Revisiting Proximal Policy Optimization

Mahdi Kallel[1]*, **Jose Luis Holgado Alvarez**[1], **Samuele Tosatto**[2], **Carlo D'Eramo**[1,3,4]
[1]Center for Artificial Intelligence and Data Science, University of Würzburg, Germany
[2]Department of Computer Science and Digital Science Center, University of Innsbruck, Austria
[3]Department of Computer Science, TU Darmstadt, Germany
[4]Hessian Center for Artificial Intelligence (Hessian.ai), Germany

Abstract

On-policy Reinforcement Learning (RL) offers desirable features such as stable learning, fewer policy updates, and the ability to evaluate a policy's return during training. While recent efforts have focused on off-policy methods, achieving significant advancements, PPO remains the go-to algorithm for on-policy RL due to its apparent simplicity and effectiveness. Nonetheless, PPO is highly sensitive to hyperparameters and relies on subtle, often poorly documented adjustments that can critically affect its performance, thereby limiting its utility in complex scenarios. In this paper, we revisit the PPO algorithm by introducing principled enhancements that improve performance while eliminating the need for extensive hyperparameter tuning and implementation-specific optimizations. Our proposed approach, PPO+, is a principled adaptation of the PPO algorithm that strengthens its adherence to the on-policy objective, enhancing stability and efficiency. PPO+ demonstrates significantly improved asymptotic performance over PPO, and a substantially reduced performance gap with off-policy algorithms in several challenging continuous control tasks. Beyond just performance, our findings offer a fresh perspective on on-policy RL.

1 Introduction

A fundamental distinction in Reinforcement Learning (RL) lies between on-policy and off-policy methods (Sutton & Barto, 2018). On-policy methods, such as Proximal Policy Optimization (PPO) (Schulman et al., 2017b) and Trust-Region Policy Optimization (TRPO) (Schulman et al., 2015a), directly optimize the expected reward under the current policy's state-action distribution, improving the policy while actively interacting with the environment. This leads to stable learning and safer exploration since the policy stays close to the data distribution it learns from, though at the cost of potentially reduced sample-efficiency. In contrast, off-policy methods optimize the expected reward under a different distribution—often using an exploration or behavior policy. By leveraging data generated from different policies, off-policy methods can reuse past experiences, boosting sample-efficiency. This flexibility supports more aggressive exploration, making off-policy methods more suitable when data collection is expensive or restricted.

Recent advancements in off-policy approaches, such as Soft Actor-Critic (SAC) (Haarnoja et al., 2018) and Twin Delayed Deep Deterministic Policy Gradients (TD3) (Fujimoto et al., 2018), have significantly improved continuous control on complex tasks. However, on-policy algorithms have not kept pace in terms of asymptotic performance and sample-efficiency. While PPO remains a dominant choice in on-policy RL, delivering impressive results across a range of applications (Berner et al., 2019; Andrychowicz et al., 2020b; Mirhoseini et al., 2021; Rudin et al., 2022), it is hindered by the complexity of its inherent mechanisms, including trust-region optimization, multiple loss functions,

18th European Workshop on Reinforcement Learning (EWRL 2025).

*Correspondence to `mahdi.kallel@uni-wuerzburg.de`.

and various implementation-specific optimizations, making it highly sensitive to hyperparameter tuning (Andrychowicz et al., 2020a; Huang et al., 2022a).

Moreover, common practices in the use of PPO have crucial shortcomings. For example, despite the empirically demonstrated success of maximum entropy RL (Haarnoja et al., 2018; Bhatt et al., 2024) and theoretical works suggesting it can enhance the convergence of policy gradient methods (Mei et al., 2020; Cen et al., 2024), its application for on-policy deep RL remains underexplored.

Additionally, many widely adopted PPO implementations neglect the usage of explicit action bounds by defaulting to standard Gaussian policies rather than their bounded counterparts. This technical oversight not only deviates from theoretical rigor but, as we will demonstrate, can also result in degenerate learning behaviors in certain environments. These challenges, combined with the inherent complexity of current on-policy deep RL methods, motivate us to pursue simpler and more sample-efficient alternatives.

In this paper, we introduce PPO+, a principled enhancement of the PPO algorithm that introduce targeted solutions to tackle PPO's drawbacks while eliminating the need of extensive hyperparameter tuning and subtle implementation-level optimizations. Our primary objective is to significantly advance PPO's capabilities adhering to the on-policy paradigm, focusing on improving its stability, performance, and simplicity, rather than to directly compete with the sample efficiency of fundamentally different off-policy algorithms.

More concretely, our **contribution** is threefold. (i.) We propose and demonstrate that leveraging off-policy data can significantly improve critic learning while preserving the on-policy formulation of the policy gradient. (ii.) We develop a numerical scheme to integrate action bounds in continuous control with PPO. (iii.) We reformulate the PPO optimization problem under the maximum entropy RL perspective for enhanced exploration. We demonstrate that PPO+ achieves state-of-the-art performance on MuJoCo Todorov et al. (2012) control problems among on-policy methods for continuous control while maintaining a simple and straightforward implementation and being closely aligned with the theoretical foundations of on-policy RL. A discussion of related literature is provided in Appendix A.

2 Background

2.1 On-policy reinforcement learning

Reinforcement Learning (RL) (Sutton & Barto, 2018) deals with the problem of an agent interacting with an environment to learn a policy that maximizes its return. Mathematically, an RL problem can be formulated as a Markov Decision Process (MDP) (Puterman, 1990), which is a tuple $\langle S, A, P, R, \mu_0, \gamma \rangle$, where $S \in \mathbb{R}^m$ is a continuous set of states and $A \in \mathbb{R}^d$ is a continuous set of actions. $P : S \times A \to \Delta S$ is the transition probability function[2], where $P(s'|s,a)$ denotes the probability of transitioning to state s' after taking action a in state s. $R : S \times A \to \mathbb{R}$ is the reward function, where $r(s,a)$ is the immediate reward received by the agent for taking action a in state s. $\mu_0 \in \Delta S$ is the initial state distribution. $\gamma \in [0,1)$ is the discount factor, which determines the importance of future rewards compared to immediate rewards.

In on-policy RL, the agent's goal is to learn a stochastic policy $\pi : S \to \Delta A$, that maximizes its expected discounted return $J(\pi) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\gamma^\pi, a\sim\pi}\left[r(s,a)\right]$, where we denote $d_\gamma^\pi(s) \triangleq (1-\gamma)\sum_{t=0}^{\infty}\gamma^t P(s_t = s)$ the discounted state visitation density of the state s under the policy π. This is in contrast to off-policy RL where the objective of the agent is to maximize the policy return under a different behaviour policy $\beta(a|s) \neq \pi(a|s)$ making the objective to maximize $J^\beta(\pi) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim\rho_\gamma^\beta, a\sim\pi}\left[r(s,a)\right]$.

2.2 Maximum entropy reinforcement learning

Traditional RL algorithms focus solely on maximizing the expected reward, which can yield overly deterministic policies vulnerable to unexpected environmental changes. Maximum entropy RL (Ziebart, 2010; Haarnoja et al., 2018) counters this by adding an entropy bonus to the objective. The policy entropy, a measure of randomness, is defined as $H(\pi(.|s)) = -\sum_a \pi(a|s)\log\pi(a|s)$

[2] ΔX denotes the set of probability measures over a set X.

Incorporating this bonus encourages exploration without sacrificing return, achieved by introducing a temperature parameter α and reformulating the objective as

$$J(\pi) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t \left(r(s_t, a_t) + \alpha H(\pi(.|s_t))\right)\right]. \tag{1}$$

The temperature α controls the trade-off between maximizing reward and entropy. A larger α leads to a greater emphasis on exploration and mode diversity in the policy. In practice, we observe that it considerably improves exploration and hence learning speed over state-of-art methods that optimize the conventional RL objective function (Schulman et al., 2017a).

2.3 Trust-region methods

Initially introduced by Schulman et al. (2015a), trust region deep RL methods are on-policy algorithms that optimize a surrogate objective by maximizing a lower bound on the policy return. Trust Region Policy Optimization (TRPO) constrains the policy update by limiting the KL divergence between the new policy π' and the old policy π, ensuring updates remain within a "trusted region" for stable learning. However, TRPO's approach originally relied on a heuristic to enforce this constraint. In Achiam et al. (2017), the authors formalized this heuristic by bounding the difference between the returns of two policies, π' and π, as follows

$$J(\pi') - J(\pi) \geq \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi}, a\sim\pi'}\left[A^{\pi}(s,a)\right] - \frac{2\gamma\epsilon^{\pi'}}{1-\gamma}\sqrt{\frac{1}{2}\mathbb{E}_{s\sim d^{\pi}}\left[D_{KL}(\pi'\|\pi)[s]\right]}, \tag{2}$$

where $\epsilon^{\pi'} \doteq \max_s \left|\mathbb{E}_{a\sim\pi'}\left[A^{\pi}(s,a)\right]\right|$.

By squaring the penalty term and applying the importance sampling trick to replace the expectation over $a \sim \pi'$ with $a \sim \pi$, this optimization problem can be rewritten as

$$\text{maximize}_{\pi'} \mathbb{E}_{s\sim d^{\pi}, a\sim\pi}\left[\frac{\pi'(a|s)}{\pi(a|s)}A^{\pi}(s,a)\right] \tag{3}$$

$$\text{subject to} \quad \mathbb{E}_{s\sim d^{\pi}}\left[D_{KL}(\pi'\|\pi)[s]\right] \leq \delta. \tag{4}$$

TRPO solves this optimization problem by approximating the KL divergence constraint using a second-order method involving the Fisher information matrix, which requires a conjugate gradient method for optimization. While this guarantees updates stay within a trusted region, making the learning process stable, it also makes the algorithm computationally expensive due to the need for calculating the Fisher information matrix and solving the constrained optimization.

To address this complexity, Schulman et al. (2017b) propose Proximal Policy Optimization (PPO), which simplifies the enforcement of the trust region by introducing a clipping mechanism. Instead of explicitly constraining the KL divergence, PPO limits the probability ratio between the new and old policies, ensuring updates remain moderate. This approach is simpler to implement and significantly reduces computational overhead while retaining stable learning performance.

2.4 Actor-critic methods

Actor-critic methods are a class of RL algorithms comprising an actor and a critic. The critic estimates policy performance using either the action-value function $Q^{\pi}(s,a) \triangleq \mathbb{E}\left[r(s,a) + \gamma Q^{\pi}(s',a') \mid s_0 = s, a_0 = a\right]$ or the value function $V^{\pi}(s) \triangleq \mathbb{E}_{a\sim\pi(s)}\left[Q^{\pi}(s,a)\right]$. The actor updates its parameters to maximize the policy return as evaluated by the critic, yielding more efficient learning than methods relying solely on Monte-Carlo estimates.

These algorithms typically employ neural networks and optimize via gradient ascent Sutton et al. (1999). Temporal Difference (TD) learning (Sutton, 1988; Sutton & Barto, 2018) provides an iterative method to estimate Q^{π} using the TD error $\delta_t = r_{t+1} + \gamma Q^{\pi}(s_{t+1}, a_{t+1}) - Q^{\pi}(s_t, a_t)$, where r_{t+1} is the reward, γ the discount factor, and (s_t, a_t) and (s_{t+1}, a_{t+1}) denote successive state-action pairs. Traditionally, the critic is trained on-policy (e.g., via SARSA (Sutton & Barto, 2018)); alternatively, it can be updated off-policy using previously collected data as in DDPG, TD3, or SAC (Haarnoja et al., 2018; Fujimoto et al., 2018).

3 On the limitations of proximal policy optimization

3.1 Critic sensitivity

Despite its widespread use (Berner et al., 2019; Andrychowicz et al., 2020b; Mirhoseini et al., 2021; Rudin et al., 2022), PPO's sensitivity to certain parameters and implementation details highlights serious shortcomings in the algorithm, many of which can be traced back to reliance on inaccurate value function estimators.

Figure 1 shows empirical evidence of these limitations. For starters, normalization of rewards or advantage functions plays a critical role in stabilizing PPO's learning process. Without it, PPO often fails to learn effective policies, especially in environments where rewards differ in magnitude, such as Hopper or Walker2d. Also, the algorithm is very sensitive to observation normalization. As a reminder, most off-policy methods can function effectively without such techniques (Haarnoja et al., 2018).

Furthermore, PPO exhibits performance degradation when the λ parameters for the Generalized Advantage Estimator (GAE) Schulman et al. (2015b) is set to values lower than its standard 0.95. The later essentially reducing the advantage estimate towards Monte-Carlo and hence reducing the influence of the critic. As we decrease the λ value, the reliance on bootstrapping from the value function increases, and the algorithm's performance suffers significantly. This sensitivity indicates that PPO works optimally only in regimes of high λ, which directly points to a strong dependence on, and potential limitations arising from, the quality of the value estimates produced by the critic. Indeed, when relying solely on the critic's instantaneous estimates (i.e., $\lambda = 0$), the algorithm demonstrably fails to learn.

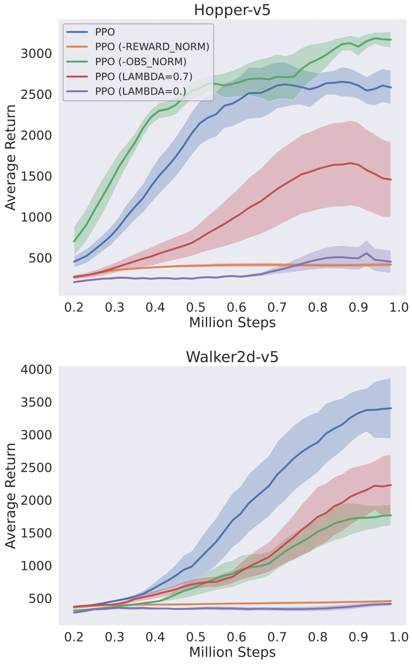
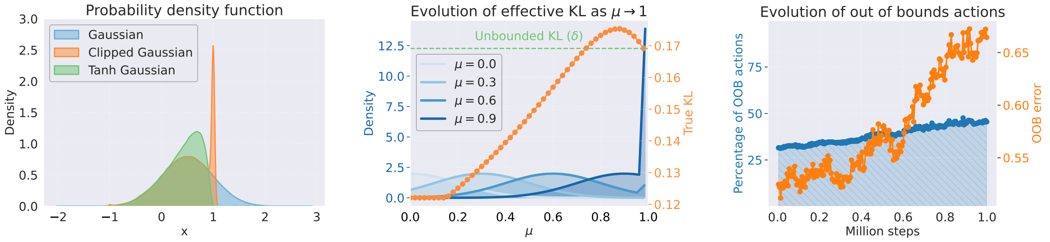


Figure 1: Sensitivity of PPO to reward normalization (`- Reward norm`), Observation normalization (`-Obs norm`) and GAE-λ (`LAMBDA={0.,0.7}`).

These limitations suggest that further improvements are possible, potentially enhancing the performance of deep on-policy methods.

3.2 Consequences of unbounded actions

One notable inconsistency in PPO for continuous control is that the actor samples from an unbounded distribution, even when the environment requires bounded actions. In Figure 2a, we compare the probability density functions of a standard Gaussian $x \sim \mathcal{N}(0, 0.5)$, its bounded variant, and the tanh-bounded version. The bounded Gaussian shows a notably higher density near the boundaries, whereas the tanh-bounded version yields a smoother distribution. In Figure 2b, we generate a pair of Gaussian distributions with identical standard deviations and a slight difference in means such that their KL divergence remains constant. We empirically demonstrate that as the means approach the action bounds to the right, the KL divergence between the clipped versions increases rapidly, potentially leading to learning instability. Figure 2c reports both the average out-of-bound error and the proportion of actions exceeding the prescribed limits. Notably, during PPO training, the frequency and magnitude of out-of-bound actions increase over time. This divergence—where the effective clipping ratio deviates from the intended ratio—can negatively impact learning by causing the critic to receive inaccurate action feedback (e.g., the critic may interpret an action as $a = 2$ when the environment sees it as $a = 1$).



(a)

(b)

(c)

Figure 2: 2a Probability density functions of a Gaussian with $x \sim \mathcal{N}(\mu = 0, \sigma = 0.5)$. 2b Average out of bound error for the actions, and percentage of out of bound actions. 2c Logarithm of the effective clipping ratio for two bounded Gaussians $\mathcal{N}_1, \mathcal{N}_2$ s.t. $KL(\mathcal{N}_2|\mathcal{N}_1) = 0.2$.

4 Enhancing proximal policy optimization

From previous remarks, many practices in PPO raise questions about its adherence to its on-policy objective. To address this, we investigate several improvements aimed at aligning PPO more closely with its theoretical foundations, leading to the development of our novel algorithm PPO+, whose pseudocode is provided in Algorithm 1.

4.1 Bounding the action space

Building upon our prior findings, a naive attempt to address the out of bounds actions issue is to directly applying a squashing function, such as the hyperbolic tangent. However, this approach leads to frequent training collapses and numerical instability in the computation of $\log \pi'(a \mid s)$, i.e., the log-probability of an action under the squashed Gaussian policy.

More formally, note that in a tanh-Gaussian policy, actions are generated as $a = \tanh(u)$, where $u \sim \mathcal{N}(\mu, \Sigma)$. By the change-of-variables formula, the density of a is given by $\pi(a \mid s) = \mu(u \mid s) \left|\det \frac{du}{da}\right|$, which expands to

$$\log \pi(a \mid s) = \log \mu(u \mid s) - \sum_{i=1}^{D} \log(1 - \tanh^2(u_i)).$$

If u is unavailable, one must recover it via $u = \operatorname{atanh}(a)$, but this introduces numerical instability when a is near ± 1, as $\operatorname{atanh}(a)$ diverges. Additionally, the naive correction term $\log(1 - \tanh^2(u))$ becomes unstable for large $|u|$, since $\tanh(u)$ saturates and $1 - \tanh^2(u)$ approaches zero.

In contrast, Soft Actor-Critic (SAC)-style implementations avoid these issues by computing the sample and log-probability simultaneously, caching the pre-tanh sample u for direct use. Since this joint computation is not directly feasible in our framework, we instead store the unsquashed Gaussian samples u and apply a numerically stable transformation, replacing the $\log(1 - \tanh^2(u))$ term with $2\Big(\log 2 - u - \operatorname{softplus}(-2u)\Big)$.[3]

This formulation avoids both the instability of direct logarithm computation on small values and the need for $\operatorname{atanh}(a)$, ensuring robust log-probability estimates and improving gradient quality, leading to more stable training in continuous control tasks.

4.2 Off-policy critic learning

Inspired by successful off-policy critic training in algorithms like SAC and TD3 (Haarnoja et al., 2018; Fujimoto et al., 2018), we explore a novel on-policy RL strategy. We train the critic of algorithms like PPO using off-policy data from a replay buffer, while crucially maintaining the on-policy actor update—a combination not previously explored to our knowledge in standard on-policy methods.

[3] This expression is mathematically equivalent to $2\Big(\log 2 + u - \operatorname{softplus}(2u)\Big)$.

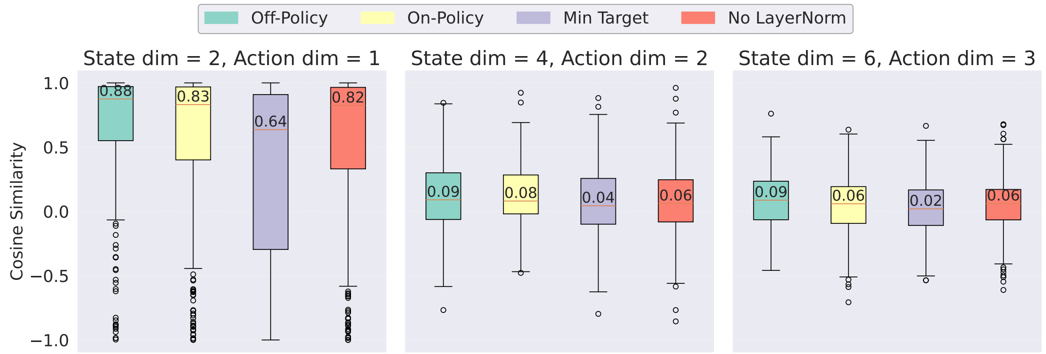


Figure 3: Distribution of cosine similarity between estimated gradients and the true policy gradient, $\mathbb{E}_{s\sim d^\pi}\nabla Q^\pi$, across different critic configurations in LQR environments of varying state dimensions. **Left:** 2D LQR, **Middle:** 8D LQR, **Right:** 16D LQR. Metrics are averaged over 10 independent runs.

We hypothesize that incorporating off-policy data to enhance critic approximation can lead to more accurate on-policy gradient estimates, potentially improving performance. To enable the desired use of off-policy data for the critic, we need to replace the value function V^π typically used in on-policy methods with action-value functions Q^π usually restricted to off-policy methods. Indeed, recall that standard temporal-difference (TD) learning with state-value functions (V-functions) is inherently on-policy. The update target for $V(s)$ relies on $V(s')$, implicitly assuming the action leading to s' was from the current policy π. In contrast, Q-functions allow the action 'a' in the transition (s, a, r, s') to originate from *any* policy, enabling the use of replay buffers. When needed, one can recover the value function as $V^\pi(s) = \mathbb{E}_{a\sim\pi(s)}\left[Q^\pi(s, a)\right]$.

In Figure 3, we experimentally validate our central hypothesis: *training critics with off-policy data enhances the estimation of the on-policy Q-function (Q^π) and its gradient.* We utilize Linear Quadratic Regulator (LQR) environments, chosen for their analytical tractability, allowing direct comparison against the true Q-function and gradient. To isolate the impact of different critic training strategies, agents were equipped with multiple critics (defaulting to two trained with off-policy data) measured on the same data distribution. We systematically varied configurations to include `On-Policy` critic training, `Minimum Target` bootstrapping, and training with `No LayerNorm`(details in Appendix D). For each configuration, 5000 on-policy interactions were collected and used for 5000 critic TD updates, with metrics evaluated every $20,000$ interactions.

Our results reveal several key insights. First, training critics using off-policy data (`off-policy`) significantly improves the quality of gradient estimates compared to using only on-policy data (`on-policy`). Second, contrary to the minimum target heuristic often employed in actor-critic methods, using the minimum prediction from twin critics (`min-target`) actually reduces the quality of the policy gradient estimate in our setting. This is evidenced by the lower cosine similarity of gradients from minimum-target critics compared to independently trained off-policy critics. Third, removing LayerNorm (`no layernorm` vs. Default) seems to degrade the quality of the gradient estimate Xu et al. (2019). These experimental findings strongly support our hypothesis that off-policy data enhances critic training, leading to improved Q-function and hence policy gradient estimation. Furthermore, our ablation study highlights that while off-policy data is beneficial, techniques like minimum-target critics may not universally improve gradient quality and can even be detrimental in certain contexts, emphasizing the importance of careful critic design.

4.3 Maximum entropy for on-policy reinforcement learning

Maximum entropy RL augments the standard objective with an entropy term to encourage exploration, a strategy proven effective in off-policy settings (Haarnoja et al., 2018). While its application to on-policy RL is less explored, no inherent limitations prevent its use. We incorporate the maximum entropy objective into PPO+ by modifying the advantage function, drawing inspiration from SAC's critic update (Haarnoja et al., 2018).

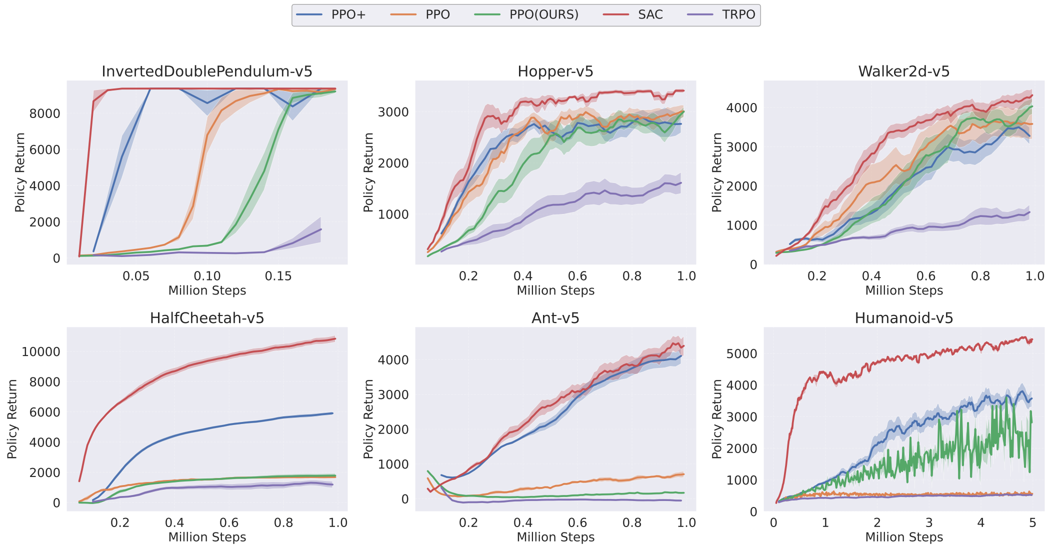


Figure 4: Undiscounted policy return on classic MuJoCo-v5 tasks.We use 10 random seeds for every algorithm and show the standard deviation.

Specifically, the critic learns an entropy-augmented Q-function. The advantage function $A(s, a)$ is then derived from this Q-function by including an instantaneous entropy term $\alpha \log \pi(a|s)$ and adjusting the baseline value estimate accordingly, as shown in Equation (5):

$$A(s, a) = Q(s, a) - \alpha \log \pi(a|s) - \mathbb{E}_{a' \sim \pi(.|s)}[Q(s, a') - \alpha \log \pi(a'|s)]. \tag{5}$$

The actor is subsequently updated using the standard PPO objective, but with this entropy-regularized advantage $\hat{A}^\pi$. Thus, the entropy bonus is incorporated into the value estimation component of the policy gradient, rather than as an explicit, separate entropy term in the PPO objective function itself. This implicitly encourages entropy maximization through the critic's evaluation of actions.

4.4 Benchmarks

We conduct a comprehensive empirical evaluation of PPO+ against related on-policy baselines PPO and TRPO. We also include Soft Actor-Critic (SAC), a leading off-policy algorithm, as a strong benchmark to contextualize the performance of PPO+ within the broader RL landscape. For all the algorithms using the maximum entropy we use automatic temperature tuning simlar to (Haarnoja et al., 2018). The evaluation is performed on the MuJoCo benchmark suite for continuous control (Todorov et al., 2012) and on high-dimensional control problems such as the Dog (stand, walk, trot, and run) tasks from DM-Control (Tunyasuvunakool et al., 2020). Futhermore, our baseline implementations are based on (Huang et al., 2022b). To ensure fairness of comparaison, we introduce PPO(ours) as a variant of PPO that uses a similar configuration to PPO+.

Consistent with the findings of our ablation study, we use two critic networks trained independently, without employing a minimum target approach. Key implementation differences between PPO+ and PPO are summarized in Table 1 and detailed hyperparameter settings for all baselines are provided in Table 2 in the Appendix.

The benchmark results, illustrated in Figure 4 illustrates that PPO+attains performance comparable to PPO on simpler tasks such as Hopper and Walker. Notably, PPO+significantly outperforms PPO on more complex, high-dimensional tasks like Ant and Humanoid, demonstrating a clear performance advantage. We attribute this widening gap to the limitations of PPO's advantage estimation; specifically, with a high $\lambda = 0.95$, its advantage estimates increasingly resemble those of REINFORCE with a baseline—a method known to be less effective in high-dimensional environments. Although PPO+lags slightly behind SAC on the Mujoco tasks, particularly for Humanoid, Figure 5 reveals that our algorithm performs significantly better on the high-dimensional DOG tasks.

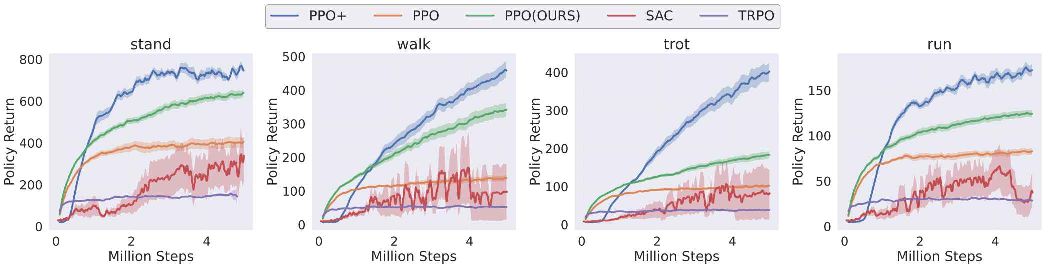


Figure 5: Undiscounted policy return on different tasks of the Dog environment of Deepmind Control Suite. We use 10 random seeds for every algorithm and show the standard deviation.

In summary, PPO+ consistently outperforms on-policy algorithms like PPO and TRPO. Moreover, it significantly closes the performance gap to leading off-policy methods such as SAC, especially in complex, high-dimensional control tasks. PPO+ achieves this performance advantage over PPO without relying on any implementation-level optimizations, showcasing the robustness of our approach.

4.5 On the impact of PPO+ enhancements

In Figure 6, we present an ablation study for PPO+ examining our three key design choices: (1) Using bounded policies; (2) the use of off-policy data for training the critic; and (3) the use of the maximum entropy objective.

First, the use of bounded action policies, a key PPO+ modification, demonstrated varied impact in ablation studies. While bounded actions clearly improved performance on tasks like Walker2d-v5 and HalfCheetah-v5, the effect was less pronounced on Hopper-v5 and Ant-v5. Notably, unbounded actions yielded better results on Humanoid-v5. This suggests that while bounded actions are theoretically important for policy coherence and preventing critic inaccuracies, their benefits can be task-dependent, potentially interacting with exploration dynamics. The benefit for the other tasks and the fundamental need for coherence lead us to consider this modification as overall beneficial.

Second, results demonstrate that restricting the critic's training to on-policy data significantly degrades performance across most tasks, particularly impeding learning on Humanoid. This is contrary to the common practice of restricting training the critic to on-policy data for on-policy gradient methods. We further explore how much off-policy data is beneficial in Figure 8 in the appendix. Interestingly, while the use of off-policy data is already well explored in deep off-policy actor-critic methods like Lillicrap (2015); Haarnoja et al. (2018); Fujimoto et al. (2018), it was previously unclear to what extent this choice would benefit on-policy algorithms, especially in terms of actor and critic performance individually. This work strongly suggests that at least the critic derives a substantial benefit from the use of off-policy data, indicating that deep neural networks can effectively overcome the state aliasing inherent to off-policy TD learning Sutton et al. (2016).

Finally, regarding the maximum entropy objective, we find that it consistently enhances our performance consistently across all tasks. While PPO can achieve reasonable performance even without an entropy bonus, we believe that the poor quality of its critic may unintentionally provide some implicit

Table 1: Key implementation differences between PPO and PPO+.

Attribute	PPO	PPO+
Observation normalization	✓	-
Reward normalization	✓	-
Advantage normalization	✓	-
Learning rate scheduler	✓	-
Clip critic target	✓	-
GAE-λ critic	✓	✓
Bounded actor policy	-	✓
Uses off-policy data	-	✓
Maximum entropy objective	-	✓

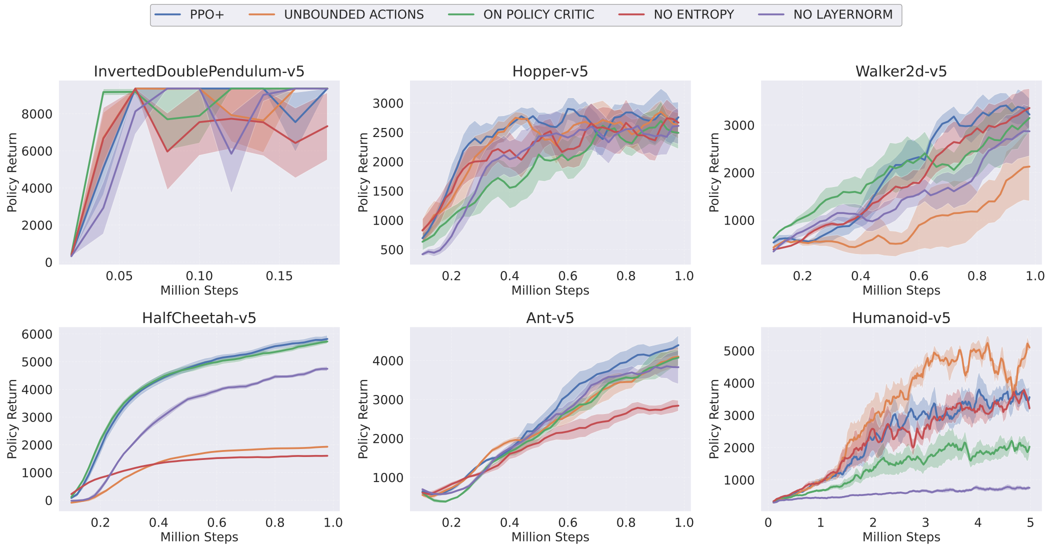


Figure 6: Evolution of the policy return on the MuJoCo-v5 tasks for various design choices. We use 5 random seeds for every algorithm. (`ON-POLICY CRITIC`) restricts the critics to using only on-policy data. (`UNBOUNDED ACTIONS`) uses an unbounded Gaussian distribution for the actor. (`NO ENTROPY`) removes the entropy bonus from the critics. PPO+ (`NO LAYERNORM`) discards using LayerNorm for the actor and critic. Further ablations are provided in Figure 7 in the Appendix.

exploration. Adding an explicit entropy bonus, as in PPO+, provides a more principled exploration strategy. Furthermore, LayerNorm is confirmed to be crucial to stabilize learning, particularly and dramatically for the Humanoid task, where its absence leads to significant performance drops.

Overall these results are in perfect agreement with our previous results on the impact of these design choices on the quality of the policy gradient demonstrated in Figure 3. In conclusion, our results strongly suggest that (1) off-policy TD learning definitively allows for better on-policy gradient estimation and consistently enhances performance across various tasks; (2) bounding the distribution of the policy is indeed necessary to ensure PPO's coherence and further boosts performance, especially in more complex environments ; (3) the entropy bonus demonstrably provides a clear and consistent benefit to PPO+ as opposed to PPO which is indifferent to the entropy bonus (albeit a slightly different one), as reported in Andrychowicz et al. (2020a). We also demonstrate that PPO+ exhibits improved robustness to the GAE-λ parameter compared to standard PPO, indicating a more reliable critic (see Appendix D).

5 Discussion and conclusion

We introduced PPO+ , a methodical enhancement of the Proximal Policy Optimization (PPO) algorithm that adheres to on-policy reinforcement learning (RL) principles with a simple, trick-free implementation. PPO+ introduces three key improvements over PPO: training the critic with off-policy data while preserving the on-policy policy gradient, bounding the policy distribution, and employing maximum entropy exploration. By focusing on the quality of the critic approximation, PPO+ avoids complex critic learning schemes and achieves state-of-the-art performance in MuJoCo locomotion tasks. Thanks to its simplicity and rigorous formulation, we believe that PPO+ offers an accessible and solid ground for future research in on-policy deep RL.

Limitations. Despite its strengths, PPO+ does not match the sample efficiency of off-policy competitors (e.g., SAC (Haarnoja et al., 2018)). Nevertheless, we hope its simplicity and the insights provided will inspire further interest in on-policy methods. Future work could explore better strategies for correcting the off-policy distribution during critic learning—potentially integrating such corrections into actor updates—or focus on enhancing critic learning through improved validation criteria (Kallel et al., 2024) or increased neuroplasticity (Nikishin et al., 2022).

Acknowledgments

This work was funded by the German Federal Ministry of Education and Research (BMBF) (Project: 01IS22078). The authors gratefully acknowledge the scientific support and HPC resources provided by the Erlangen National High Performance Computing Center (NHR@FAU) of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) under the NHR project b187cb. NHR funding is provided by federal and Bavarian state authorities. NHR@FAU hardware is partially funded by the German Research Foundation (DFG) – 440719683.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International conference on machine learning*, pp. 22–31. PMLR, 2017.

Zafarali Ahmed, Nicolas Le Roux, Mohammad Norouzi, and Dale Schuurmans. Understanding the impact of entropy on policy optimization. In *International conference on machine learning*, pp. 151–160. PMLR, 2019.

Marcin Andrychowicz, Anton Raichuk, Piotr Stańczyk, Manu Orsini, Sertan Girgin, Raphaël Marinier, Leonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, et al. What matters for on-policy deep actor-critic methods? a large-scale study. In *International conference on learning representations*, 2020a.

OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020b.

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

Leemon Baird. Residual algorithms: reinforcement learning with function approximation. In *machine learning proceedings 1995*, pp. 30–37. Elsevier, 1995.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

Aditya Bhatt, Daniel Palenicek, Boris Belousov, Max Argus, Artemij Amiranashvili, Thomas Brox, and Jan Peters. Crossq: Batch normalization in deep reinforcement learning for greater sample efficiency and simplicity. *arXiv preprint arXiv:1902.05605*, 2024.

Shicong Cen, Yuting Wei, and Yuejie Chi. Fast policy extragradient methods for competitive games with entropy regularization. *Journal of machine learning Research*, 25(4):1–48, 2024.

Jean Seong Bjorn Choe and Jong-Kook Kim. Maximum entropy on-policy actor-critic via entropy advantage estimation. *arXiv preprint arXiv:2407.18143*, 2024.

Peter Dayan. The convergence of td (λ) for general λ. *machine learning*, 8:341–362, 1992.

Thomas Degris, Martha White, and Richard S Sutton. Off-policy actor-critic. *arXiv preprint arXiv:1205.4839*, 2012.

Raghuram Bharadwaj Diddigi, Chandramouli Kamanchi, and Shalabh Bhatnagar. A convergent off-policy temporal difference algorithm. *arXiv preprint arXiv:1911.05697*, 2019.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

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

Matteo Gallici, Mattie Fellows, Benjamin Ellis, Bartomeu Pou, Ivan Masmitja, Jakob Nicolaus Foerster, and Mario Martin. Simplifying deep temporal difference learning. *arXiv preprint arXiv:2407.04811*, 2024.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Shengyi Huang, Rousslan Fernand Julien Dossa, Antonin Raffin, Anssi Kanervisto, and Weixun Wang. The 37 implementation details of proximal policy optimization. *The ICLR Blog Track 2023*, 2022a.

Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and João G.M. Araújo. Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23(274):1–18, 2022b. URL `http://jmlr.org/papers/v23/21-1342.html%7D`.

Tommi Jaakkola, Michael Jordan, and Satinder Singh. Convergence of stochastic iterative dynamic programming algorithms. *Advances in neural information processing systems*, 6, 1993.

Mahdi Kallel, Debabrota Basu, Riad Akrour, and Carlo D'Eramo. Augmented bayesian policy search. In *The Twelfth International Conference on Learning Representations*, 2024.

J Kolter. The fixed points of off-policy td. *Advances in neural information processing systems*, 24, 2011.

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in neural information processing systems*, 32, 2019.

TP Lillicrap. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Jincheng Mei, Chenjun Xiao, Csaba Szepesvari, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods. In *International conference on machine learning*, pp. 6820–6829. PMLR, 2020.

Azalia Mirhoseini, Anna Goldie, Mustafa Yazgan, Joe Wenjie Jiang, Ebrahim Songhori, Shen Wang, Young-Joon Lee, Eric Johnson, Omkar Pathak, Azade Nazi, et al. A graph placement methodology for fast chip design. *Nature*, 594(7862):207–212, 2021.

Volodymyr Mnih. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron Courville. The primacy bias in deep reinforcement learning. In *International conference on machine learning*, pp. 16828–16847. PMLR, 2022.

Martin L Puterman. Markov decision processes. *Handbooks in operations research and management science*, 2:331–434, 1990.

Nikita Rudin, David Hoeller, Marko Bjelonic, and Marco Hutter. Advanced skills by learning locomotion and local navigation end-to-end. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2497–2503. IEEE, 2022.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015a.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015b.

John Schulman, Xi Chen, and Pieter Abbeel. Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*, 2017a.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017b.

Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3: 9–44, 1988.

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

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

Richard S Sutton, A Rupam Mahmood, and Martha White. An emphatic approach to the problem of off-policy temporal-difference learning. *Journal of machine learning Research*, 17(73):1–29, 2016.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

John Tsitsiklis and Benjamin Van Roy. Analysis of temporal-diffference learning with function approximation. *Advances in neural information processing systems*, 9, 1996.

Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, Nicolas Heess, and Yuval Tassa. dm$_{c}ontrol : Softwareandtasksforcontinuouscontrol$. *Software Impacts*$,6 : 100022, 2020. ISSN2665-9638. DOI : .$ URL `https://www.sciencedirect.com/science/article/pii/S2665963820300099`.

Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

Jingjing Xu, Xu Sun, Zhiyuan Zhang, Guangxiang Zhao, and Junyang Lin. Understanding and improving layer normalization. *Advances in neural information processing systems*, 32, 2019.

Brian D Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University, 2010.

A Related works

Numerous studies underscore the sensitivity of current deep on-policy methods to hyperparameters and implementation details (Huang et al., 2022a; Andrychowicz et al., 2020a), urging the community to simplify and close the gap between theory and practical implementation. Several works have explored training critics using off-policy data (Degris et al., 2012; Haarnoja et al., 2018; Fujimoto et al., 2018), with Bhatt et al. (2024) being one of the first to simplify the critic learning process by eliminating the need for target networks, which were initially popularized by Mnih (2013). In Gallici et al. (2024) the authors study how introducing LayerNorm Ba et al. (2016) stabilizes TD-learning.

Entropy regularization plays a pivotal role in numerous deep RL algorithms (Haarnoja et al., 2018; Bhatt et al., 2024). In fact, the entropy of the policy acts as a regularizer shaping the objective landscape (Ahmed et al., 2019). The prevalent strategy regularizes the policy evaluation phase by supplementing the standard RL task objective with an entropy term. This method guides policies towards regions of higher expected trajectory entropy, a scheme often referred to as maximum entropy RL (Ziebart, 2010; Haarnoja et al., 2018), which is recognized for enhancing the exploration capabilities and robustness of policies by fostering stochasticity. Recent studies on policy gradient methods have highlighted the effectiveness of maximum entropy RL in speeding up convergence (Mei et al., 2020; Ahmed et al., 2019; Cen et al., 2024). In Choe & Kim (2024) the authors integrate the maximum entropy principle to PPO in the context of discrete control.

Temporal-Difference (TD) learning, as described by Sutton & Barto (2018), is a fundamental algorithm for learning value functions. While TD learning with tabular representations and on-policy linear function approximation has established convergence properties (Dayan, 1992; Jaakkola et al., 1993; Tsitsiklis & Van Roy, 1996), standard TD learning is known to potentially diverge with off-policy samples and linear function approximation (Baird, 1995). This divergence concern has motivated the development of convergent off-policy TD methods (Kolter, 2011; Diddigi et al., 2019) and, in part, the on-policy data collection strategy prevalent in algorithms like TRPO and PPO (Schulman et al., 2015a, 2017b), which typically learn a critic Q^π using only on-policy data and often avoid replay buffers. Despite these challenges in basic off-policy TD learning, *off-policy* algorithms (e.g., DDPG, TD3, SAC) (Lillicrap, 2015; Haarnoja et al., 2018; Fujimoto et al., 2018) and offline RL methods (Wu et al., 2019; Kumar et al., 2019; Fujimoto & Gu, 2021) have successfully leveraged off-policy data for critic training. A persistent challenge in off-policy TD learning with function approximation is non-convergence, frequently attributed to state aliasing and the potential for increased variance with larger function approximators (Sutton et al., 2016).

B Supplementary ablation studies

To further analyze the behavior of PPO+, we conducted other ablation studies. First, we evaluated the algorithm using a single critic network to isolate the impact of our dual-critic architecture. This experiment aimed to verify that the performance gains observed compared to PPO are not solely attributable to the presence of two critics. Next, we examined the performance of PPO+ when trained without the Generalized Advantage Estimation (GAE). Our findings indicate that while removing GAE estimation leads to a performance degradation, PPO+ retains its ability to learn effectively. This contrasts with standard PPO implementations, suggesting that our modifications and simplifications have resulted in a more robust critic. However, the observed sensitivity to the absence of GAE highlights that further enhancements to the critic's estimation quality remain a valuable area for future research. Finally, using the minimum of two critics seems to be detrimental for performance, we believe this is because it inhibits exploration in most tasks, thus proving that this design choice is not universal.

C How much off-policy can we go

A pertinent question is to investigate the sensitivity of PPO+ to the proportion of off-policy data utilized for critic training. To this end, the provided plot illustrates the algorithm's performance across a range of critic replay buffer sizes, specifically from $5,000$ to $200,000$. As a reminder, policy rollouts were performed for $5,000$ steps before each update epoch. The results indicate that PPO+ generally maintains its performance across a wide spectrum of buffer sizes for most tasks. Generally, larger buffer sizes tend to yield improved performance. This trend is particularly

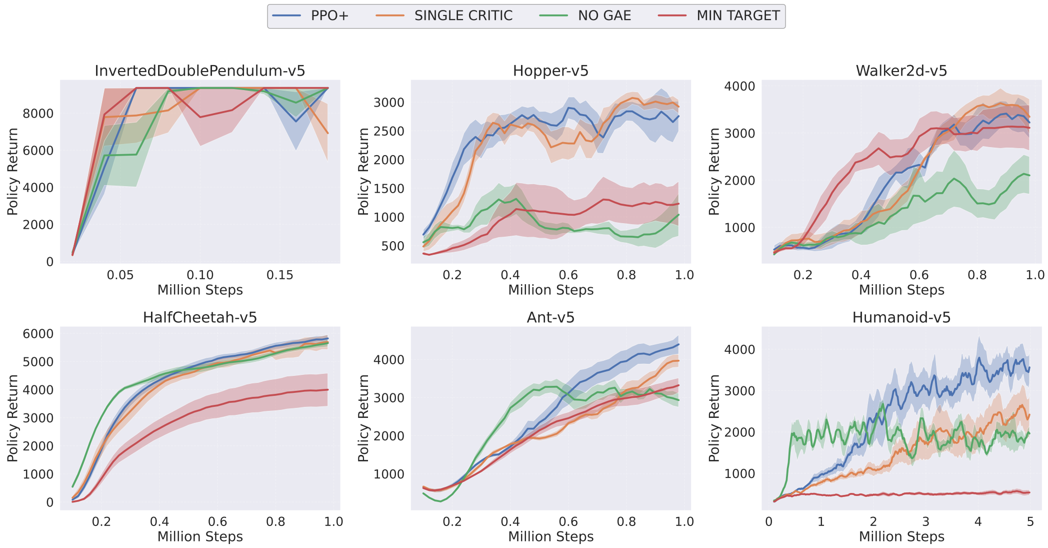


Figure 7: Evolution of the policy return on the MuJoCo-v5 tasks for various design choices. We use 5 random seeds for every algorithm. (`Single critic`) uses a single critic instead of the usual pair. (`NO GAE`) foregoes using GAE-λ. (`MIN TARGET`) uses the minimum prediction of the two critics for bootsrapping.

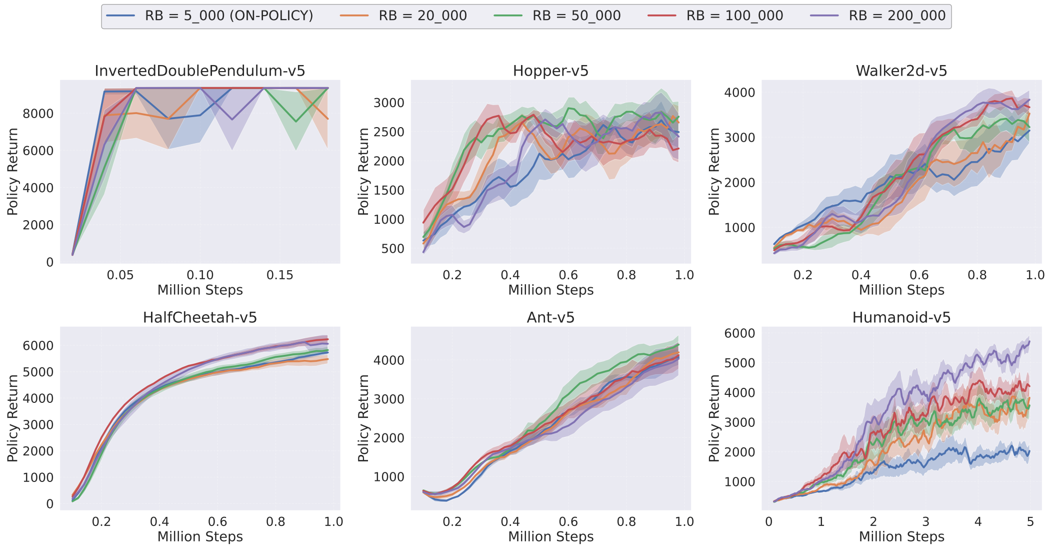


Figure 8: Evolution of the policy return on the MuJoCo-v5 tasks for various replay buffer sizes. We use 5 random seeds for every algorithm.

pronounced in the Humanoid task, where a larger buffer size appears to exert a more significant impact. Employing a replay buffer size of $50,000$ appears to strike a reasonable performance to compute balance.

D Sensitivity to the GAE-λ Parameter

We investigated the sensitivity of PPO+ to the GAE-λ parameter, comparing it against standard PPO's behavior (Figure 9). Standard PPO (bottom row) exhibits strong sensitivity to λ; its performance significantly degrades as λ decreases from the typical 0.95, and learning collapses entirely for $\lambda = 0.0$.

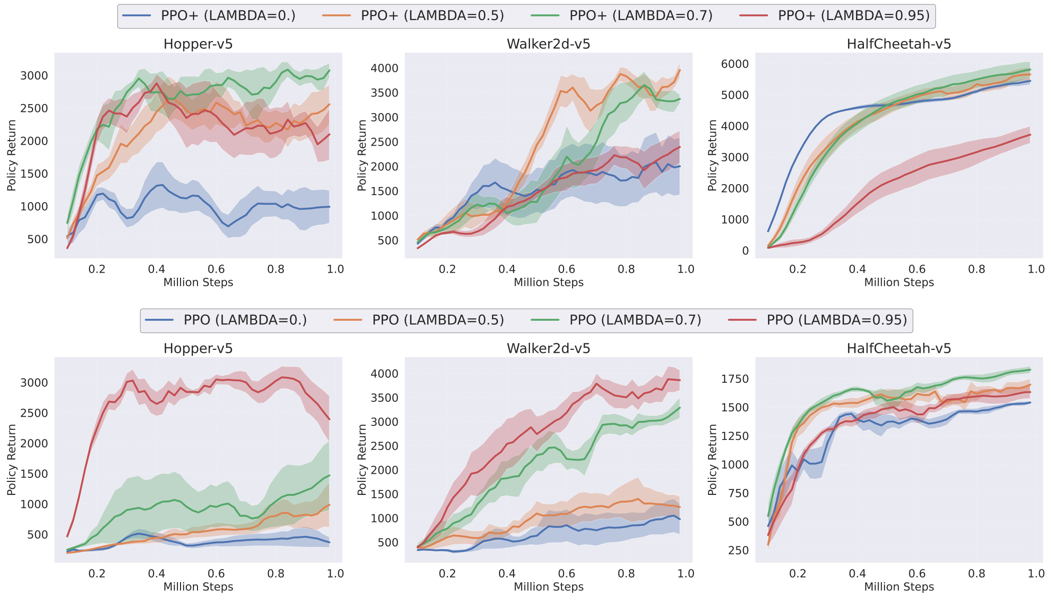


Figure 9: Evolution of the policy return on the MuJoCo-v5 tasks for different values of the GAE-λ. We use 5 random seeds for every configuration.

This suggests PPO heavily relies on near-Monte-Carlo estimates, potentially due to inaccuracies in its critic when more bootstrapping is used.

In contrast, PPO+ (top row) demonstrates greater robustness across different λ values. While higher λ values like 0.95 or 0.7 can yield the best asymptotic performance in some environments (e.g., Hopper-v5, Walker2d-v5), PPO+ with $\lambda = 0.5$ (our default) still achieves effective learning. Crucially, even with $\lambda = 0.0$ (relying entirely on the critic's one-step TD estimate), PPO+ is capable of learning, albeit to a lesser extent, unlike standard PPO which fails completely. This improved resilience to lower λ values points to a more reliable and accurate critic in PPO+, which is less dependent on long-term Monte-Carlo returns and can effectively leverage bootstrapped estimates. The better critic quality in PPO+ likely stems from its principled enhancements, such as off-policy training and bounded actions, which are not present in standard PPO.

E LQR Experiment Configurations

In Section 4.2, we described experiments using Linear Quadratic Regulator (LQR) environments to evaluate the impact of different critic training strategies on the quality of on-policy policy gradient estimation. Figure 3 in the main text presents these results. The core setup involved equipping an agent with multiple critics, with a default configuration of two critics trained using off-policy data from a replay buffer. We then systematically varied one element at a time from this default to analyze the following configurations:

1. `Off-Policy`: This is the default configuration, where two critic networks are trained using off-policy data stored in a replay buffer.
2. `On-Policy`: In this configuration, critics are trained exclusively using data generated by the current policy, with data from previous policies discarded.
3. `Minimum Target`: The critics are trained using the minimum Q-value prediction from the two (twin) critics for bootstrapping the target values. This follows a common heuristic in some off-policy actor-critic methods.
4. `No LayerNorm`: Critics are trained without Layer Normalization in their network architecture.

The experimental protocol for each configuration was as follows:

- For each update step, $5,000$ environment interactions were collected by rolling out the current policy. This data approximates the on-policy state-action visitation distribution d^π and is used for actor updates and, in the `On-Policy` critic setting, for critic updates.

- For critic training (especially in `Off-Policy`, `Minimum Target`, and `No LayerNorm` settings utilizing a replay buffer), $5,000$ Temporal Difference (TD) updates were performed using batches sampled from the replay buffer (which includes data from current and past policies).

- Performance metrics, including the cosine similarity between estimated and true policy gradients, were evaluated over intervals of $20,000$ environment interactions.

This setup was designed to ensure that the quality of the critics under different configurations was measured using the same underlying data distribution at each evaluation point, allowing for a precise isolation of the impact of each specific design choice.

F Pseudocode

Algorithm 1 One Step of PPO+

```
Require: Current actor parameters $\phi$, critic parameters $\theta_1$, $\theta_2$, critic replay buffer $\mathcal{B}$

$\mathcal{D} \leftarrow \emptyset$                                          ▷ Reset the actor replay buffer
for $N_e$ environment steps do
    $a_t \sim \pi_\phi(a_t|s_t)$                                       ▷ Sample action from the policy
    $s_{t+1} \sim p(s_{t+1}|s_t, a_t)$                               ▷ Sample transition from the environment
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{s_t, a_t\}$                             ▷ Update the actor replay buffer
    $\mathcal{B} \leftarrow \mathcal{B} \cup \{s_t, a_t, r_t, s_{t+1}\}$                    ▷ Update the critic replay buffer
    $s_t = s_{t+1}$
end for
for $N_e$ epochs do
    for $N_c$ critic steps do
        $B \leftarrow \{s, a, r, s'\} \sim \mathcal{U}(\mathcal{B})$                   ▷ Sample a batch of off-policy transitions
        $y_i(s, a) = r + \gamma(Q_{\theta_i}(s', a') - \log \pi_\phi(a'|s')),\ a' \sim \pi_\phi(.|s')$    ▷ Compute critic targets
        $\nabla_{\theta_i} \frac{1}{|B|} \sum_{(s,a,r,s') \in B} \left(Q_{\theta_i}(s, a) - y_i(s, a)\right)^2$ for $i = 1, 2$    ▷ Update the critic networks
    end for
    for $N_a$ actor steps do
        $B \leftarrow \{s, a, r, s'\} \sim \mathcal{U}(\mathcal{D})$                   ▷ Sample a batch of on-policy transitions
        $\hat{V}_i(s) = \mathbb{E}_{a \sim \pi}\left[\hat{Q}^\pi_{\theta_i}(s, a)\right], \forall s \in \mathcal{D}$, for $i = 1, 2$    ▷ Compute value function estimates
        $\hat{A}^\pi(s, a) = \frac{1}{2} \sum_{i \in 1,2} \hat{Q}^\pi_{\theta_i}(s, a) - \hat{V}_i(s), \forall s, a \in \mathcal{D}$        ▷ Compute advantage function
    estimates (Can also use GAE-$\lambda$)
        $\phi = \arg\max_{\phi'} \sum_{(s_t, a_t, \gamma_t) \in \mathcal{D}} \gamma_t \min\left(\frac{\pi_{\phi'}(a_t|s_t)}{\pi_\phi(a_t|s_t)} \hat{A}^\pi(s_t, a_t), \text{clip}\left(\epsilon, \hat{A}^\pi(s_t, a_t)\right)\right)$
    end for
end for
return $\phi$, $\theta_1$, $\theta_2$                                       ▷ Optimized parameters
```

G Hyperparameters

Table 2: Hyperparameters for on-policy baselines. For PPO, TRPO and SAC we use the standard implementations provided in CleanRL (for TRPO we used a code that is awaiting approval to be integrated into CleanRL) Huang et al. (2022b)

Parameter	PPO+	PPO (ours)
optimizer	Adam	Adam
learning rate	$3 \cdot 10^{-4}$	$3 \cdot 10^{-4}$
discount (γ)	0.99	0.99
replay buffer size	$5 \cdot 10^{4}$	$\varnothing$
number of critics	2	1
LayerNorm	True	True
number of hidden layers	2	2
number of hidden units per layer	256	256
number of samples per minibatch	250	250
activation function	TanH	TanH
actor update interval N_e	5000 steps	5000 steps
GAE Lamda	0.5	0.95