel-based case we focus on evaluation return and critic overfitting as measured by o_ϕ. As with SAC, we see a decrease in performance with smaller actors, as well as a general increase in o_ϕ.

3.1 Smaller actors collect worse data

Training overparameterized neural networks on small datasets is a common cause for overfitting. Given that the actor is in charge of data collection and reducing its expressivity results in overfit critics, we continue our investigation by evaluating the quality of the data gathered by differently-sized actors. For this, we evaluate training on data collected by separate, and differently-sized, actors. Specifically, we train the regularly-sized actor π_r with data provided by one of the smaller actors, where the data collection exactly mimics that obtained by the smaller actor during training. This is depicted by the black bars in the top row of fig. 2, where we can see the performance to be clearly correlated with actor size.

Nikishin et al. (2022) demonstrated the tendency of RL agents to overfit to early experience, affecting their plasticity and downstream performance. It is thus worth considering whether the quality of the training data on an actor is most important in the early stages of training. To evaluate this, in the bottom row of fig. 2 we explore training the smallest actor (π_{xs}) on data provided by the π_r actor, again matching the data collection of the smaller actor. Our analyses here explore using the data from π_r for only a fraction of training, and then switching to data collected by π_{xs} itself. As more data is collected from the bigger actor π_r, the performance of π_{xs} generally improves. We note that using all of the data from π_r (i.e. at 100%) sometimes results in degraded performance; we hypothesize that this may be due to the tandem effect observed by Ostrovski et al. (2021). Overall, we see an improvement in performance in the environments most impacted by reducing the size of the actor for SAC. With DrQ, we do not see a pronounced effect when training the smallest actor on data from a regular-sized actor (see fig. 9), but similarly, a small trend may be observed for environments with the biggest degradation in performance with reduced actor sizes.

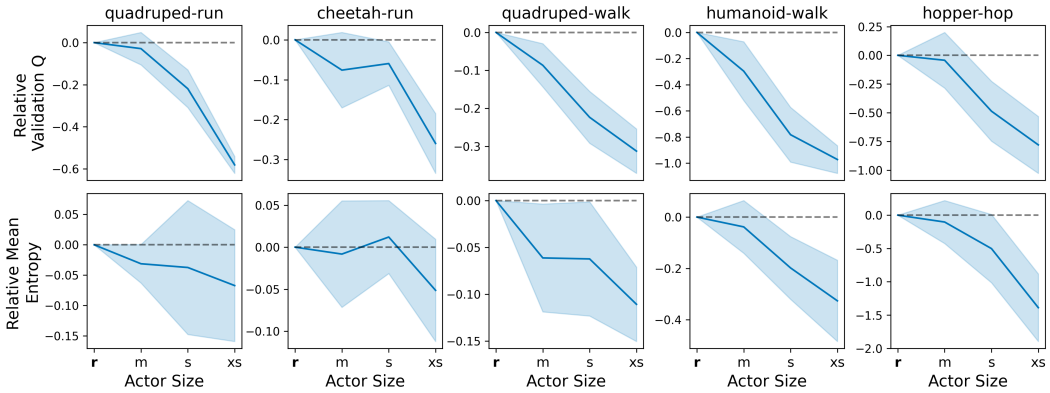

Figure 3: **Decreasing the size of the actor results in Q-value underestimation and reduced policy entropy.** In the top row we estimate the average Q-values on a batch of data gathered during evaluation, and plot the values relative to the baseline **r**. In the bottom row we compute the entropy of the policy π and plot the values relative to the entropy of the baseline **r**. In both cases we report the values obtained at the end of training, where the solid line represents the mean with shaded areas indicating 95% confidence intervals, computed over 10 independent seeds.

3.2 Smaller actors result in critic underestimation

The reduction in the quality of data gathered by small actors can possibly be attributed to under-exploration of the state space. This can often be a consequence of an overly-conservative critic which under-estimates values, as well as a low-entropy actor with low diversity in action selection. In fig. 3 we compare the average critic validation Q-values (computed on the same validation dataset) as well as the entropy of the actor's action distribution π of the smaller actors relative to the regularly-sized actor, and confirm that smaller actors result in Q-value underestimation, as well as a reduction in entropy during training (see fig. 11 for the same comparison throughout training). The observed underestimation is interesting, given that it stands in contrast to the *over-estimation* that's commonly addressed in many popular algorithms (Hasselt, 2010; Hasselt et al., 2016; Fujimoto et al., 2018).

4 Empowering small actors

The results from the last section suggest that the performance reduction resulting from the use of small actors is largely due to poor data collection, which in turn appears to be a consequence of value underestimation and low action variability. In this section we explore a variety of approaches for strengthening small actors.

4.1 Average and maximal critics

We begin by a simple modification to the original SAC losses to directly address the observed value underestimation. Specifically, we replace the minimization of the two independent Q estimates in equations 4 and 5 with either their mean ($\text{avg}(Q_{\phi_1}, Q_{\phi_2})$) or their maximum ($\max(Q_{\phi_1}, Q_{\phi_2})$). As can be seen in the top and middle rows of fig. 4 and the top row of fig. 5, this approach can be quite effective at boosting the performance of small actors in SAC, sometimes even improving over the minimization approach with the regular sized model (e.g. hopper-hop). The bottom row of fig. 5 confirms that this technique does increase the validation value estimates. As can be seen in fig. 13, we find that the mean and the max approaches also improve several overfitting and plasticity metrics in the critics, most notably o_ϕ and the rank of the last hidden layer (Kumar et al., 2021; Nauman et al., 2024). However, they do not appear to have a notable impact on these metrics in the actor (see fig. 14). The results on the smallest actor on DrQ (bottom row of fig. 4) display a similar

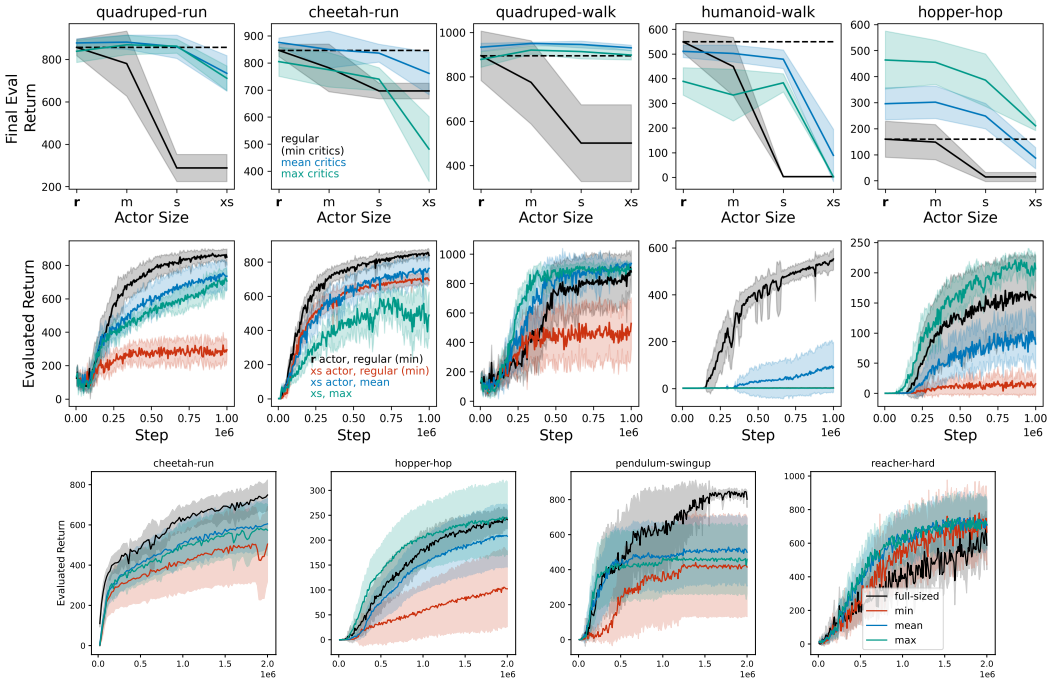

Figure 4: **Taking the mean or max of the two critics can empower smaller actors** in SAC (top and middle rows) and on the smallest actor in DrQ (bottom row). Replacing the minimums in equations 4 and 5 with mean and max can help reduce Q-value underestimation and boost performance. The top row displays final performance while the bottom two rows display performance throughout training, with solid lines indicating average over 10 seeds, and shaded areas 95% confidence intervals. In DrQ, the performance over the non-full-sized settings is computed using 20 seeds to account for higher observed variance.

performance trend, although the results are less pronounced. We also observe a corresponding trend with an increase in validation Q values with the mean and max approaches in DrQ in fig. 10.

4.2 Critic regularization

Prior work has shown that resetting or regularizing the critic in particular improves plasticity (Ma et al., 2023; Nikishin et al., 2022; Liu et al., 2021) and can help mitigate value-estimation issues, particularly in the case of layer normalization, (Nauman et al., 2024), albeit with overestimation. Given both the increased overfitting observed in the critics, and how much value estimation is affected by smaller actors (see fig. 3), we investigate whether critic regularization alone can be effective mitigating this impact by applying a number of regularization techniques, focusing on SAC: (a) **Layer Normalization** (Ba et al., 2016); (b) **Spectral Normalization** (Miyato et al., 2018); (c) **weight decay** (van Laarhoven, 2017) with a regularization value of 0.01 (Li et al., 2023a); (d) **L2 distance from initialization** (Kumar et al., 2024): with a value of 1×10^{-7} after tuning on the range $[5 \times 10^{-8}, 1 \times 10^{-4}]$ in increments of 0.5 with quadruped-run; and (e) **Network resets:** resetting neural network layers during training (Nikishin et al., 2022). We apply layer normalization and spectral normalization to the second hidden layer in the critics, and we reset only the output layer of the critics every 50K steps. Although many of these methods do appear to help with mitigating value under-estimation (bottom row of fig. 5), they do not appear to help much with performance (top row of fig. 5 and fig. 12). For DrQ, we investigate resetting the MLP of the critics (Nikishin et al., 2022) for the smallest actors, and similarly do not see a notable rescue effect (see fig. 10).

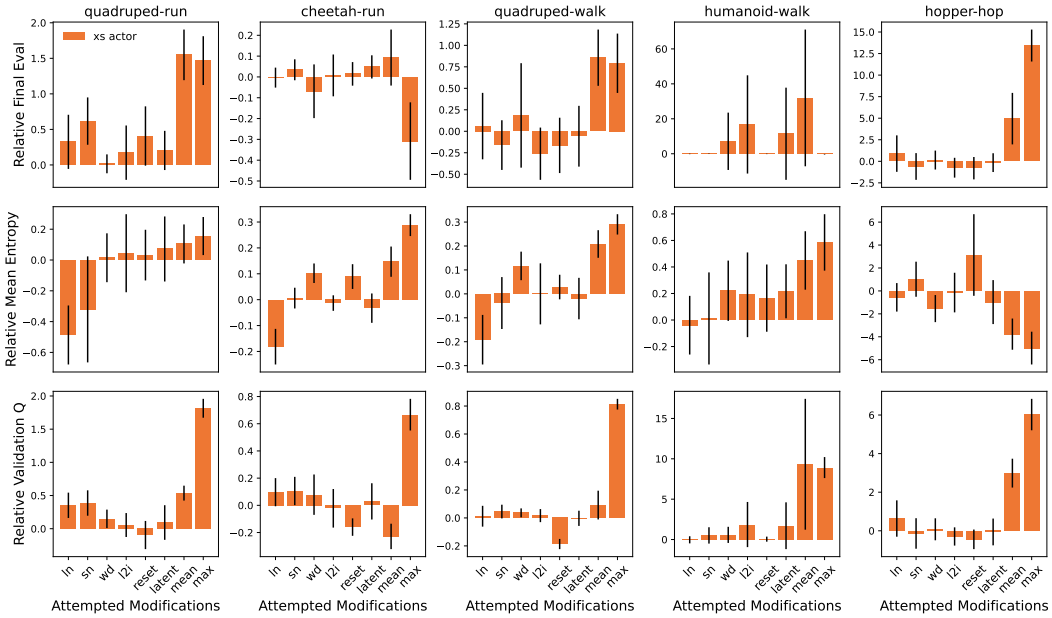

Figure 5: **Impact of attempted modifications on the final performance (top), mean entropy of the actor's action distribution (middle), and validation Q-value estimation (bottom) of the smallest actor in SAC**. The values are relative to the smallest actor in unmodified SAC. Bars indicate mean with error bars denoting 95% confidence intervals, computed over 10 independent seeds.

4.3 Addressing bias in the critic via actor representations

In asymmetric actor-critics methods, imbalances in information received by the critic versus the actor can lead to biased gradients that may negatively impact performance; Baisero & Amato (2022) and Lyu et al. (2022) propose to alleviate this by giving the (limited) information received by the actor as additional input into the critic. In our case, the critics do not receive privileged information over the actor, but we theorize that a similar effect may be occurring within the policy network due to potentially impacted information flow through the smaller actors. We attempt a similar bias correction by concatenating the latent state of the final hidden layer of the actor as input to the final hidden layer of the critics. The latent state is first projected through an untrained neural network layer to a size of 8 to maintain consistency across actor sizes. As shown in the top row of fig. 5 and the bottom row of fig. 12, the bias correction performs similarly to other attempted critic regularization methods in SAC.

5 Discussion

Real-world problems are often subject to constraints such as latency, model size, and interpretability, which are largely absent in the academic benchmarks where machine learning solutions are developed. As such, it is imperative that we develop the necessary techniques for training reinforcement learning agents under such limitations. The use of small actors can help reduce latency, memory, and inference costs, and can help improve interpretability; these are all practical considerations, as ultimately it is a trained actor which will be deployed for action selection. Our work demonstrates that naïvely shrinking the actor can result in value underestimation, poor data collection, and ultimately degraded performance. We evaluated a number of approaches for mitigating this deterioration and found the most effective to be simply replacing the min operation with a mean or max when combining the values of the two critics (section 4.1).

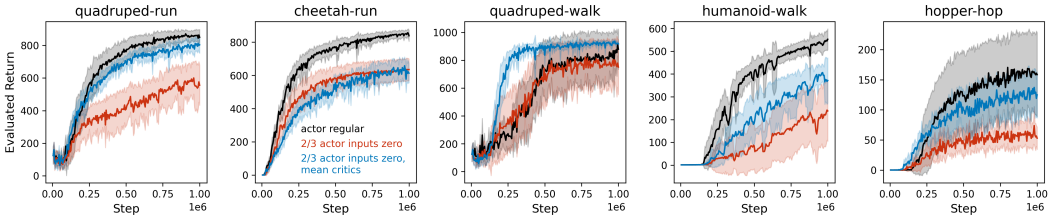

Figure 6: **Taking the mean of the two critics can help deal with partial observability in the actor.** We zero-out 2/3 of the inputs into the actor and compare the performance when using the min or mean of the two critics.

It is often necessary to provide the actor with less information than the critic, as was employed by Vasco et al. (2024) to better match the inputs used by humans. In fig. 6 we explore whether this additional type of limitation on the actor may have a similar effect to what we observe when decreasing the size of the actor. To test this, we zero out two-thirds of the actor inputs in SAC (retaining every third dimension) and find that taking the mean of the two critics - rather than the minimum - alleviates performance loss here as well. Of note, the alleviation is more pronounced in the same environments where underestimation mitigation helped the most with smaller actors (fig. 4). This suggests that addressing underestimation in SAC can additionally help mitigate the challenges arising from partial observability.

Figure 14 suggest that smaller actors result in larger parameters and reduced effective rank, which are often tied to optimization difficulties; figure 13 suggests that these effects are less pronounced on the critics. Interestingly, the most effective technique we found for mitigating value underestimation seems to have little impact on the actor's parameter norms and effective rank, but does seem to play an important role on the critics.

In general, developing a greater understanding between optimization, exploration, expressivity, and estimation accuracy will lead to more robust and reliable reinforcement learning agents. While our work has focused on the case of small actors, the insights provided help strengthen our collective understanding of these learning dynamics. Addressing overestimation in AC methods by taking the minimum of estimated Q values has been a continuing trend - for example, with Deep Deterministic Policy Gradient (DDPG; Lillicrap et al., 2016) being followed by Twin-Delayed DDPG (TD3; Fujimoto et al., 2018). However, our work contributes to findings showing that this approach contributes to underestimation, which warrants further consideration particularly in settings where data collection is more challenging (Ciosek et al., 2019; Li et al., 2023b; He & Hou, 2020). Further, all these considerations are aligned with the continued relevance of the exploration-exploration dilemma (Li et al., 2023b; Sutton & Barto, 2018), which has been explored via Thompson sampling-like techniques (Ishfaq et al., 2025) and through information gain maximization (Sukhija et al., 2025).

Limitations Our empirical investigations were mostly focused on SAC evaluated on DMC with feature-based observations. Although we did conduct subsets of our analyses on DrQ with the more challenging pixel-based observations, further evaluations on different benchmarks and agents would be necessary to strengthen the generality of our claims. For consistency and computational considerations, in our work we used the default hyper-parameters of the baseline models for all experiments; however, RL agents can often be sensitive to hyper-parameter choices (Ceron et al., 2024), so ideally one would perform a hyper-parameter search for each the various settings considered, although this can be computationally prohibitive.

Broader impact statement

This paper presents work whose goal is to advance the field of Reinforcement Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgements

We thank Johan Obando-Ceron, Roger Creus Castanyer, Gandharv Patil, Ayoub Echchahed, Ali Saheb Pasand, Joao Madeira Araujo, and the RLC technical and senior reviewers for providing helpful feedback on our work.

We would also like to thank the Python community (Van Rossum & Drake Jr, 1995; Oliphant, 2007) for developing tools that enabled this work, including NumPy (Harris et al., 2020), Matplotlib (Hunter, 2007), Jupyter (Kluyver et al., 2016), and Pandas (McKinney, 2013).

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. URL https://arxiv.org/abs/1607.06450.

Andrea Baisero and Christopher Amato. Unbiased Asymmetric Reinforcement Learning under Partial Observability. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '22, pp. 44–52, Richland, SC, May 2022. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-1-4503-9213-6.

Johan Samir Obando Ceron, João Guilherme Madeira Araújo, Aaron Courville, and Pablo Samuel Castro. On the consistency of hyper-parameter selection in value-based deep reinforcement learning. In *Reinforcement Learning Conference*, 2024. URL https://openreview.net/forum?id=szUyvvwoZB.

Kamil Ciosek, Quan Vuong, Robert Loftin, and Katja Hofmann. Better exploration with optimistic actor critic. 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. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/a34bacf839b923770b2c360eefa26748-Paper.pdf.

Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan D. Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de Las Casas, Craig Donner, Leslie Fritz, Cristian Galperti, Andrea Huber, James Keeling, Maria Tsimpoukelli, Jackie Kay, Antoine Merle, Jean-Marc Moret, Seb Noury, Federico Pesamosca, David Pfau, Olivier Sauter, Cristian Sommariva, Stefano Coda, Basil Duval, Ambrogio Fasoli, Pushmeet Kohli, Koray Kavukcuoglu, Demis Hassabis, and Martin A. Riedmiller. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nat.*, 602(7897):414–419, 2022. DOI: 10.1038/S41586-021-04301-9. URL https://doi.org/10.1038/s41586-021-04301-9.

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures. In *Proceedings of the 35th International Conference on Machine Learning*. PMLR, July 2018.

Feng-Lei Fan, Jinjun Xiong, Mengzhou Li, and Ge Wang. On Interpretability of Artificial Neural Networks: A Survey. *IEEE Transactions on Radiation and Plasma Medical Sciences*, 5(6), November 2021. ISSN 2469-7303.

Roya Firoozi, Johnathan Tucker, Stephen Tian, Anirudha Majumdar, Jiankai Sun, Weiyu Liu, Yuke Zhu, Shuran Song, Ashish Kapoor, Karol Hausman, Brian Ichter, Danny Driess, Jiajun Wu, Cewu Lu, and Mac Schwager. Foundation models in robotics: Applications, challenges, and the future. *The International Journal of Robotics Research*, September 2024. ISSN 0278-3649.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1582–1591. PMLR, 2018. URL http://proceedings.mlr.press/v80/fujimoto18a.html.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *Proceedings of the 35th International Conference on Machine Learning*. PMLR, July 2018a.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft actor-critic algorithms and applications. *CoRR*, abs/1812.05905, 2018b. URL http://arxiv.org/abs/1812.05905.

Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with numpy. *Nature*, 585(7825):357–362, 2020.

Hado 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.

Qiang He and Xinwen Hou. Wd3: Taming the estimation bias in deep reinforcement learning. In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 391–398. IEEE, November 2020. DOI: 10.1109/ictai50040.2020.00068. URL http://dx.doi.org/10.1109/ICTAI50040.2020.00068.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015. URL https://api.semanticscholar.org/CorpusID:7200347.

Yafei Hu, Quanting Xie, Vidhi Jain, Jonathan Francis, Jay Patrikar, Nikhil Keetha, Seungchan Kim, Yaqi Xie, Tianyi Zhang, Hao-Shu Fang, Shibo Zhao, Shayegan Omidshafiei, Dong-Ki Kim, Ali-akbar Agha-mohammadi, Katia Sycara, Matthew Johnson-Roberson, Dhruv Batra, Xiaolong Wang, Sebastian Scherer, Chen Wang, Zsolt Kira, Fei Xia, and Yonatan Bisk. Toward General-Purpose Robots via Foundation Models: A Survey and Meta-Analysis, October 2024.

John D Hunter. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9 (03):90–95, 2007.

Haque Ishfaq, Guangyuan Wang, Sami Nur Islam, and Doina Precup. Langevin soft actor-critic: Efficient exploration through uncertainty-driven critic learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=FvQsk3la17.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artif. Intell.*, 101(1–2):99–134, May 1998. ISSN 0004-3702.

Timo Klein, Lukas Miklautz, Kevin Sidak, Claudia Plant, and Sebastian Tschiatschek. Plasticity loss in deep reinforcement learning: A survey, 2024. URL https://arxiv.org/abs/2411.04832.

Thomas Kluyver, Benjain Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, Carol Willing, and Jupyter Development Team. Jupyter Notebooks—a publishing format for reproducible computational workflows. In *IOS Press*, pp. 87–90. 2016. DOI: 10.3233/978-1-61499-649-1-87.

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

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. URL https://openreview.net/forum?id=O9bnihsFfXU.

Saurabh Kumar, Henrik Marklund, and Benjamin Van Roy. Maintaining plasticity in continual learning via regenerative regularization, 2024. URL https://openreview.net/forum?id=lyoOWX0e0O.

Qiyang Li, Aviral Kumar, Ilya Kostrikov, and Sergey Levine. Efficient deep reinforcement learning requires regulating overfitting, 2023a. URL https://arxiv.org/abs/2304.10466.

Sicen Li, Qinyun Tang, Yiming Pang, Xinmeng Ma, and Gang Wang. Realistic actor-critic: A framework for balance between value overestimation and underestimation. *Frontiers in Neurorobotics*, 16, 2023b. ISSN 1662-5218. DOI: 10.3389/fnbot.2022.1081242. URL https://www.frontiersin.org/journals/neurorobotics/articles/10.3389/fnbot.2022.1081242.

Xuhong Li, Haoyi Xiong, Xingjian Li, Xuanyu Wu, Xiao Zhang, Ji Liu, Jiang Bian, and Dejing Dou. Interpretable deep learning: interpretation, interpretability, trustworthiness, and beyond. *Knowledge and Information Systems*, 64(12):3197–3234, December 2022. ISSN 0219-3116.

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.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1509.02971.

Minghuan Liu, Zixuan Chen, Xuxin Cheng, Yandong Ji, Ri-Zhao Qiu, Ruihan Yang, and Xiaolong Wang. Visual whole-body control for legged loco-manipulation. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=cT2N3p1AcE.

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. URL https://openreview.net/forum?id=yr1mzrH3IC.

Clare Lyle, Zeyu Zheng, Khimya Khetarpal, H. V. Hasselt, Razvan Pascanu, James Martens, and Will Dabney. Disentangling the causes of plasticity loss in neural networks. *ArXiv*, abs/2402.18762, 2024. URL https://api.semanticscholar.org/CorpusID:268063557.

Xueguang Lyu, Andrea Baisero, Yuchen Xiao, and Chris Amato. A deeper understanding of state-based critics in multi-agent reinforcement learning. In *AAAI Conference on Artificial Intelligence*, 2022. URL https://api.semanticscholar.org/CorpusID:245669036.

Guozheng Ma, Lu Li, Sen Zhang, Zixuan Liu, Zhen Wang, Yixin Chen, Li Shen, Xueqian Wang, and Dacheng Tao. Revisiting Plasticity in Visual Reinforcement Learning: Data, Modules and Training Stages. October 2023.

Wes McKinney. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. O'Reilly Media, 1 edition, February 2013. ISBN 9789351100065. URL http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/1449319793.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1QRgziT-.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nat.*, 518(7540):529–533, 2015. DOI: 10.1038/NATURE14236. URL https://doi.org/10.1038/nature14236.

Siddharth Mysore, Bassel El Mabsout, Renato Mancuso, and Kate Saenko. Honey. I Shrunk The Actor: A Case Study on Preserving Performance with Smaller Actors in Actor-Critic RL. In *2021 IEEE Conference on Games (CoG)*, pp. 01–08, Copenhagen, Denmark, 2021. IEEE. ISBN 978-1-66543-886-5.

Michal Nauman, Michał Bortkiewicz, Piotr Miłoś, Tomasz Trzciński, Mateusz Ostaszewski, and Marek Cygan. Overestimation, overfitting, and plasticity in actor-critic: the bitter lesson of reinforcement learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *ICML'24*, pp. 37342–37364, Vienna, Austria, July 2024. JMLR.org.

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron Courville. The Primacy Bias in Deep Reinforcement Learning. In *Proceedings of the 39th International Conference on Machine Learning*. PMLR, June 2022.

Travis E. Oliphant. Python for scientific computing. *Computing in Science & Engineering*, 9(3): 10–20, 2007. DOI: 10.1109/MCSE.2007.58.

Georg Ostrovski, Pablo Samuel Castro, and Will Dabney. The difficulty of passive learning in deep reinforcement learning. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=nPHA8fGicZk.

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

Andrei A. Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation, 2016. URL https://arxiv.org/abs/1511.06295.

Thomas Schmied, Thomas Adler, Vihang Patil, Maximilian Beck, Korbinian Pöppel, Johannes Brandstetter, Günter Klambauer, Razvan Pascanu, and Sepp Hochreiter. A Large Recurrent Action Model: xLSTM enables Fast Inference for Robotics Tasks, February 2025.

Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. The dormant neuron phenomenon in deep reinforcement learning. 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. 32145–32168. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/sokar23a.html.

Bhavya Sukhija, Stelian Coros, Andreas Krause, Pieter Abbeel, and Carmelo Sferrazza. Maxinforl: Boosting exploration in reinforcement learning through information gain maximization. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=R4q3cY3kQf.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL http://incompleteideas.net/book/the-book-2nd.html.

Chen Tang, Ben Abbatematteo, Jiaheng Hu, Rohan Chandra, Roberto Martín-Martín, and Peter Stone. Deep Reinforcement Learning for Robotics: A Survey of Real-World Successes. November 2024.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. DeepMind Control Suite, January 2018.

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, November 2020. ISSN 2665-9638.

Twan van Laarhoven. L2 regularization versus batch and weight normalization. *CoRR*, abs/1706.05350, 2017. URL http://arxiv.org/abs/1706.05350.

Guido Van Rossum and Fred L Drake Jr. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

Miguel Vasco, Takuma Seno, Kenta Kawamoto, Kaushik Subramanian, Peter R. Wurman, and Peter Stone. A Super-human Vision-based Reinforcement Learning Agent for Autonomous Racing in Gran Turismo. 2024.

Xuesu Xiao, Bo Liu, Garrett Warnell, and Peter Stone. Motion planning and control for mobile robot navigation using machine learning: a survey. *Autonomous Robots*, 46(5):569–597, June 2022. ISSN 1573-7527.

Zifan Xu, Bo Liu, Xuesu Xiao, Anirudh Nair, and Peter Stone. Benchmarking Reinforcement Learning Techniques for Autonomous Navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9224–9230, May 2023.

Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images, 2020. URL https://openreview.net/forum?id=HklE01BYDB.

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*, 2021. URL https://openreview.net/forum?id=GY6-6sTvGaf.

Supplementary Materials

The following content was not necessarily subject to peer review.

6 Extra results

We include extra results that support the claims made in the main sections, but are not necessary to properly follow the paper.

Figure 7: **Decreasing the size of the actor in DrQ decreases performance (top row) and increases overfitting in the critic**, as measured by o_ϕ (Nauman et al., 2024, bottom row). The solid lines represent mean performance, while the shaded area represents the 95% confidence interval, computed across 10 seeds. In all rows we report values relative to the default baseline.

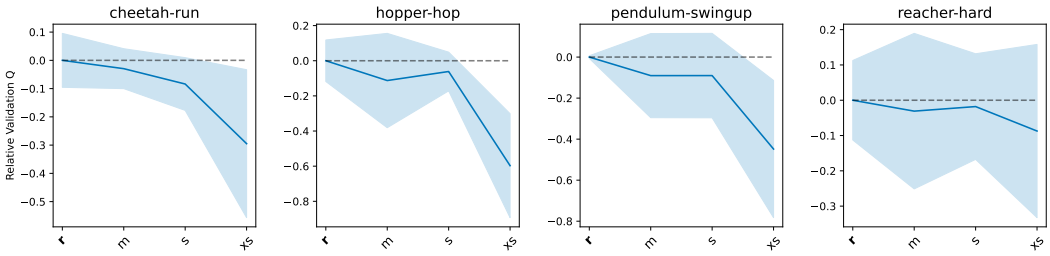

Figure 8: **Decreasing the size of the actor in DrQ decreases validation Q values**, as described in fig. 3. We report Q values at the end of training relative to the default baseline. The solid lines represent mean performance, while the shaded area represents the 95% confidence interval, computed across 10 seeds.

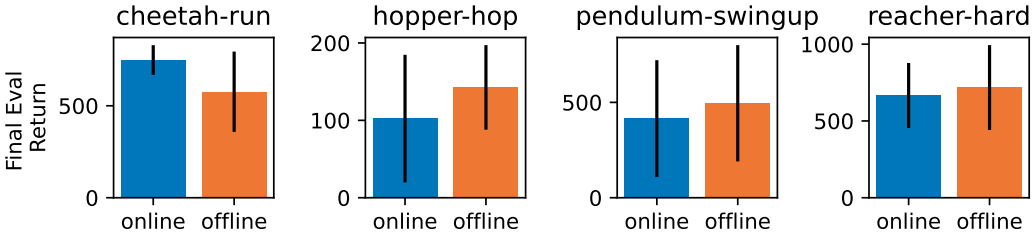

Figure 9: **Training the smallest actor on data collected by the largest, high-performing actor does not appear to lead to a clear improvement in performance in DrQ across a suite of environments,** possibly due to high variance, although an improvement trend might be suggested for hopper-hop and pendulum-swingup. The blue bars are the default final performances of the smallest actors, and the orange bars are the final performances of the smallest actors trained on data collected by a regular-sized actor. Results are aggregated across 10 seeds, and the error bars are the confidence 95% confidence intervals.

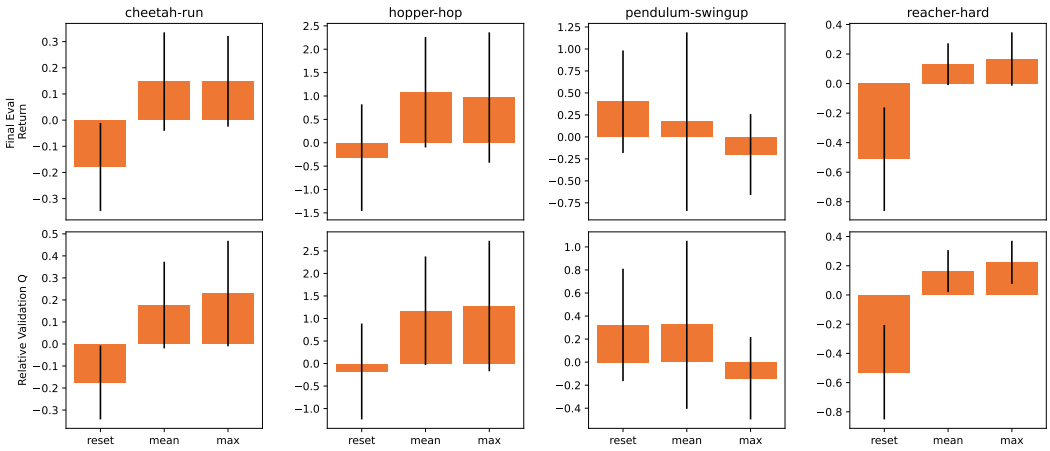

Figure 10: **Impact of various modifications on final performance and validation Q-values in DrQ.** In the resetting experiments, e reset every $100,000$ steps, and only reset the critics. As in (Nikishin et al., 2022), we only reset the MLP of the critics, and leave the encoder untouched. The error bars indicate the 95% confidence interval, computed over 10 seeds.

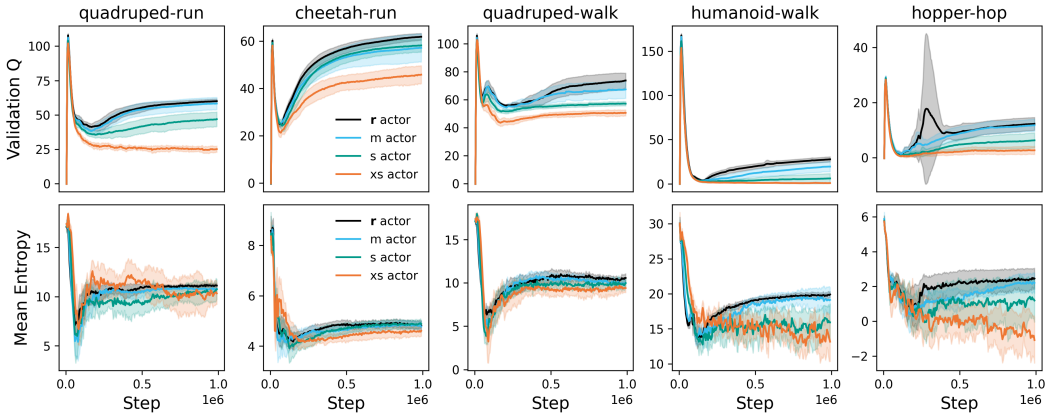

Figure 11: **Decreasing the size of the actor results in Q-value underestimation and reduced policy entropy.** In the top row we estimate the average Q-values on a batch of data gathered during evaluation, and plot the values relative to the baseline **r**. In the bottom row we compute the entropy of the policy π and plot the values relative to the entropy of the baseline **r**. In both cases the solid line represents the mean with shaded areas indicating 95% confidence intervals, computed over 10 independent seeds.

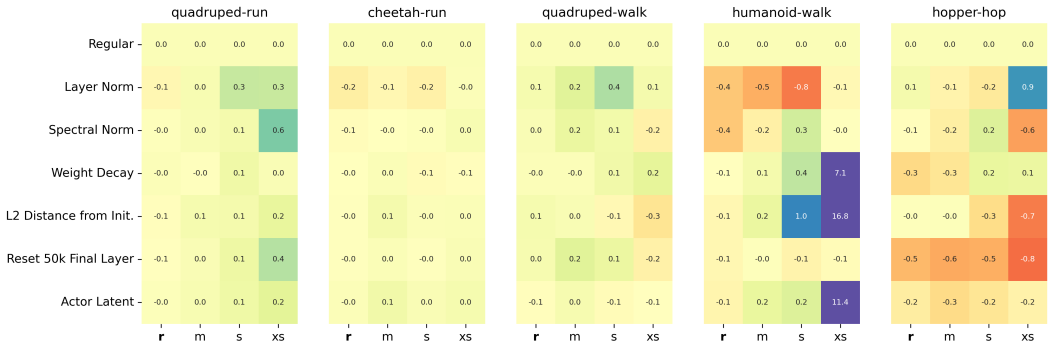

Figure 12: **Impact of critic regularizations on downstream performance with actors of varying sizes.** Each table row corresponds to one of the normalization mechanisms explored, each column indicates the actor size used, and the value in each cell denotes the change relative to the unnormalized version (top row). In most environments there is little change, although in humanoid-walk some regularization techniques do appear to mitigate the performance loss from smaller actors.

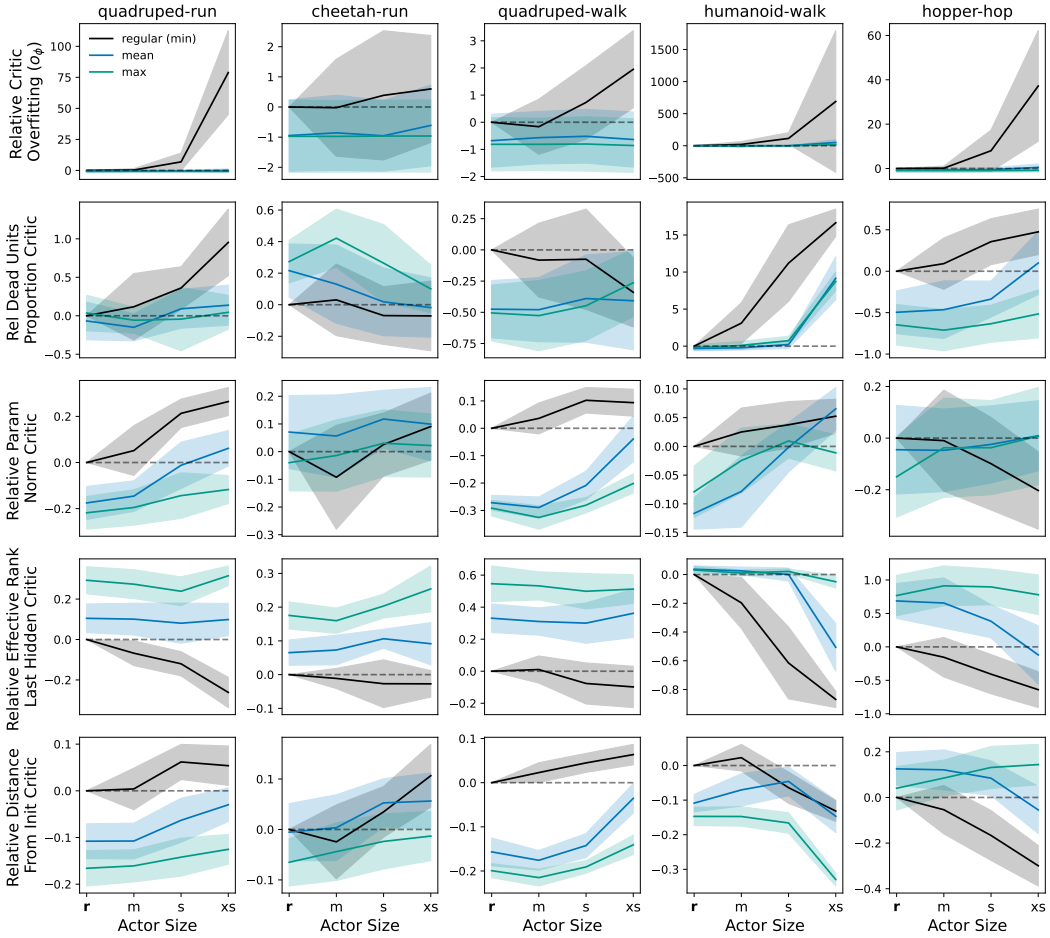

Figure 13: **The impact of small actors on a number of metrics related to plasticity as measured on the critics.** We also evaluate these metrics when using the mean and max of the two critics, as discussed in section 4.1.

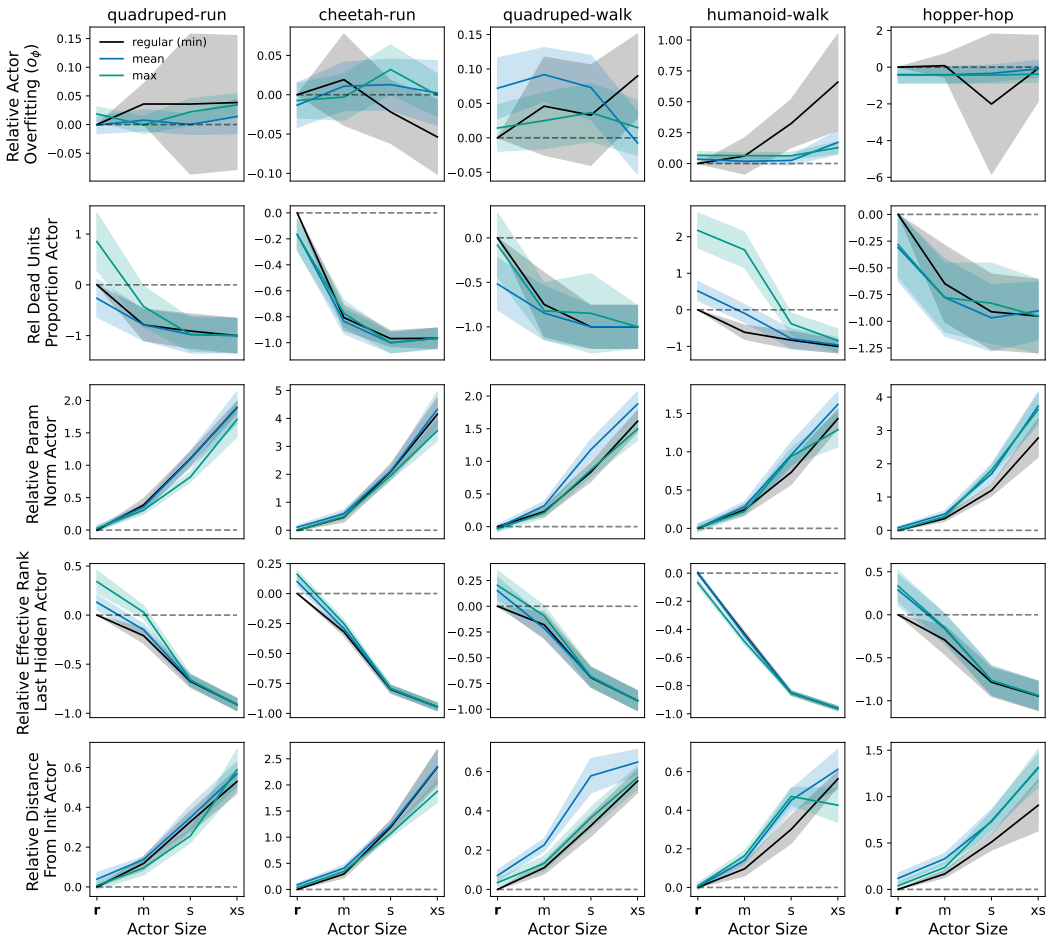

Figure 14: **The impact of small actors on a number of metrics related to plasticity as measured on the actor.** We also evaluate these metrics when using the mean and max of the two critics, as discussed in section 4.1. We define o_ϕ on the actor as $o_\phi := \frac{\mathbb{E}_{\mathcal{D}} H}{\mathbb{E}_{\mathcal{D}_V} H}$, where H is the entropy of the actor's action distribution, and $\mathcal{D}_V$ is a validation dataset.