
Reusing Past Samples in Proximal Policy Optimization: When and How Does It Help?

Alessandro Montenegro

Politecnico di Milano, Milan, Italy
alessandro.montenegro@polimi.it

Riccardo Venturelli

Politecnico di Milano, Milan, Italy
riccardo.venturelli@polimi.it

Marco Mussi

Politecnico di Milano, Milan, Italy
marco.mussi@polimi.it

Matteo Papini

Università degli Studi di Milano, Milan, Italy
matteo.papini@unimi.it

Alberto Maria Metelli

Politecnico di Milano, Milan, Italy
albertomaria.metelli@polimi.it

Abstract

Among on-policy *deep reinforcement learning* methods, Proximal Policy Optimization (PPO) has become the de facto standard, due to its consistently strong empirical performance across diverse application domains. However, on-policy methods are inherently sample inefficient: fresh data collected under the current policy is used for just a *few updates* before being discarded. Off-policy methods avoid this inefficiency via experience replay, achieving notable sample efficiency gains, but at the cost of training instabilities or extensive tuning. This motivated the rise of *hybrid strategies* that augment PPO with off-policy data reuse. Existing sample-reuse variants of PPO demonstrated improved sample efficiency over *vanilla* PPO, yet a systematic study of when reuse helps, in which scenarios, and to what extent remains missing. In this work, we study the effectiveness of sample reuse in PPO by instantiating two variants within a *multiple importance weighting* framework. Both retain the core PPO mechanics, reusing only samples from a *window* of recent iterations, thereby isolating the effect of data reuse from other factors. The variants, termed $\omega\text{PPO-U}$ and $\omega\text{PPO-BH}$, employ *vanilla importance weights* or *balance-heuristic-corrected ones*, respectively. For both, we derive policy improvement lower bounds providing theoretical grounding for their respective losses. We use them to empirically study when and how data reuse improves sample efficiency or final performance of PPO across continuous control tasks.

1 Introduction

Within *reinforcement learning* [RL, 55] approaches, *policy optimization* [PO, 9] methods have proved to be highly effective for solving complex, large-scale sequential decision-making problems, both in simulated [11] and real-world [7, 32, 59] environments. Their success is attributable to several desirable properties: applicability to continuous settings [43], robustness to noise [18], resilience to partial observability [5], and possibility of incorporating the expert domain knowledge during the policy design phase [17]. Taken together, these properties have established PO as the dominant paradigm for continuous control, robot locomotion, and complex simulated tasks.

A key distinction among PO methods is between *off-policy* and *on-policy* approaches. Off-policy methods require an RL agent to update its *target policy* using interaction data collected under a

different *behavioral policy*, including data gathered across multiple past policies. Notable representatives of this family include SAC [21], DDPG [29], and TD3 [16]. These methods are known for their sample efficiency, as they can store and reuse past experience via a replay buffer, and often achieve strong final performance. However, they tend to be computationally intensive, frequently requiring careful implementation choices and extensive tuning. On-policy methods, by contrast, restrict updates to fresh interaction data collected under the current target policy, discarding those samples after one or a few gradient steps. This family includes REINFORCE [64], GPOMDP [6], TRPO [50], PPO [52], POIS [34], and GRPO [53]. By construction, on-policy methods offer stable training dynamics and admit meaningful theoretical guarantees, ranging from convergence results [68, 15] to policy improvement bounds [23, 44, 1]. Their main drawback, however, is poor sample efficiency: each collected trajectory contributes to only a single round of updates before being discarded.¹

Given the sample efficiency of off-policy methods and the stability and theoretical foundations of on-policy ones, an active research direction pursues methods that exhibit the best of both worlds. Most efforts have focused on augmenting on-policy methods with off-policy data [20, 40, 14], enabling the reuse of samples collected under past policies. This is commonly achieved via *importance weighting* [37], a technique that corrects for the distributional mismatch between the behavioral and target policies by reweighting samples by the ratio of their likelihoods under the two policies. Since these weights can exhibit high variance when the two policies are dissimilar, some corrections have been proposed to make their use more reliable, including the *balance heuristic* [BH, 58, 35] estimator. In the context of policy gradient methods [43], substantial work has analyzed whether reusing either past gradients [40, 65, 67, 39] or past trajectories [42, 30, 36] can yield faster convergence rates, with affirmative results in both cases, that, however, remain tied to theoretical settings.

These findings have motivated the investigation of similar gains in practical deep on-policy algorithms [63, 19, 20, 14]. Much of this effort has concentrated on PPO, which has become the de facto standard deep on-policy method thanks to its strong empirical performance across application domains ranging from robot locomotion [27, 7, 32, 59] to LLM fine-tuning [62, 66]. This success has, in turn, motivated deep investigation of PPO from multiple standpoints, including theoretical convergence guarantees [22, 10], implementation best practices [12, 4, 24], and algorithmic variants [60, 61, 54, 8]. Among the latter, several works have specifically targeted the incorporation of off-policy data into PPO. Queeney et al. [46] derive an off-policy policy improvement lower bound from the *performance difference lemma* [23], proposing GePPO, a PPO variant that reuses data from a window of recent iterations. GePPO requires re-evaluating advantages on past samples via V-Trace [13], incurring additional computation, and employs a modified clipping mechanism together with an adaptive learning rate based on the total variation between the current and behavioral policies. Zhao et al. [69] propose SARD-PPO, a sample reuse PPO variant that adaptively discards samples and re-evaluates past data under the current critic, employing a modified dual clipping mechanism. Liang et al. [28] incorporate *prioritized experience replay* [49] into PPO, proposing PTR-PPO and studying different priority assignment strategies. Meng et al. [33] introduce Off-Policy PPO, which also requires re-evaluating past trajectories under the current critic to obtain fresh advantage estimates, alongside a heuristic clipping mechanism. Recently, Liu et al. [31] proposed HP3O, a sample reuse PPO variant maintaining a FIFO trajectory replay buffer, re-evaluates past samples under the current critic, and retains the best-performing trajectory in each optimization batch. Finally, Lin et al. [30] provide an initial experimental exploration of RPPO, a PPO variant that reuses samples from recent policies without re-evaluating advantages under the new critic. All these methods confirm both the ongoing research interest in PPO and the practical benefits of past-data reuse. However, most approaches require re-evaluation of stored samples under the current critic, rely on heuristics or modified clipping mechanisms, and are presented as standalone methods with tuned configurations rather than as controlled studies of the reuse mechanism itself. To the best of our knowledge, a systematic investigation of when, how, and why data reuse in PPO is beneficial remains unaddressed.

Original Contribution. In this paper, we aim to conduct a controlled study of the effects of sample reuse in PPO. To preserve the core functioning of PPO and isolate the effect of data reuse from other factors, we consider the incorporation of past collected data within the standard *multiple importance weighting* [MIW, 38, 35] framework. This amounts to the simplest possible form of sample reuse

¹The distinction between on-policy and off-policy methods is a spectrum rather than strictly binary. Methods such as PPO and POIS perform multiple gradient steps on a fixed batch, introducing a mild distributional shift between the behavioral and target policies within each update cycle. Nevertheless, these methods are conventionally classified as on-policy, since the collected data is discarded at the end of each iteration and never reused across policy updates.

for PPO: augmenting its on-policy dataset by retaining data collected over a window of the ω most recent iterations, optionally correcting the vanilla *importance weights* [IW, 38]. Specifically, our main contributions can be summarized as follows:

- **Uniform Sample Reuse.** We study the straightforward MIW instantiation of PPO, which we term $\omega_{\text{PPO-U}}$. Similarly to [30], it uniformly weights all samples in the composite dataset using vanilla IWs between the target and behavioral policies, and employs advantage estimates computed under the critic available at collection time. To ground this approach, we derive a policy improvement lower bound that accounts for stale advantages. Beyond validating the $\omega_{\text{PPO-U}}$ loss, the bound reveals a penalty term independent of the optimization variable, which therefore cannot be controlled by actor updates, suggesting that the policies within the considered window should be close (Section 3.1).
- **BH-Corrected Sample Reuse.** Motivated by the usual inefficiency of uniform weighting in MIW, we additionally study a PPO instantiation employing BH-corrected IWs, which we term $\omega_{\text{PPO-BH}}$. This consists of weighting each sample via the ratio between the target policy and the uniform mixture of behavioral policies in the window. $\omega_{\text{PPO-BH}}$ inherits all properties of $\omega_{\text{PPO-U}}$ and is similarly grounded by a policy improvement lower bound, whose uncontrollable penalty terms are non-decreasing in ω , reinforcing the recommendation for keeping the policies within the window close (Section 3.2).
- **Systematic Empirical Study.** We compare $\omega_{\text{PPO-U}}$ and $\omega_{\text{PPO-BH}}$ with standard PPO on MuJoCo [57] tasks under controlled settings that vary how updates and data collection are matched between methods. We study whether reuse helps convergence speed, final performance, or both, whether the benefit grows indefinitely with larger windows, and when excessive reuse becomes harmful. Finally, we study the differences between the two considered weightings (Section 4). Proofs, experimental details, and additional experiments are in Appendices B, C, and D, respectively.

2 Preliminaries

Notation. For $n, m \in \mathbb{N}$ with $n \geq m$, we denote $\llbracket m, n \rrbracket := \{m, m+1, \dots, n\}$ and $\llbracket n \rrbracket := \llbracket 1, n \rrbracket$. We denote by $\Delta(\mathcal{X})$ the set of probability measures over the measurable set $\mathcal{X}$. For $P \in \Delta(\mathcal{X})$, we denote by p its density function w.r.t. a reference measure that we assume to exist whenever needed. The *total variation* (TV) distance is defined as $D_{\text{TV}}(P, Q) := \frac{1}{2} \int_{\mathcal{X}} |p(x) - q(x)| dx$.

Markov Decision Processes. A Markov Decision Process [MDP, 45] is represented by $\mathcal{M} := (\mathcal{S}, \mathcal{A}, p, r, d_0, \gamma, T)$, where $\mathcal{S} \subseteq \mathbb{R}^{d_{\mathcal{S}}}$ and $\mathcal{A} \subseteq \mathbb{R}^{d_{\mathcal{A}}}$ are the measurable state and action spaces; $p : \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{S})$ is the transition model, where $p(s'|s, a)$ is the probability density of landing in state $s' \in \mathcal{S}$ by playing action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$; $r : \mathcal{S} \times \mathcal{A} \rightarrow [-R_{\max}, R_{\max}]$ is the reward function, where $r(s, a)$ is the reward the agent gets when playing action a in state s ; $d_0 \in \Delta(\mathcal{S})$ is the initial-state distribution; $\gamma \in [0, 1]$ is the discount factor. A trajectory $\tau = (s_{\tau,1}, a_{\tau,1}, \dots, s_{\tau,T}, a_{\tau,T})$ of length $T \in \mathbb{N} \cup \{+\infty\}$ is a sequence of T state-action pairs. We denote by $\mathcal{T}$ the set of all trajectories. The *discounted return* of a trajectory $\tau \in \mathcal{T}$ is given by $R(\tau) := \sum_{t=1}^T \gamma^{t-1} r(s_{\tau,t}, a_{\tau,t})$.²

Policy Optimization. We consider *parametric stochastic policies* $\pi_{\theta} : \mathcal{S} \rightarrow \Delta(\mathcal{A})$ with parameters $\theta \in \Theta \subseteq \mathbb{R}^{d_{\Theta}}$. The policy π_{θ} is used to sample actions at each interaction instant t as $a_t \sim \pi_{\theta}(\cdot | s_t)$, where s_t is the state observed at time t . The performance of a policy π_{θ} is assessed via the *expected return* $J : \Theta \rightarrow \mathbb{R}$, defined as:

$$J(\theta) := \mathbb{E}_{\substack{s \sim d_{\theta} \\ a \sim \pi_{\theta}(\cdot | s)}} [r(s, a)],$$

where, with an abuse of notation, the expectation is taken over $d_{\theta}(s) := \sum_{t=1}^T \gamma^{t-1} \Pr(s_t = s | s_1 \sim d_0, \pi_{\theta})$, the γ -discounted state occupancy under π_{θ} , which is not a proper probability distribution. Moreover, the γ -discounted state-action occupancy under π_{θ} is defined as $\mu_{\theta}(s, a) := d_{\theta}(s) \pi_{\theta}(a | s)$, for every $s \in \mathcal{S}$ and $a \in \mathcal{A}$. Two additional performance indices are the *value function* $V^{\pi_{\theta}} : \mathcal{S} \rightarrow \mathbb{R}$ and the *action-value function* $Q^{\pi_{\theta}} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, defined respectively as: $V^{\pi_{\theta}}(s) := \mathbb{E}[\sum_{t=1}^T \gamma^{t-1} r(s_t, a_t) | s_1 = s, \pi_{\theta}]$ and $Q^{\pi_{\theta}}(s, a) := \mathbb{E}[\sum_{t=1}^T \gamma^{t-1} r(s_t, a_t) | s_1 = s, a_1 = a, \pi_{\theta}]$. The goal of PO is to find $\theta^* \in \arg \max_{\theta \in \Theta} J(\theta)$.

Trust Region Methods and Proximal Policy Optimization. To learn θ^* in an on-policy fashion, one can either follow the *policy gradient theorem* [56] or one can build surrogate losses allowing for

²We allow $\gamma = 1$ only when $T < +\infty$.

using multiple times the same collected data while constraining the target policy to remain inside a *trust region*, meaning that the target policy is kept near the behavioral one, ensuring the sampled data remain trustworthy. This approach is grounded in the following *policy improvement* lower bound [1]:

$$J(\theta') - J(\theta) \geq \mathbb{E}_{\substack{s \sim d_\theta \\ a \sim \pi_\theta(\cdot | s)}} \left[\frac{\pi_{\theta'}(a | s)}{\pi_\theta(a | s)} A^{\pi_\theta}(s, a) \right] - \frac{2\gamma C(\theta', \theta)}{(1 - \gamma)^2} \mathbb{E}_{s \sim d_\theta} [D_{\text{TV}}(\pi_{\theta'}(\cdot | s), \pi_\theta(\cdot | s))],$$

where $A^{\pi_\theta}(s, a) := Q^{\pi_\theta}(s, a) - V^{\pi_\theta}(s)$ is the *advantage function*, $\frac{\pi_{\theta'}(a | s)}{\pi_\theta(a | s)}$ is the *importance weight* [IW, 37], and $C(\theta', \theta) := \sup_{s \in S} |\mathbb{E}_{a \sim \pi_{\theta'}(\cdot | s)} [A^{\pi_\theta}(s, a)]|$. The bound reveals that the closer $\pi_{\theta'}$ and π_θ are in TV, the tighter the bound. TRPO [50] operationalizes this by maximizing the importance-weighted surrogate subject to a KL constraint, at the cost of requiring second-order information in practice. PPO [52] replaces this constraint with a clipping heuristic, optimizing:

$$L^{\text{PPO}}(\theta') := \mathbb{E}_{\substack{s \sim d_\theta \\ a \sim \pi_\theta(\cdot | s)}} \left[\min \left\{ \frac{\pi_{\theta'}(a | s)}{\pi_\theta(a | s)} A^{\pi_\theta}(s, a), \text{clip} \left(\frac{\pi_{\theta'}(a | s)}{\pi_\theta(a | s)}, 1 - \epsilon, 1 + \epsilon \right) A^{\pi_\theta}(s, a) \right\} \right],$$

where $\text{clip}(a, b, c) = \min \{ \max \{ a, b \}, c \}$. In practice, this loss is approximated using a parameterized critic $V_w(s) \approx V^{\pi_\theta}(s)$ with parameters $w \in \mathcal{W} \subseteq \mathbb{R}^{d_w}$. The policy π_θ is used to collect a dataset of transitions $\mathcal{D}_\theta = \{(s_t, a_t, \pi_\theta(a_t | s_t), A_t, R_t)_{t=1}^H\}$, where H is the number of samples, $A_t \approx A^{\pi_\theta}(s_t, a_t)$ is estimated via *generalized advantage estimation* [GAE, 51], and $R_t \approx V^{\pi_\theta}(s_t)$. The surrogate is then approximated as $\hat{L}^{\text{PPO}}(\theta', \mathcal{D})$ and the critic is fitted via mean squared error minimization over R_t . Further details on the practical functioning of PPO are deferred to Appendix A.

Multiple Importance Weighting. Let $x \in \mathcal{X}$ be a random variable and let $f : \mathcal{X} \rightarrow \mathbb{R}$ be a function. Consider the goal of estimating the expected value of f under a target distribution P , but only having samples coming from ω behavioral distributions $\{Q_i\}_{i=1}^\omega$. Specifically, for each $i \in [\omega]$ consider to have access to a dataset $\mathcal{D}_i = \{x_{i,j}\}_{j=1}^N$ of N independent samples drawn from Q_i . Assuming for every $i \in [\omega]$ that $P \ll Q_i$ (P is absolutely continuous w.r.t. Q_i), the *multiple importance weighting* (MIW) estimator of the expected value of f can be written as [38]:

$$\hat{\mu}_{P/\{Q_i\}_{i=1}^\omega} = \sum_{i=1}^\omega \frac{1}{N} \sum_{j=1}^N \alpha_i(x_{i,j}) \frac{p(x_{i,j})}{q_i(x_{i,j})} f(x_{i,j}),$$

where the coefficients α_i are a partition of the unit, i.e., for every $i \in [\omega]$ and $x \in \mathcal{X}$ we have that $\alpha_i(x) \in \mathbb{R}_{\geq 0}$ and $\sum_{i=1}^\omega \alpha_i(x) = 1$. Notice that $\hat{\mu}$ is unbiased. Among the choices for the α_i coefficients the most simple but inefficient choice is the uniform weighting, meaning for each $i \in [\omega]$ and $x \in \mathcal{X}$ we select $\alpha_i(x) = 1/\omega$, thus giving equal weight to all the samples [35]. Among the various choices for such coefficients, the most studied one is the *balance heuristic* [BH, 58], provably nearly optimal in terms of variance within the class of MIW estimators. The BH prescribes to select the weights as: $\alpha_i^{\text{BH}}(x) = \frac{q_i(x)}{\sum_{l=1}^\omega q_l(x)}$, leading to the following unbiased estimator:

$$\hat{\mu}_{P/\{Q_i\}_{i=1}^\omega}^{\text{BH}} = \frac{1}{\omega N} \sum_{i=1}^\omega \sum_{j=1}^N \frac{p(x_{i,j})}{\frac{1}{\omega} \sum_{l=1}^\omega q_l(x_{i,j})} f(x_{i,j}),$$

that, whenever $P \in \{Q_i\}_{i=1}^\omega$, makes the BH-corrected ratio bounded by ω .

3 Sample Reuse in PPO

In this section, we first revisit the functioning of PPO and the data reuse mechanism already implicit in its design. We then instantiate two PPO variants within a MIW framework [38, 35] that incorporate past collected data from a window of at most ω recent iterations, without re-evaluating the advantage signals under the current critic. The first, $\omega\text{PPO-U}$, employs uniform weighting across the window; the second, $\omega\text{PPO-BH}$, applies a BH [58] correction to the IWs. For both, we derive a policy improvement lower bound that accounts for stale advantages: beyond providing theoretical grounding for the losses optimized by the reuse variants, the bounds reveal penalty terms independent of the optimization variable, suggesting that the policies in the reuse window should remain close to each other, while the target policy should remain close to the behavioral ones.

Native PPO's Sample Reuse. PPO already incorporates a form of sample reuse. However, this is limited in two ways: (i) it only considers data collected under the *same* policy, and (ii) it only

considers data from the *most recent* behavior policy. Specifically, at iteration $k \in \mathbb{N}$, policy π_{θ_k} and critic V_{w_k} are used to build dataset $\mathcal{D}_k$ by interacting with the environment for H steps, so that $|\mathcal{D}_k| = D$. For $\tilde{K}$ epochs, $\mathcal{D}_k$ is shuffled and divided into N_B mini-batches of size B , each used to perform one update of both actor and critic. After performing $\tilde{K} N_B$ updates, $\mathcal{D}_k$ is discarded.

Cross-Iteration Sample Reuse Framework. As highlighted above, one of the causes of PPO’s sample inefficiency lies in the fact that the dataset $\mathcal{D}_k$ is discarded after a few updates. Instead, past samples could be retained and reused in future iterations, increasing sample efficiency as done in related work [46, 33, 69, 31]. Specifically, suppose we have access to a window of $\omega_k := \min\{k, \omega\}$ recent iterations, where ω is the maximum window size. When updating the actor and critic, we consider the composite dataset $\mathcal{D}(\omega_k) := \bigcup_{i=0}^{\omega_k-1} \mathcal{D}_{k-i}$, which we shuffle and from which we sample mini-batches, equally representing all the iterations in the window. We assume each policy performs H interaction steps with the environment, so that all the datasets will have the same cardinality, i.e., $|\mathcal{D}_{k-i}| = D$ for every $i \in [0, \omega_k - 1]$ and $|\mathcal{D}(\omega_k)| = \sum_{i=0}^{\omega_k-1} |\mathcal{D}_{k-i}| = \omega_k D$. Moreover, for each $i \in [0, \omega_k - 1]$ let $\mathcal{D}_{k-i} = \{(s_{i,j}, \mathbf{a}_{i,j}, \pi_{\theta_{k-i}}(\mathbf{a}_{i,j} | s_{i,j}), A_{i,j}, R_{i,j})\}_{j=1}^D$, where $A_{i,j}$ and $R_{i,j}$ are obtained with the critic available at iteration $k - i$. The key difference with respect to the native PPO reuse is that the data buffer now contains samples from *multiple* behavioral policies rather than from a single, most recent one. Next, we discuss two MIW instantiations of PPO for exploiting $\mathcal{D}(\omega_k)$.

3.1 Cross-Iteration Uniform Sample Reuse in PPO

Within the described *reuse framework*, the simplest form of reuse consists of *uniformly weighting* all the samples in $\mathcal{D}(\omega_k)$. While uniform weighting is known to be inefficient in MIW frameworks [38, 35], it may nevertheless be meaningful in the PPO setting, where the clipping mechanism already limits the contribution of samples far from the target policy. A first empirical hint of this approach is discussed in [30]. We refer to the PPO variant employing this weighting scheme as $\omega\text{PPO-U}$, whose empirical loss is:

$$\hat{L}^U(\theta, \mathcal{D}(\omega_k)) = \frac{1}{\omega_k D} \sum_{i=0}^{\omega_k-1} \sum_{j=1}^D \min \left\{ \frac{\pi_{\theta}(\mathbf{a}_{i,j} | s_{i,j})}{\pi_{\theta_{k-i}}(\mathbf{a}_{i,j} | s_{i,j})} A_{i,j}, \text{clip} \left(\frac{\pi_{\theta}(\mathbf{a}_{i,j} | s_{i,j})}{\pi_{\theta_{k-i}}(\mathbf{a}_{i,j} | s_{i,j})}, 1 - \epsilon, 1 + \epsilon \right) A_{i,j} \right\},$$

where θ is the target actor parameter vector. Some comments are in order. First, in line with our goal of preserving the core PPO functioning, we deliberately use *stale advantages*, i.e., advantage estimates computed at data collection time under the critic available at that iteration, rather than re-evaluating past samples under the most recent critic V_{w_k} [e.g., 46, 33, 69], a practice typically carried out via V-Trace [13] and incurring additional computation. Second, without re-evaluating past data under the current critic, updating the critic on stale returns from distant iterations would mean to update the critic on increasingly stale values, especially for large ω and $\tilde{K}$ values. We therefore restrict critic updates to the most recent dataset $\mathcal{D}_k$ only, using past samples exclusively to produce additional actor updates. Third, this loss can be easily employed in mini-batch stochastic gradient ascent as soon as each mini-batch equally represents the full dataset $\mathcal{D}(\omega_k)$, meaning that each window of reuse should have the same amount of data inside each mini-batch.

Now, we show a policy improvement lower bound providing the theoretical grounding for $\hat{L}^U$.

Theorem 3.1. *Let $\omega_k \in \mathbb{N}_{\geq 1}$. Having access to $\{\pi_{\theta_{k-i}}\}_{i=0}^{\omega_k-1}$, for any target π_{θ} , the following holds:*

$$\begin{aligned} J(\theta) - J(\theta_k) &\geq \frac{1}{\omega_k} \sum_{i=0}^{\omega_k-1} \mathbb{E}_{\substack{s \sim d_{\theta_{k-i}} \\ \mathbf{a} \sim \pi_{\theta_{k-i}}(\cdot | s)}} \left[\frac{\pi_{\theta}(\mathbf{a} | s)}{\pi_{\theta_{k-i}}(\mathbf{a} | s)} A^{\pi_{\theta_{k-i}}}(s, \mathbf{a}) \right] \\ &\quad - \frac{2\gamma C(\theta, \theta_k)}{(1 - \gamma)^2 \omega_k} \sum_{i=0}^{\omega_k-1} \mathbb{E}_{s \sim d_{\theta_{k-i}}} [D_{\text{TV}}(\pi_{\theta}(\cdot | s), \pi_{\theta_{k-i}}(\cdot | s))] \\ &\quad - \frac{4R_{\max}(1 + \gamma)}{(1 - \gamma)\omega_k} \sum_{i=0}^{\omega_k-1} \sup_{s \in \mathcal{S}} D_{\text{TV}}(\pi_{\theta_k}(\cdot | s), \pi_{\theta_{k-i}}(\cdot | s)). \end{aligned}$$

Several remarks are in order. First, unlike prior work [46, 33, 69], this bound accounts for *stale advantages*: it features $A^{\pi_{\theta_{k-i}}}$ rather than $A^{\pi_{\theta_k}}$, and therefore does not require access to advantage values under the current policy π_{θ_k} for samples collected under past behavioral policies $\pi_{\theta_{k-i}}$. Second, compared to the standard policy improvement lower bound [44, 1], recovered when $\omega = 1$,

the factor $C(\theta, \theta_k)$ is unchanged, while two additional averaged TV penalty terms appear. The first, involving $D_{\text{TV}}(\pi_\theta(\cdot|s), \pi_{\theta_{k-i}}(\cdot|s))$, already appears in [46, 69, 31] and accounts for sampling $(s, \mathbf{a})$ from $\mu_{\theta_{k-i}}$. The second, involving $D_{\text{TV}}(\pi_{\theta_k}(\cdot|s), \pi_{\theta_{k-i}}(\cdot|s))$, is the price paid for considering stale advantages. Crucially, the latter does not depend on the target policy θ , so it vanishes when taking the gradient of the surrogate loss, leaving the optimization unaffected and thus grounding $\hat{L}^\text{U}$. However, since it cannot be controlled by the actor update, it suggests that the policies belonging to the reuse window ω should remain close to each other. Additionally, the first penalty, which depends on θ , requires the target policy to remain close to all the behavioral policies in the window. In PPO-like methods, this is partially enforced by the clipping mechanism, which limits how far consecutive policies can drift.

3.2 Cross-Iteration Balance-Heuristic-Corrected Sample Reuse in PPO

As the *balance heuristic* [BH, 58] is the most studied MIW weighting due to its desirable theoretical properties [35] and effectiveness [41, 36] over uniform weighting, here we study its instantiation in PPO, which we term $\omega\text{PPO-BH}$. Specifically, we replace the vanilla IWs in the actor loss with per-step BH-corrected ones. To do so, for every $s \in \mathcal{S}$ and $\mathbf{a} \in \mathcal{A}$, let $\pi_{\omega_k}^{\text{BH}}(\mathbf{a} | s) := \frac{1}{\omega_k} \sum_{i=0}^{\omega_k-1} \pi_{\theta_{k-i}}(\mathbf{a} | s)$ be the uniform mixture of policies in the window. The empirical loss of $\omega\text{PPO-BH}$ is:

$$\hat{L}^{\text{BH}}(\theta, \mathcal{D}(\omega_k)) = \frac{1}{\omega_k D} \sum_{i=0}^{\omega_k-1} \sum_{j=1}^D \min \left\{ \frac{\pi_\theta(\mathbf{a}_{i,j} | s_{i,j})}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a}_{i,j} | s_{i,j})} A_{i,j}, \text{clip} \left(\frac{\pi_\theta(\mathbf{a}_{i,j} | s_{i,j})}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a}_{i,j} | s_{i,j})}, 1 - \epsilon, 1 + \epsilon \right) A_{i,j} \right\}.$$

Some comments are in order. First, we highlight that $\hat{L}^{\text{BH}}$ retains most of the features already discussed for $\hat{L}^\text{U}$ including exploiting stale advantages, without re-evaluating past samples under the most recent critic. Also in this case, we consider fitting the critic over $\mathcal{D}_k$ only, thus employing older data exclusively for additional actor updates. Second, the BH correction weights each sample with all behavioral policies in the window jointly, rather than only its own. As a consequence, unlike $\omega\text{PPO-U}$, where the IWs of samples in $\mathcal{D}_k$ equal 1 at the beginning of each iteration, the BH ratio $\pi_{\theta_k}/\pi_{\omega_k}^{\text{BH}}$ generally differs from 1 whenever $\omega_k > 1$, so that even the most recent samples may be clipped from the first update. Third, the BH is known to reduce the variance of the IWs compared to uniform weighting [58, 35]. Moreover, since π_{θ_k} is a component of the mixture at the beginning of each iteration, the BH ratios are bounded by ω_k , whereas the vanilla ratios $\pi_{\theta_k}/\pi_{\theta_{k-i}}$ of older samples are unbounded. Fourth, when updating, the target policy π_θ is pushed towards the boundary of the trust region of $\pi_{\omega_k}^{\text{BH}}$, rather than towards the individual trust regions around each $\pi_{\theta_{k-i}}$ as in $\omega\text{PPO-U}$. This provides a single trust region centered at the mixture of the policies in the window. Finally, it is worth noting that $\omega\text{PPO-BH}$ incurs a higher computational cost than $\omega\text{PPO-U}$. Specifically, evaluating the BH-corrected loss requires evaluating all samples in $\mathcal{D}(\omega_k)$ under all policies $\{\pi_{\theta_{k-i}}\}_{i=0}^{\omega_k-1}$ in the window. However, this cost of $2(\omega_k - 1)D$ evaluations is paid *once per data collection step*.

Now, we show a policy improvement lower bound serving as a theoretical grounding of $\hat{L}^{\text{BH}}$.

Theorem 3.2. *Let $\omega_k \in \mathbb{N}_{\geq 1}$. Having access to $\{\pi_{\theta_{k-i}}\}_{i=0}^{\omega_k-1}$, let $d_{\omega_k}^{\text{BH}}(s) := \frac{1}{\omega_k} \sum_{i=0}^{\omega_k-1} d_{\theta_{k-i}}(s)$ be the uniform mixture of the discounted state occupancies. Then, for any π_θ , the following holds:*

$$\begin{aligned} J(\theta) - J(\theta_k) &\geq \\ &\frac{1}{\omega_k} \sum_{i=0}^{\omega_k-1} \mathbb{E}_{\substack{s \sim d_{\theta_{k-i}} \\ \mathbf{a} \sim \pi_{\theta_{k-i}}(\cdot|s)}} \left[\frac{\pi_\theta(\mathbf{a} | s)}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} | s)} A^{\pi_{\theta_{k-i}}}(s, \mathbf{a}) \right] - \frac{2\gamma C(\theta, \theta_k)}{(1-\gamma)^2 \omega_k} \sum_{i=0}^{\omega_k-1} \mathbb{E}_{s \sim d_{\theta_{k-i}}} [D_{\text{TV}}(\pi_\theta(\cdot|s), \pi_{\omega_k}^{\text{BH}}(\cdot|s))] \\ &- \frac{4R_{\max}(1+\gamma)}{1-\gamma} \sum_{i=0}^{\omega_k-1} \sup_{s \in \mathcal{S}} D_{\text{TV}}(\pi_{\theta_k}(\cdot|s), \pi_{\theta_{k-i}}(\cdot|s)) - \left(\frac{4R_{\max}}{(1-\gamma)} + \frac{2\gamma C(\theta, \theta_k)}{(1-\gamma)^2 \omega_k} \right) \sum_{i=0}^{\omega_k-1} D_{\text{TV}}(d_{\omega_k}^{\text{BH}}, d_{\theta_{k-i}}). \end{aligned}$$

Several remarks are in order. First, as for Theorem 3.1, this bound accounts for stale advantages and recovers the standard policy improvement lower bound [1, 44] when $\omega = 1$. Second, the bound features three TV-based penalty terms. The first involves an average of $D_{\text{TV}}(\pi_\theta(\cdot|s), \pi_{\omega_k}^{\text{BH}}(\cdot|s))$ over the state occupancies in the window, suggesting that the effective trust region should be defined with respect to $\pi_{\omega_k}^{\text{BH}}$ rather than any individual behavioral policy; note that $C(\theta, \theta_k)$ remains unchanged relative to Theorem 3.1 and prior work [1, 46, 69]. The second penalty, involving $\sup_s D_{\text{TV}}(\pi_{\theta_k}(\cdot|s), \pi_{\theta_{k-i}}(\cdot|s))$, arises from window-specific stale advantages and, unlike the analo-

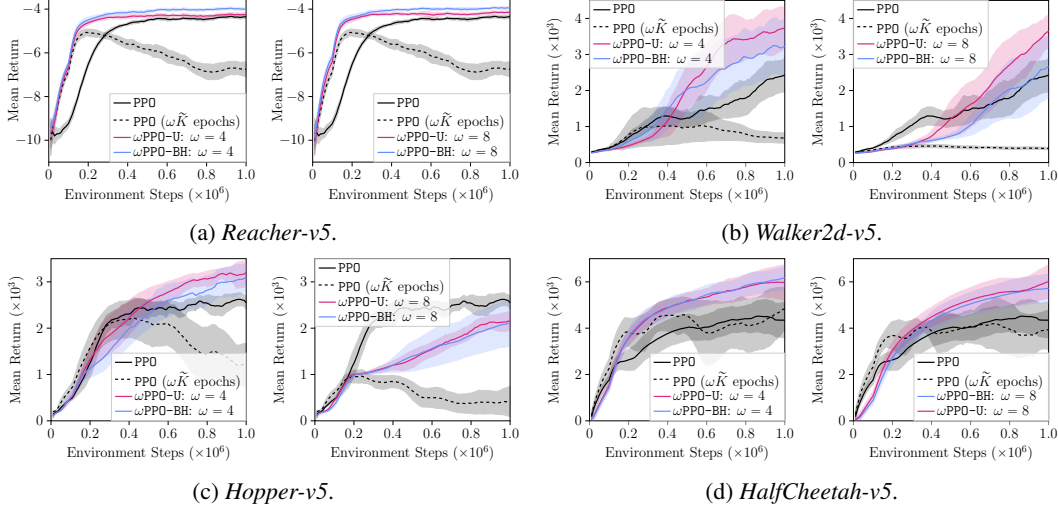


Figure 1: PPO, ω PPO-U, and ω PPO-BH in the *fixed B* setting (10 runs, mean $\pm$ 95% C.I.).

gous term in Theorem 3.1, is non-decreasing in ω . The third, involving $D_{\text{TV}}(q_{\omega_k}^{\text{BH}}, d_{\theta_{k-i}})$, captures the mismatch between the mixture state occupancy and the individual occupancies and similarly is non-decreasing in ω . Crucially, both the second and third penalties are independent of θ and vanish when taking the gradient of the surrogate loss, thus (i) grounding $\hat{L}^{\text{BH}}$ and (ii) suggesting that the policies in the window should remain close to each other, while the first penalty, which depends on θ , requires the target policy to remain close to the mixture $\pi_{\omega_k}^{\text{BH}}$.

4 Experiments

We provide an experimental validation of the effects of sample reuse, implemented via ω PPO-U and ω PPO-BH, on PPO. Here, we report only the most interesting results; an extended experimental campaign across diverse MuJoCo [57] tasks is provided in Appendix D. To ensure a fair comparison, we always start from the *tuned* PPO configuration of [47], and the reuse variants all share common hyperparameters unless otherwise specified. All experimental details are deferred to Appendix C.

Comparison Settings. When comparing ω PPO-U and ω PPO-BH against PPO, several experimental settings are possible. In this section we consider the following two:

- **Fixed mini-batch size:** the reuse variants share all hyperparameters with PPO ($\tilde{K}$, B , and $N_E H$). Since $|\mathcal{D}(\omega_k)| = \omega_k D$, fixing B means the reuse variants perform ω_k times as many actor updates per iteration as PPO, with each datum having a maximum life cycle of $\omega_k \tilde{K}$ actor updates. Each mini-batch contains B/ω_k samples per window, so each update sees a smaller fraction of fresh data than PPO. To isolate the effect of data reuse from the effect of additional updates, we also run PPO with $\omega_k \tilde{K}$ epochs per iteration, matching the total update count of the reuse variants.
- **Fixed mini-batch count:** ω PPO-U and ω PPO-BH keep the same number of mini-batches N_B as PPO, yielding larger mini-batches of size $\omega_k B$. Each update thus sees the same amount of fresh data as PPO, while the reuse variants additionally incorporate past samples into each mini-batch. All methods therefore perform the same number of updates per iteration, isolating the effect of additional data from that of additional gradient steps.

These settings directly measure sample efficiency: given the same number of environment interactions, they test how well each method exploits the available information. Both settings are designed to maintain fairness with respect to PPO and to isolate distinct aspects of the data reuse mechanism.

Fixed B . Figure 1 shows training evaluation curves across four tasks, comparing PPO, ω PPO-U, ω PPO-BH, and a PPO variant running $\omega \tilde{K}$ epochs per iteration. The latter collapses in almost every setting, while the reuse variants, whose mini-batches come from $\mathcal{D}(\omega_k)$, do not exhibit the same instability. At $\omega = 4$, both ω PPO-U and ω PPO-BH consistently outperform PPO in terms of final performance of the deployed policy (Table 1), suggesting that the benefit of data reuse is not merely due to performing more updates. However, this trend does not hold uniformly across tasks when enlarging ω . The *Hopper-v5* environment is a pathological case: at $\omega = 8$, the reuse

Algorithm		Reacher-v5		Walker2d-v5		Hopper-v5		HalfCheetah-v5	
		Fixed B	Fixed N_B	Fixed B	Fixed N_B	Fixed B	Fixed N_B	Fixed B	Fixed N_B
PPO		-4.35 ± 0.063		2430 ± 443		2530 ± 137		4330 ± 725	
PPO ($\omega \tilde{K}$ epochs)	$2\times$	-4.70 ± 0.069	—	2510 ± 556	—	2790 ± 305	—	4500 ± 1120	—
	$4\times$	-6.76 ± 0.308	—	682 ± 138	—	1190 ± 442	—	4840 ± 861	—
	$8\times$	-7.97 ± 0.531	—	392 ± 32.4	—	415 ± 317	—	3940 ± 810	—
ω PPO-U	$\omega = 2$	-4.29 ± 0.065	-4.22 ± 0.060	3350 ± 556	3340 ± 535	3240 ± 94.3	2680 ± 163	5560 ± 335	5160 ± 515
	$\omega = 4$	-4.25 ± 0.075	-4.17 ± 0.063	3740 ± 546	2820 ± 485	3220 ± 192	2810 ± 157	5960 ± 729	5760 ± 569
	$\omega = 8$	-4.15 ± 0.057	-4.05 ± 0.057	3650 ± 426	3030 ± 430	2130 ± 195	2750 ± 131	5870 ± 497	5360 ± 384
ω PPO-BH	$\omega = 2$	-4.14 ± 0.047	-4.12 ± 0.052	3360 ± 491	3250 ± 530	3220 ± 94.6	2740 ± 128	5530 ± 550	4580 ± 949
	$\omega = 4$	-3.99 ± 0.045	-4.01 ± 0.045	3230 ± 735	3810 ± 642	3110 ± 274	2920 ± 159	6170 ± 444	5780 ± 558
	$\omega = 8$	-3.94 ± 0.052	-3.96 ± 0.054	2660 ± 815	2990 ± 375	2120 ± 476	2910 ± 177	5720 ± 586	5240 ± 447

Table 1: Mean final performance (average of last 10 evaluations, 10 runs $\pm 95\%$ C.I.) for the settings of Figures 1–2. Colored cells: best mean performance; bold: methods whose C.I. intersects the best.

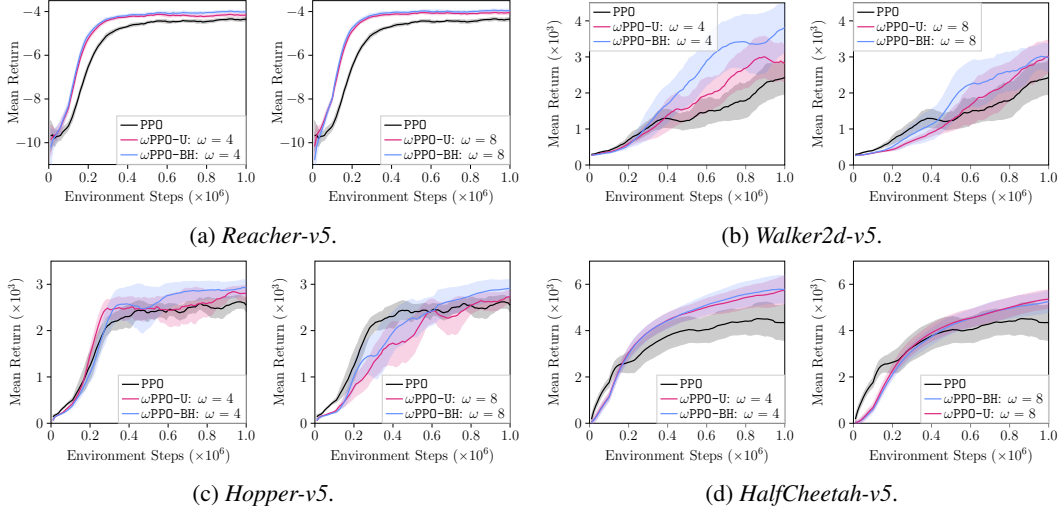


Figure 2: PPO, ω PPO-U, and ω PPO-BH in the *fixed* N_B setting (10 runs, mean $\pm 95\%$ C.I.).

variants not only fail to outperform PPO but exhibit degraded training. We conjecture this results from several compounding factors, amplified by this environment’s configuration ($H = 2048$, $B = 64$, $\tilde{K} = 10$). First, at $\omega = 8$ the reuse variants perform 2560 updates per iteration against the 320 of PPO, potentially amplifying the bias introduced by stale advantages across many gradient steps. Second, with $B/\omega = 8$ fresh samples per mini-batch, the fraction of on-policy signal per update is particularly small. Third, since all hyperparameters are inherited from PPO, the larger number of updates may increase the drift between consecutive policies and hence the diversity of the policies in the window, possibly making past samples falling outside the clipping range more frequently. All in all, in this setting PPO consistently benefits from data reuse when the reuse variants are allowed to perform ω times as many updates as PPO, up to a moderate window size, beyond which the factors above may hinder training.

Fixed N_B . Figure 2 shows training evaluation curves across the same four environments, comparing PPO, ω PPO-U, and ω PPO-BH. We recall that in this setting all methods perform the same number of updates per iteration and each update sees the same amount of fresh data, while the reuse variants employ mini-batches of size ωB . A consistent trend is easier to identify here: PPO benefits from sample reuse across all environments, despite the update count being matched. Most of the benefits are observable at $\omega = 4$, particularly under ω PPO-BH, as further confirmed by Table 1. Increasing to $\omega = 8$ yields diminishing returns, with the reuse variants either matching or only marginally improving over PPO. We conjecture this is due to the increasing proportion of stale advantages and of samples collected under increasingly distant policies within each mini-batch, which gradually offsets the benefit of additional data as the window grows.

ω PPO-U vs. ω PPO-BH. In general, neither weighting scheme consistently dominates the other: the best-performing policy is achievable with both, given an appropriate ω and environment (Table 1). However, the two schemes exploit past data differently. In ω PPO-U, each sample is weighted

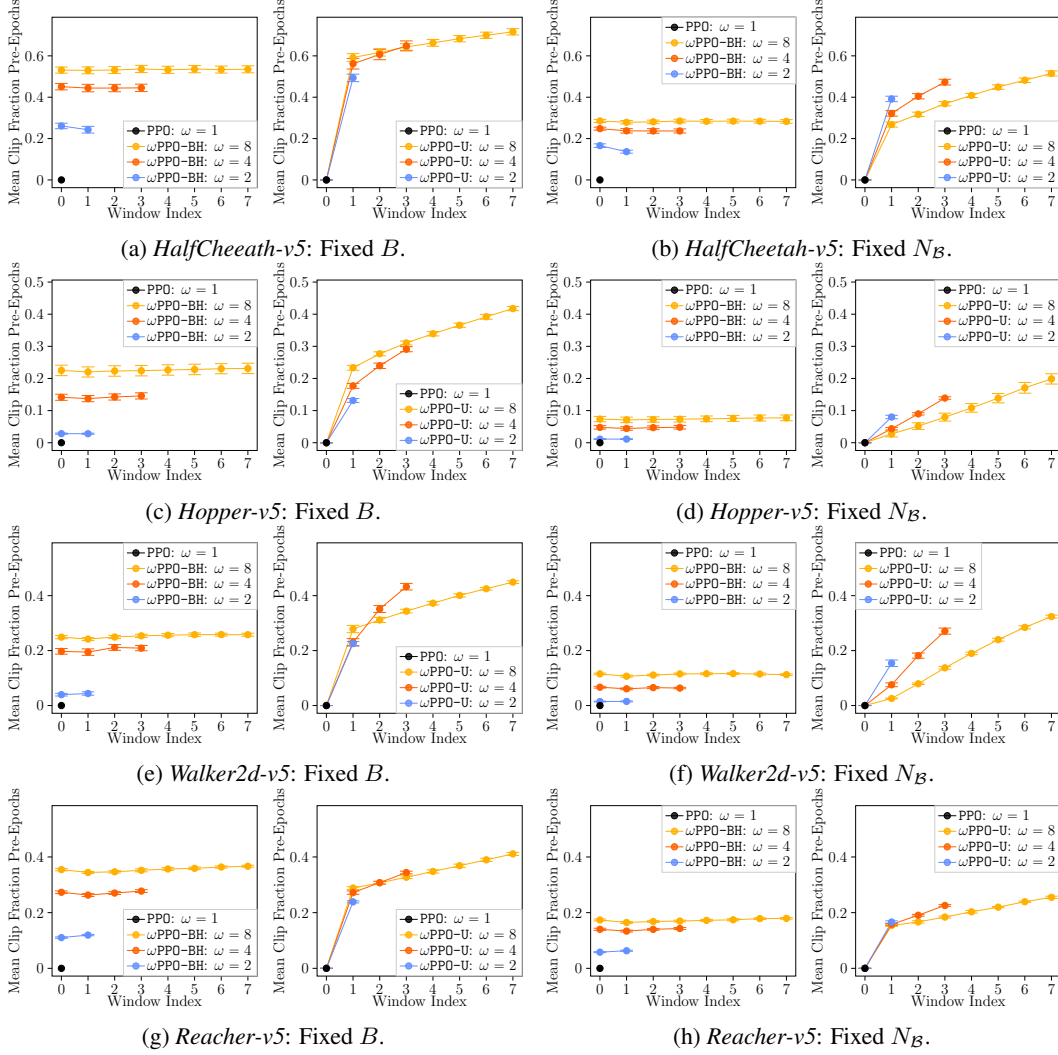


Figure 3: Clip ratio diagnostics (10 runs, mean $\pm$ 95% C.I.).

against its own behavioral policy $\pi_{\theta_{k-i}}$, so the ratio deviates further from 1 for older window indices, causing $\omega\text{PPO-U}$ to clip an increasing fraction of past data as ω grows. In $\omega\text{PPO-BH}$, the mixture denominator $\pi_{\omega_k}^{\text{BH}}$ has two consequences: (i) at the beginning of each iteration, the IWs are bounded by ω_k , which limits how far above the clipping range older samples can start; and (ii) the IWs of the most recent samples generally differ from 1, so that, at small ϵ and large ω , part of the fresh data may be clipped from the first update. Figure 3 reports the average pre-optimization clip fractions, grouped by window index: $\omega\text{PPO-BH}$ maintains a lower pre-optimization clip fraction for older windows, consistently with (i), while $\omega\text{PPO-U}$ exhibits pre-optimization clip fractions that grow with the window index, confirming that older samples contribute increasingly less.

Main Takeaways. We now summarize when and how PPO may benefit from data reuse, based on the results of this section and of Appendix D. We recall that this paper studies data reuse per se: feeding PPO with a composite buffer $\mathcal{D}(\omega_k)$ from the most recent ω iterations, either with vanilla importance weights ($\omega\text{PPO-U}$) or with a BH correction ($\omega\text{PPO-BH}$). First, PPO benefits from sample reuse in both considered settings (fixed B and fixed N_B) and the benefit *stems from the additional data* rather than from the increased number of updates alone. Second, the impact of reuse manifests *primarily in the final performance* of the deployed policy, while *improvements in convergence speed* also occur but are closely tied to the environment and hyperparameter configuration, appearing consistently only in *Reacher-v5*, the simplest here, and *Ant-v5* (see Appendix D). Third, in most environments, reuse is most effective at moderate window sizes (e.g., $\omega = 4$). With larger windows, particularly in the fixed B setting, *stale advantages*, the *increased diversity of the policies* in the

window, and the *dilution of fresh data* may offset the benefits of the additional data. Finally, while both $\omega_{\text{PPO-U}}$ and $\omega_{\text{PPO-BH}}$ are beneficial depending on the task and hyperparameters, they exploit past data differently: $\omega_{\text{PPO-BH}}$ clips a smaller fraction of older samples thanks to its mixture denominator but may clip part of the most recent ones from the start, while $\omega_{\text{PPO-U}}$ fully exploits fresh data at the cost of clipping a larger fraction of older samples.

5 Conclusion

We studied sample reuse in PPO within a MIW framework, deliberately preserving its core mechanics to isolate the effect of data reuse from other factors. We instantiated $\omega_{\text{PPO-U}}$ and $\omega_{\text{PPO-BH}}$, employing uniform and BH-corrected IWs, respectively, and derived policy improvement lower bounds for both. Beyond grounding their losses, the bounds reveal penalties uncontrollable by the actor updates, suggesting that the policies in the reuse window should remain close to each other. Empirically, we found that moderate data reuse consistently improves the final performances across diverse control tasks, while excessive reuse can be harmful. Neither $\omega_{\text{PPO-U}}$ nor $\omega_{\text{PPO-BH}}$ consistently dominates the other, though $\omega_{\text{PPO-BH}}$ may better exploit older samples at the cost of potentially clipping part of the fresh data. Future work should extend this analysis to other deep on-policy methods [e.g., 56, 53], and systematically isolate the individual contributions of advantage re-evaluation, modified clipping, and other heuristics introduced by existing PPO reuse variants [e.g., 46, 33, 69].

References

- [1] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International conference on machine learning*, pages 22–31. Pmlr, 2017.
- [2] Alekh Agarwal, Nan Jiang, Sham M Kakade, and Wen Sun. Reinforcement learning: Theory and algorithms. *CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep*, 32:96, 2019.
- [3] Firas Al-Hafez, Guoping Zhao, Jan Peters, and Davide Tateo. Locomujoco: A comprehensive imitation learning benchmark for locomotion. In *6th Robot Learning Workshop, NeurIPS*, 2023.
- [4] 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*, 2021.
- [5] Kamyar Azizzadenesheli, Yisong Yue, and Animashree Anandkumar. Policy gradient in partially observable environments: Approximation and convergence. *arXiv preprint arXiv:1810.07900*, 2018.
- [6] Jonathan Baxter and Peter L Bartlett. Infinite-horizon policy-gradient estimation. *journal of artificial intelligence research*, 15:319–350, 2001.
- [7] Dieter Büchler, Simon Guist, Roberto Calandra, Vincent Berenz, Bernhard Schölkopf, and Jan Peters. Learning to play table tennis from scratch using muscular robots. *IEEE Transactions on Robotics*, 38(6):3850–3860, 2022.
- [8] Yuhu Cheng, Qingbang Guo, and Xuesong Wang. Proximal policy optimization with advantage reuse competition. *IEEE Transactions on Artificial Intelligence*, 5(8):3915–3925, 2024.
- [9] Marc Peter Deisenroth, Gerhard Neumann, Jan Peters, et al. A survey on policy search for robotics. *Foundations and Trends® in Robotics*, 2(1–2):1–142, 2013.
- [10] Leif Doering, Daniel Schmidt, Moritz Melcher, Sebastian Kassing, Benedikt Wille, Tilman Aach, and Simon Weissmann. An approximate ascent approach to prove convergence of ppo. *arXiv preprint arXiv:2602.03386*, 2026.
- [11] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In *International conference on machine learning*, pages 1329–1338. PMLR, 2016.

- [12] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep rl: A case study on ppo and trpo. In *International conference on learning representations*, 2019.
- [13] Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International conference on machine learning*, pages 1407–1416. PMLR, 2018.
- [14] Rasool Fakoor, Pratik Chaudhari, and Alexander J. Smola. P3O: policy-on policy-off policy optimization. In Amir Globerson and Ricardo Silva, editors, *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, Proceedings of Machine Learning Research, pages 1017–1027. AUAI Press, 2019.
- [15] Ilyas Fatkhullin, Anas Barakat, Anastasia Kireeva, and Niao He. Stochastic policy gradient methods: Improved sample complexity for fisher-non-degenerate policies. In *International Conference on Machine Learning*, pages 9827–9869. PMLR, 2023.
- [16] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.
- [17] Mohammad Ghavamzadeh and Yaakov Engel. Bayesian policy gradient algorithms. *Advances in neural information processing systems*, 19, 2006.
- [18] Benjamin Gravell, Peyman Mohajerin Esfahani, and Tyler Summers. Learning optimal controllers for linear systems with multiplicative noise via policy gradient. *IEEE Transactions on Automatic Control*, 66(11):5283–5298, 2020.
- [19] Shixiang Gu, Timothy Lillicrap, Zoubin Ghahramani, Richard E Turner, and Sergey Levine. Q-prop: Sample-efficient policy gradient with an off-policy critic. *arXiv preprint arXiv:1611.02247*, 2016.
- [20] Shixiang Shane Gu, Timothy Lillicrap, Richard E Turner, Zoubin Ghahramani, Bernhard Schölkopf, and Sergey Levine. Interpolated policy gradient: Merging on-policy and off-policy gradient estimation for deep reinforcement learning. *Advances in neural information processing systems*, 30, 2017.
- [21] 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*, pages 1861–1870. Pmlr, 2018.
- [22] Ruinan Jin, Shuai Li, and Baoxiang Wang. On stationary point convergence of ppo-clip. In *The Twelfth International Conference on Learning Representations*, 2023.
- [23] Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the nineteenth international conference on machine learning*, pages 267–274, 2002.
- [24] Mahdi Kallel, Jose-Luis Holgado-Alvarez, Samuele Tosatto, and Carlo D’Eramo. Revisiting proximal policy optimization. In *Eighteenth European Workshop on Reinforcement Learning*, 2025.
- [25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [26] Augustine Kong. A note on importance sampling using standardized weights. *University of Chicago, Dept. of Statistics, Tech. Rep.*, 348:14, 1992.
- [27] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.
- [28] Xingxing Liang, Yang Ma, Yanghe Feng, and Zhong Liu. PTR-PPO: proximal policy optimization with prioritized trajectory replay. *CoRR*, abs/2112.03798, 2021.

- [29] 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, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.
- [30] Yifan Lin, Yuhao Wang, and Enlu Zhou. Reusing historical trajectories in natural policy gradient via importance sampling: Convergence and convergence rate. *Operations Research*, 73(6): 3010–3026, 2025.
- [31] Qisai Liu, Zhanhong Jiang, Hsin-Jung Yang, Mahsa Khosravi, Joshua R Waite, and Soumik Sarkar. Enhancing ppo with trajectory-aware hybrid policies. *IEEE Transactions on Neural Networks and Learning Systems*, 2026.
- [32] Gabriel B Margolis, Ge Yang, Kartik Paigwar, Tao Chen, and Pulkit Agrawal. Rapid locomotion via reinforcement learning. *The International Journal of Robotics Research*, 43(4):572–587, 2024.
- [33] Wenjia Meng, Qian Zheng, Gang Pan, and Yilong Yin. Off-policy proximal policy optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 9162–9170, 2023.
- [34] Alberto Maria Metelli, Matteo Papini, Francesco Faccio, and Marcello Restelli. Policy optimization via importance sampling. *Advances in neural information processing systems*, 31, 2018.
- [35] Alberto Maria Metelli, Matteo Papini, Nico Montali, and Marcello Restelli. Importance sampling techniques for policy optimization. *Journal of Machine Learning Research*, 21(141): 1–75, 2020.
- [36] Alessandro Montenegro, Federico Mansutti, Marco Mussi, Matteo Papini, and Alberto Maria Metelli. Reusing trajectories in policy gradients enables fast convergence. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, 2026.
- [37] Art Owen and Yi Zhou. Safe and effective importance sampling. *Journal of the American Statistical Association*, 95(449):135–143, 2000.
- [38] Art B. Owen. *Monte Carlo Theory, Methods and Examples*. 2013.
- [39] Gabor Paczolay, Matteo Papini, Alberto Maria Metelli, Istvan Harmati, and Marcello Restelli. Sample complexity of variance-reduced policy gradient: weaker assumptions and lower bounds. *Machine Learning*, 113(9):6475–6510, 2024.
- [40] Matteo Papini, Damiano Binaghi, Giuseppe Canonaco, Matteo Pirota, and Marcello Restelli. Stochastic variance-reduced policy gradient. In *International conference on machine learning*, pages 4026–4035. PMLR, 2018.
- [41] Matteo Papini, Alberto Maria Metelli, Lorenzo Lupo, and Marcello Restelli. Optimistic policy optimization via multiple importance sampling. In *International Conference on Machine Learning*, pages 4989–4999. PMLR, 2019.
- [42] Matteo Papini, Giorgio Manganini, Alberto Maria Metelli, and Marcello Restelli. Policy gradient with active importance sampling. *arXiv preprint arXiv:2405.05630*, 2024.
- [43] Jan Peters and Stefan Schaal. Policy gradient methods for robotics. In *2006 IEEE/RSJ international conference on intelligent robots and systems*, pages 2219–2225. IEEE, 2006.
- [44] Matteo Pirota, Marcello Restelli, Alessio Pecorino, and Daniele Calandriello. Safe policy iteration. In *International conference on machine learning*, pages 307–315. PMLR, 2013.
- [45] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. John Wiley & Sons, 2014.

- [46] James Queeney, Yannis Paschalidis, and Christos G Cassandras. Generalized proximal policy optimization with sample reuse. *Advances in Neural Information Processing Systems*, 34: 11909–11919, 2021.
- [47] Antonin Raffin. RL baselines3 zoo. <https://github.com/DLR-RM/rl-baselines3-zoo>, 2020.
- [48] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.
- [49] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.
- [50] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.
- [51] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.
- [52] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [53] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
- [54] Mingfei Sun, Vitaly Kurin, Guoqing Liu, Sam Devlin, Tao Qin, Katja Hofmann, and Shimon Whiteson. You may not need ratio clipping in ppo. *arXiv preprint arXiv:2202.00079*, 2022.
- [55] Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.
- [56] 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.
- [57] 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*, pages 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.
- [58] Eric Veach and Leonidas J Guibas. Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 419–428, 1995.
- [59] Huayi Wang, Zirui Wang, Junli Ren, Qingwei Ben, Tao Huang, Weinan Zhang, and Jiangmiao Pang. Beamdojo: Learning agile humanoid locomotion on sparse footholds. *arXiv preprint arXiv:2502.10363*, 2025.
- [60] Yuhui Wang, Hao He, Xiaoyang Tan, and Yaozhong Gan. Trust region-guided proximal policy optimization. *Advances in Neural Information Processing Systems*, 32, 2019.
- [61] Yuhui Wang, Hao He, and Xiaoyang Tan. Truly proximal policy optimization. In *Uncertainty in artificial intelligence*, pages 113–122. PMLR, 2020.
- [62] Zhichao Wang, Bin Bi, Shiva Kumar Pentyala, Kiran Ramnath, Sougata Chaudhuri, Shubham Mehrotra, Xiang-Bo Mao, Sitaram Asur, et al. A comprehensive survey of llm alignment techniques: RLhf, rlaiif, ppo, dpo and more. *arXiv preprint arXiv:2407.16216*, 2024.

- [63] Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando De Freitas. Sample efficient actor-critic with experience replay. *arXiv preprint arXiv:1611.01224*, 2016.
- [64] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.
- [65] Pan Xu, Felicia Gao, and Quanquan Gu. Sample efficient policy gradient methods with recursive variance reduction. *arXiv preprint arXiv:1909.08610*, 2019.
- [66] Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. Is dpo superior to ppo for llm alignment? a comprehensive study. *arXiv preprint arXiv:2404.10719*, 2024.
- [67] Huizhuo Yuan, Xiangru Lian, Ji Liu, and Yuren Zhou. Stochastic recursive momentum for policy gradient methods. *arXiv preprint arXiv:2003.04302*, 2020.
- [68] Rui Yuan, Robert M Gower, and Alessandro Lazaric. A general sample complexity analysis of vanilla policy gradient. In *International Conference on Artificial Intelligence and Statistics*, pages 3332–3380. PMLR, 2022.
- [69] Li-yang Zhao, Tian-qing Chang, Jie Zhang, Lei Zhang, Kai-xuan Chu, Li-bin Guo, and De-peng Kong. A policy optimization algorithm based on sample adaptive reuse and dual-clipping for robotic action control. *Applied Soft Computing*, 134:109967, 2023.

A Algorithmic Details: PPO and Data Collection

In this section, we provide details on PPO, remaining tied to its common practical implementation [e.g., 48].

Algorithm 1: On-Policy Data Collection.

Input : Actor π_θ , critic V_w , interaction length H , environments $\{E_e\}_{e=1}^{N_E}$, GAE parameter λ , discount factor γ .

```

1 Initialize global buffer  $\mathcal{D} \leftarrow \emptyset$ ;
2 for  $e \in \llbracket N_E \rrbracket$  do
3   Initialize local buffer  $\mathcal{D}_{\text{local}} \leftarrow \emptyset$ ;
4   Receive initial state  $\mathbf{s} \leftarrow \mathbf{s}_1$  from  $E_e$ ;
   // Interaction
5   for  $h \in \llbracket H \rrbracket$  do
6     Set  $\mathbf{s}_h \leftarrow \mathbf{s}$ ;
7     Sample  $\mathbf{a}_h \sim \pi_\theta(\cdot \mid \mathbf{s}_h)$  and execute it in  $E_e$ ;
8     Observe  $r_h, \mathbf{s}_{h+1}$ , and done flag  $\xi_h$ ;
9     Compute value estimate  $v_h = V_w(\mathbf{s}_h)$  and  $\rho_h = \pi_\theta(\mathbf{a}_h \mid \mathbf{s}_h)$ ;
10    Store tuple  $(\mathbf{s}_h, \mathbf{a}_h, r_h, v_h, \rho_h, \xi_h)$  in  $\mathcal{D}_{\text{local}}$ ;
11    if  $\xi_h$  is True then
12      Reset  $E_e$  and receive new  $\mathbf{s}_{h+1}$ ;
13    end
14    Set  $\mathbf{s} \leftarrow \mathbf{s}_{h+1}$ ;
15  end
  // GAE Calculation taking samples from  $\mathcal{D}_{\text{local}}$ 
16  Compute the last value function  $v_{H+1} = V_w(\mathbf{s}_{H+1})$ ;
17  Initialize next advantage  $A_{H+1} = 0$ ;
18  for  $h \in \llbracket H \rrbracket_\downarrow$  do
19     $\delta_h = r_h + \gamma(1 - \xi_h)v_{h+1} - v_h$ ;
20     $A_h = \delta_h + \gamma\lambda(1 - \xi_h)A_{h+1}$ ;
21     $R_h = A_h + v_h$ ;
22    Add  $(\mathbf{s}_h, \mathbf{a}_h, \rho_h, A_h, R_h)$  to  $\mathcal{D}$ ;
23  end
24 end
Output : Buffer  $\mathcal{D}$ .

```

A.1 On-Policy Data Collection

We begin by detailing the data collection process. Unlike the theoretical formulation based on complete episodes, practical implementations [48] interact with the environment for a *fixed* number of timesteps $H \in \mathbb{N}$, regardless of the task’s actual horizon. This fixed-length rollout is adopted primarily for computational efficiency and memory management. To handle episode termination, we employ a boolean flag ξ_h , which is true if the environment enters an absorbing state at timestep h . Crucially, if the rollout ends at timestep H without reaching an absorbing state (truncation), the value of the last state is used for bootstrapping.

For the sake of generality, we consider a set of parallel environments $\{E_e\}_{e=1}^{N_E}$ collecting independent trajectory segments of length H . During interaction, the agent stores the essential data required for the optimization phase: the state-action pairs $(\mathbf{s}_h, \mathbf{a}_h)$, the rewards r_h , the value estimates $V_w(\mathbf{s}_h)$, and the probabilities $\pi_\theta(\mathbf{a}_h \mid \mathbf{s}_h)$. After collection, the algorithm computes the advantage estimates A_h and the return targets R_h . The standard approach is Generalized Advantage Estimation [GAE, 51], which balances the bias-variance trade-off akin to TD(λ). While a Monte Carlo estimate is unbiased but suffers from high variance, a TD(0) approach has low variance but introduces significant bias due to bootstrapping. GAE interpolates between these regimes. Formally, let $\delta_h = r_h + \gamma(1 - \xi_h)V_w(\mathbf{s}_{h+1}) - V_w(\mathbf{s}_h)$ be the TD-error at step h . The GAE estimate is defined as the exponentially weighted sum of these errors:

$$A_h = \sum_{k=0}^{H-h-1} (\gamma\lambda)^k \delta_{h+k},$$

where $\lambda \in [0, 1]$ controls the trade-off, and the sum is truncated at the rollout boundary H (handling the last state via bootstrapping as previously described). Algorithm 1 details the data collection. There, for $n, m \in \mathbb{N}$ with $n \geq m$, we denote $\llbracket n, m \rrbracket_{\downarrow} := \{n, n-1, \dots, m\}$ and $\llbracket n \rrbracket_{\downarrow} := \llbracket n, 1 \rrbracket_{\downarrow}$.

Algorithm 2: PPO [52].

Input : Initial parameters θ_0 and w_0 , iterations K , epochs per iteration $\tilde{K}$, batch size B , environments $\{E_e\}_{e=1}^{N_E}$, GAE parameter λ , interaction length H , discount factor γ , loss coefficients c_1 and c_2 , learning rate α , clipping parameter ϵ .

1 Initialize optimization variables: $\theta \leftarrow \theta_0, w \leftarrow w_0$.

2 **for** $k \in \llbracket 1, K \rrbracket$ **do**

// Data Collection

3 $\mathcal{D}_k \leftarrow \text{Run Algorithm 1};$

// Multi-Epoch Optimization

4 **for** $j \in \llbracket \tilde{K} \rrbracket$ **do**

5 Shuffle $\mathcal{D}_k$;

6 Create $\tilde{\mathcal{D}}_k = \{\mathcal{B}_z\}_{z=1}^{N_{\mathcal{B}}}$, where $N_{\mathcal{B}} = \lceil \frac{|\mathcal{D}_k|}{B} \rceil$, such that $\forall z \in \llbracket N_{\mathcal{B}} \rrbracket : |\mathcal{B}_z| = B$;

7 **for** $\mathcal{B} \in \tilde{\mathcal{D}}_k$ **do**

// Compute Loss

8
$$\hat{L}^{\text{PPO}}(\theta, w, \mathcal{B}) = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} L_i^{\text{clip}}(\theta) - c_1 L_i^{\text{vf}}(w) + c_2 \mathcal{H}(\pi_{\theta}(\cdot | s_i)),$$

where

$$L_i^{\text{clip}}(\theta) = \min \left\{ \frac{\pi_{\theta}(\mathbf{a}_i | s_i)}{\pi_{\theta_i}(\mathbf{a}_i | s_i)} A_i, \text{clip} \left(\frac{\pi_{\theta}(\mathbf{a}_i | s_i)}{\pi_{\theta_i}(\mathbf{a}_i | s_i)}, 1 - \epsilon, 1 + \epsilon \right) A_i \right\};$$

$$L_i^{\text{vf}}(w) = \frac{1}{2} (V_w(s_i) - R_i)^2.$$

// Update

9
$$\theta \leftarrow \theta + \alpha \nabla_{\theta} \hat{L}(\theta, w) \quad \text{and} \quad w \leftarrow w + \alpha \nabla_w \hat{L}(\theta, w).$$

10 **end**

11 **end**

12 **end**

Output : θ_K .

A.2 PPO Details

As highlighted in Section 1, PPO [52] is one of the most used and successful RL methods. In order to extend it in a meaningful way, we detail here an (almost) practical version of it. Before proceeding, we need to introduce additional notation. Let $L^{\text{PPO}} : \Theta \times \mathcal{W} \rightarrow \mathbb{R}$ be a joint loss for both the actor and the critic, defined as:

$$\begin{aligned} L^{\text{PPO}}(\theta, w) = & \mathbb{E}_{\substack{s \sim d_{\theta_b} \\ \mathbf{a} \sim \pi_{\theta_b}(\cdot | s)}} \left[\min \left\{ \frac{\pi_{\theta}(\mathbf{a} | s)}{\pi_{\theta_b}(\mathbf{a} | s)} A^{\pi_{\theta_b}}(s, \mathbf{a}), \text{clip} \left(\frac{\pi_{\theta}(\mathbf{a} | s)}{\pi_{\theta_b}(\mathbf{a} | s)}, 1 - \epsilon, 1 + \epsilon \right) A^{\pi_{\theta_b}}(s, \mathbf{a}) \right\} \right] \\ & - \frac{c_1}{2} \mathbb{E}_{s \sim d_{\theta_b}} \left[(V_w(s) - V^{\pi_{\theta_b}}(s))^2 \right] \\ & + c_2 \mathbb{E}_{s \sim d_{\theta_b}} [\mathcal{H}(\pi_{\theta}(\cdot | s))], \end{aligned}$$

where π_{θ_b} is a behavioral policy used to collect data, where $\mathcal{H}(\pi_{\theta}(\cdot | s)) := \mathbb{E}_{\mathbf{a} \sim \pi_{\theta}} [-\log \pi_{\theta}(\mathbf{a} | s)]$ is the entropy of the policy π_{θ} in state s , and where $c_1, c_2 \in \mathbb{R}_{\geq 0}$ are weighting constants.

Now, let $\mathcal{B} = \{(s_i, \mathbf{a}_i, \rho_i, A_i, R_i)\}_{i=1}^M$ be a batch of M interaction tuples, where A_i and R_i are estimates of the advantage function and the return-to-go, both w.r.t. the policy used to collect data,

respectively. We can define the sample-based version of L^{ppo} as follows:

$$\begin{aligned}\hat{L}^{\text{ppo}}(\theta, w, \mathcal{B}) = & \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \min \left\{ \frac{\pi_{\theta}(\mathbf{a}_i | \mathbf{s}_i)}{\pi_{\theta_i}(\mathbf{a}_i | \mathbf{s}_i)} A_i, \text{clip} \left(\frac{\pi_{\theta}(\mathbf{a}_i | \mathbf{s}_i)}{\pi_{\theta_i}(\mathbf{a}_i | \mathbf{s}_i)}, 1 - \epsilon, 1 + \epsilon \right) A_i \right\} \\ & - \frac{c_1}{2} (V_w(\mathbf{s}_i) - R_i)^2 + c_2 \mathcal{H}(\pi_{\theta}(\cdot | \mathbf{s}_i)).\end{aligned}$$

That being said, the parameters can then be updated as:

$$\theta \leftarrow \theta + \alpha \nabla_{\theta} \hat{L}^{\text{ppo}}(\theta, w, \mathcal{B}) \quad \text{and} \quad w \leftarrow w + \alpha \nabla_w \hat{L}^{\text{ppo}}(\theta, w, \mathcal{B}),$$

where α is the step size, usually managed by Adam [25] or by other schedulers (e.g., by linearly decreasing the step size or adapting the step size in order to maintain a certain KL divergence among policies across iterations [48, 3, 46]). Notice that, in many implementations, the *critic* loss is clipped too. The detailed pseudo-code is reported in Algorithm 2.

B Proofs

Lemma B.1 (Performance Difference Lemma [Lemma 6.1, 23]). *For every $\theta_1, \theta_2 \in \Theta$, the following holds:*

$$J(\theta_1) - J(\theta_2) = \mathbb{E}_{\substack{\mathbf{s} \sim d_{\theta_1} \\ \mathbf{a} \sim \pi_{\theta_1}(\cdot | \mathbf{s})}} [A^{\pi_{\theta_2}}(\mathbf{s}, \mathbf{a})].$$

Lemma B.2 (Policy Improvement Lower Bound [Corollary 1, 1]). *For every $\theta_1, \theta_2 \in \Theta$, the following holds:*

$$J(\theta_1) - J(\theta_2) \geq \mathbb{E}_{\substack{\mathbf{s} \sim d_{\theta_2} \\ \mathbf{a} \sim \pi_{\theta_2}(\cdot | \mathbf{s})}} \left[\frac{\pi_{\theta_1}(\mathbf{a} | \mathbf{s})}{\pi_{\theta_2}(\mathbf{a} | \mathbf{s})} A^{\pi_{\theta_2}}(\mathbf{s}, \mathbf{a}) \right] - \frac{2\gamma C(\theta_1, \theta_2)}{(1 - \gamma)^2} \mathbb{E}_{\mathbf{s} \sim d_{\theta_2}} [D_{\text{TV}}(\pi_{\theta_1}(\cdot | \mathbf{s}), \pi_{\theta_2}(\cdot | \mathbf{s}))],$$

where $C(\theta_1, \theta_2) := \sup_{\mathbf{s} \in \mathcal{S}} \left| \mathbb{E}_{\mathbf{a} \sim \pi_{\theta_1}(\cdot | \mathbf{s})} [A^{\pi_{\theta_2}}(\mathbf{s}, \mathbf{a})] \right|$.

Lemma B.3. *For every $\theta, \theta_1, \theta_2 \in \Theta$, the following holds:*

$$\left| \mathbb{E}_{\substack{\mathbf{s} \sim d_{\theta_2} \\ \mathbf{a} \sim \pi_{\theta}(\cdot | \mathbf{s})}} [A^{\pi_{\theta_1}}(\mathbf{s}, \mathbf{a}) - A^{\pi_{\theta_2}}(\mathbf{s}, \mathbf{a})] \right| \leq \frac{4R_{\max}(1 + \gamma)}{1 - \gamma} \sup_{\mathbf{s} \in \mathcal{S}} D_{\text{TV}}(\pi_{\theta_1}(\cdot | \mathbf{s}), \pi_{\theta_2}(\cdot | \mathbf{s})).$$

Proof. To start the proof, we can simply note that:

$$\begin{aligned}& \left| \mathbb{E}_{\substack{\mathbf{s} \sim d_{\theta_2} \\ \mathbf{a} \sim \pi_{\theta}(\cdot | \mathbf{s})}} [A^{\pi_{\theta_1}}(\mathbf{s}, \mathbf{a}) - A^{\pi_{\theta_2}}(\mathbf{s}, \mathbf{a})] \right| \\ & \leq \mathbb{E}_{\substack{\mathbf{s} \sim d_{\theta_2} \\ \mathbf{a} \sim \pi_{\theta}(\cdot | \mathbf{s})}} [|A^{\pi_{\theta_1}}(\mathbf{s}, \mathbf{a}) - A^{\pi_{\theta_2}}(\mathbf{s}, \mathbf{a})|] \\ & \leq \underbrace{\mathbb{E}_{\substack{\mathbf{s} \sim d_{\theta_2} \\ \mathbf{a} \sim \pi_{\theta}(\cdot | \mathbf{s})}} [|Q^{\pi_{\theta_1}}(\mathbf{s}, \mathbf{a}) - Q^{\pi_{\theta_2}}(\mathbf{s}, \mathbf{a})|]}_{=: \text{A}} + \underbrace{\mathbb{E}_{\mathbf{s} \sim d_{\theta_2}} [|V^{\pi_{\theta_1}}(\mathbf{s}) - V^{\pi_{\theta_2}}(\mathbf{s})|]}_{=: \text{B}},\end{aligned}$$

where we employed the Jensen's inequality, then the advantage function definition, and finally we used the triangular inequality and the linearity of the expectation. Notice that in the **B** term we lose the action sampling from π_{θ} .

Bounding the Term B. We now focus on term **B**. Before starting with this part, let us define for every $\mathbf{s}, \mathbf{s}_0 \in \mathcal{S}$ and $\theta \in \Theta$ the following:

$$d_{\theta}^{\mathbf{s}_0}(\mathbf{s}) := \sum_{t=1}^T \gamma^{t-1} \Pr(\mathbf{s}_t = \mathbf{s} | \mathbf{s}_1 = \mathbf{s}_0, \pi_{\theta}),$$

which is the γ -discounted state s visitation under policy π_θ and starting from state s_0 . Notice that it holds $d_\theta(s) = \mathbb{E}_{s_0 \sim d_0} [d_\theta^{s_0}(s)]$. Specifically, considering the argument of the expectation, we have the following:

$$\begin{aligned} V^{\pi_{\theta_1}}(s) - V^{\pi_{\theta_2}}(s) &= \mathbb{E}_{\substack{s' \sim d_{\theta_1}^s \\ \mathbf{a} \sim \pi_{\theta_1}(\cdot | s')}} [A^{\pi_{\theta_2}}(s', \mathbf{a})] \\ &= \mathbb{E}_{s' \sim d_{\theta_1}^s} \left[\mathbb{E}_{\mathbf{a} \sim \pi_{\theta_1}(\cdot | s')} [A^{\pi_{\theta_2}}(s', \mathbf{a})] - \mathbb{E}_{\mathbf{a} \sim \pi_{\theta_2}(\cdot | s')} [A^{\pi_{\theta_2}}(s', \mathbf{a})] \right] \\ &= \mathbb{E}_{s' \sim d_{\theta_1}^s} \left[\int (\pi_{\theta_1}(\mathbf{a} | s') - \pi_{\theta_2}(\mathbf{a} | s')) A^{\pi_{\theta_2}}(s', \mathbf{a}) d\mathbf{a} \right], \end{aligned}$$

where we applied Lemma B.1 (see also [Lemma 1.16, 2]) and then we exploited the fact that, for every $\theta \in \Theta$ and $s \in \mathcal{S}$, it holds that $\mathbb{E}_{\mathbf{a} \sim \pi_\theta(\cdot | s)} [A^{\pi_\theta}(s, \mathbf{a})] = 0$ (as similarly done in [Lemma 2, 50]). Thus, by considering the absolute value of the interested quantity, we have the following:

$$\begin{aligned} |V^{\pi_{\theta_1}}(s) - V^{\pi_{\theta_2}}(s)| &\leq \mathbb{E}_{s' \sim d_{\theta_1}^s} \left[\int |\pi_{\theta_1}(\mathbf{a} | s') - \pi_{\theta_2}(\mathbf{a} | s')| |A^{\pi_{\theta_2}}(s', \mathbf{a})| d\mathbf{a} \right] \\ &\leq \frac{2R_{\max}}{1-\gamma} \mathbb{E}_{s' \sim d_{\theta_1}^s} \left[\int |\pi_{\theta_1}(\mathbf{a} | s') - \pi_{\theta_2}(\mathbf{a} | s')| d\mathbf{a} \right] \\ &= \frac{4R_{\max}}{1-\gamma} \mathbb{E}_{s' \sim d_{\theta_1}^s} [D_{\text{TV}}(\pi_{\theta_1}(\cdot | s'), \pi_{\theta_2}(\cdot | s'))] \\ &\leq \frac{4R_{\max}}{1-\gamma} \sup_{s \in \mathcal{S}} D_{\text{TV}}(\pi_{\theta_1}(\cdot | s), \pi_{\theta_2}(\cdot | s)), \end{aligned}$$

for which we just employed the Jensen's inequality, the fact that, for every $\theta \in \Theta$, $s \in \mathcal{S}$, and $\mathbf{a} \in \mathcal{A}$, it holds that $|A^{\pi_\theta}(s, \mathbf{a})| \leq \frac{2R_{\max}}{1-\gamma}$, and finally the TV definition. We can finally bound the $\mathbf{B}$ quantity, by exploiting this last obtained result. Specifically, we have:

$$\begin{aligned} \mathbf{B} &= \mathbb{E}_{s \sim d_{\theta_2}} [|V^{\pi_{\theta_1}}(s) - V^{\pi_{\theta_2}}(s)|] \\ &\leq \frac{4R_{\max}}{1-\gamma} \sup_{s \in \mathcal{S}} D_{\text{TV}}(\pi_{\theta_1}(\cdot | s), \pi_{\theta_2}(\cdot | s)). \end{aligned}$$

Bounding the Term A. To bound the term $\mathbf{A}$, we will exploit what done for the term $\mathbf{B}$, by exploiting the following:

$$Q^{\pi_{\theta_1}}(s, \mathbf{a}) - Q^{\pi_{\theta_2}}(s, \mathbf{a}) = \gamma \int p(\mathbf{x} | s, \mathbf{a}) (V^{\pi_{\theta_1}}(\mathbf{x}) - V^{\pi_{\theta_2}}(\mathbf{x})) d\mathbf{x},$$

just by exploiting the action-value function definition. Thus, we have the following:

$$\begin{aligned} \mathbf{A} &= \mathbb{E}_{\substack{s \sim d_{\theta_2} \\ \mathbf{a} \sim \pi_\theta(\cdot | s)}} [|Q^{\pi_{\theta_1}}(s, \mathbf{a}) - Q^{\pi_{\theta_2}}(s, \mathbf{a})|] \\ &= \gamma \mathbb{E}_{\substack{s \sim d_{\theta_2} \\ \mathbf{a} \sim \pi_\theta(\cdot | s)}} \left[\mathbb{E}_{\mathbf{x} \sim p(\cdot | s, \mathbf{a})} [|V^{\pi_{\theta_1}}(\mathbf{x}) - V^{\pi_{\theta_2}}(\mathbf{x})|] \right] \\ &\leq \frac{4\gamma R_{\max}}{1-\gamma} \sup_{s \in \mathcal{S}} D_{\text{TV}}(\pi_{\theta_1}(\cdot | s), \pi_{\theta_2}(\cdot | s)). \end{aligned}$$

Final Result. Putting all together, we can conclude that:

$$\left| \mathbb{E}_{\substack{s \sim d_{\theta_2} \\ \mathbf{a} \sim \pi_\theta(\cdot | s)}} [A^{\pi_{\theta_1}}(s, \mathbf{a}) - A^{\pi_{\theta_2}}(s, \mathbf{a})] \right| \leq \frac{4R_{\max}(1+\gamma)}{1-\gamma} \sup_{s \in \mathcal{S}} D_{\text{TV}}(\pi_{\theta_1}(\cdot | s), \pi_{\theta_2}(\cdot | s)).$$

□

Theorem B.4. Consider to have access to $\omega \in \mathbb{N}_{\geq 1}$ policies $\Pi_\omega = \{\pi_{\theta_i}\}_{i=1}^\omega$ and to a distribution ν over the policies into Π_ω . Then, for every $\theta, \bar{\theta} \in \Theta$, the following holds:

$$\begin{aligned} J(\theta) - J(\bar{\theta}) &\geq \mathbb{E}_{i \sim \nu} \left[\mathbb{E}_{\substack{s \sim d_{\theta_i} \\ a \sim \pi_{\theta_i}(\cdot | s)}} \left[\frac{\pi_\theta(a | s)}{\pi_{\theta_i}(a | s)} A^{\pi_{\theta_i}}(s, a) \right] \right] \\ &\quad - \frac{2\gamma C(\theta, \bar{\theta})}{(1-\gamma)^2} \mathbb{E}_{i \sim \nu} \left[\mathbb{E}_{s \sim d_{\theta_i}} [D_{\text{TV}}(\pi_\theta(\cdot | s), \pi_{\theta_i}(\cdot | s))] \right] \\ &\quad - \frac{4R_{\max}(1+\gamma)}{1-\gamma} \mathbb{E}_{i \sim \nu} \left[\sup_{s \in \mathcal{S}} D_{\text{TV}}(\pi_{\bar{\theta}}(\cdot | s), \pi_{\theta_i}(\cdot | s)) \right]. \end{aligned}$$

Proof. We start the proof from Lemma B.1 [23]. For ever $\theta, \bar{\theta} \in \Theta$ the following holds:

$$\begin{aligned} J(\theta) - J(\bar{\theta}) &= \mathbb{E}_{\substack{s \sim d_\theta \\ a \sim \pi_\theta(\cdot | s)}} [A^{\pi_\theta}(s, a)] \\ &= \mathbb{E}_{\substack{s \sim d_{\bar{\theta}} \\ a \sim \pi_{\bar{\theta}}(\cdot | s)}} \left[\frac{d_\theta(s) \pi_\theta(a | s)}{d_{\bar{\theta}}(s) \pi_{\bar{\theta}}(a | s)} A^{\pi_\theta}(s, a) \right]. \end{aligned}$$

Now, by following the same procedure as Lemma B.2 [44, 1] and [Lemma 7, 46], for every $\pi_{\theta_i} \in \Pi_\omega$ we have the following:

$$J(\theta) - J(\bar{\theta}) \geq \underbrace{\mathbb{E}_{\substack{s \sim d_{\theta_i} \\ a \sim \pi_{\theta_i}(\cdot | s)}} \left[\frac{\pi_\theta(a | s)}{\pi_{\theta_i}(a | s)} A^{\pi_{\bar{\theta}}}(s, a) \right]}_{=:A} - \frac{2\gamma C(\theta, \bar{\theta})}{(1-\gamma)^2} \mathbb{E}_{s \sim d_{\theta_i}} [D_{\text{TV}}(\pi_\theta(\cdot | s), \pi_{\theta_i}(\cdot | s))],$$

where $C(\theta, \bar{\theta}) := \sup_{s \in \mathcal{S}} |\mathbb{E}_{a \sim \pi_\theta(\cdot | s)} [A^{\pi_{\bar{\theta}}}(s, a)]|$ is the same as in Lemma B.2, depending just on the policies appearing on the left-hand side of the policy improvement lower bound. Stopping here, we would have the policy improvement bound of [46, 33, 69, 31] which would require to have access to advantages relative to policy $\pi_{\bar{\theta}}$ in every sampled state-action pairs from policies in Π_ω . To fix this, we would leverage Lemma B.3.³

To continue the proof, let us sum and subtract $A^{\pi_{\theta_i}}(s, a)$ into the argument of the expectation of term **A**:

$$\begin{aligned} \mathbf{A} &= \mathbb{E}_{\substack{s \sim d_{\theta_i} \\ a \sim \pi_{\theta_i}(\cdot | s)}} \left[\frac{\pi_\theta(a | s)}{\pi_{\theta_i}(a | s)} A^{\pi_{\bar{\theta}}}(s, a) \right] \\ &= \mathbb{E}_{\substack{s \sim d_{\theta_i} \\ a \sim \pi_{\theta_i}(\cdot | s)}} \left[\frac{\pi_\theta(a | s)}{\pi_{\theta_i}(a | s)} A^{\pi_{\theta_i}}(s, a) \right] + \underbrace{\mathbb{E}_{\substack{s \sim d_{\theta_i} \\ a \sim \pi_{\theta_i}(\cdot | s)}} \left[\frac{\pi_\theta(a | s)}{\pi_{\theta_i}(a | s)} (A^{\pi_{\bar{\theta}}}(s, a) - A^{\pi_{\theta_i}}(s, a)) \right]}_{=:B}. \end{aligned}$$

Now, we just need to bound term **B**, that can be easily done thanks to Lemma B.3. Specifically, we have the following:

$$\begin{aligned} \mathbf{B} &= \mathbb{E}_{\substack{s \sim d_{\theta_i} \\ a \sim \pi_{\theta_i}(\cdot | s)}} \left[\frac{\pi_\theta(a | s)}{\pi_{\theta_i}(a | s)} (A^{\pi_{\bar{\theta}}}(s, a) - A^{\pi_{\theta_i}}(s, a)) \right] \\ &= \mathbb{E}_{\substack{s \sim d_{\theta_i} \\ a \sim \pi_\theta(\cdot | s)}} [A^{\pi_{\bar{\theta}}}(s, a) - A^{\pi_{\theta_i}}(s, a)]. \end{aligned}$$

³A similar need is faced in [Lemma 3, 31] to change the appearing advantage, however, it is not the one related to the data-sampling policy π_i .

By noting that $\mathbf{B} \geq -|\mathbf{B}|$ and then applying Lemma B.3, we can conclude that:

$$\begin{aligned} J(\boldsymbol{\theta}) - J(\bar{\boldsymbol{\theta}}) &\geq \mathbb{E}_{\substack{\mathbf{s} \sim d_{\boldsymbol{\theta}_i} \\ \mathbf{a} \sim \pi_{\boldsymbol{\theta}_i}(\cdot | \mathbf{s})}} \left[\frac{\pi_{\boldsymbol{\theta}}(\mathbf{a} | \mathbf{s})}{\pi_{\boldsymbol{\theta}_i}(\mathbf{a} | \mathbf{s})} A^{\pi_{\boldsymbol{\theta}_i}}(\mathbf{s}, \mathbf{a}) \right] - \frac{2\gamma C(\boldsymbol{\theta}, \bar{\boldsymbol{\theta}})}{(1-\gamma)^2} \mathbb{E}_{\mathbf{s} \sim d_{\boldsymbol{\theta}_i}} [D_{\text{TV}}(\pi_{\boldsymbol{\theta}}(\cdot | \mathbf{s}), \pi_{\boldsymbol{\theta}_i}(\cdot | \mathbf{s}))] \\ &\quad - \frac{4R_{\max}(1+\gamma)}{1-\gamma} \sup_{\mathbf{s} \in \mathcal{S}} D_{\text{TV}}(\pi_{\bar{\boldsymbol{\theta}}}(\cdot | \mathbf{s}), \pi_{\boldsymbol{\theta}_i}(\cdot | \mathbf{s})). \end{aligned}$$

Finally, as done in [Theorem 1, 46], since the previous bound is holding for every $\pi_{\boldsymbol{\theta}_i} \in \Pi_{\omega}$, then for any choice of distribution ν over the policies in Π_{ω} we have the following:

$$\begin{aligned} J(\boldsymbol{\theta}) - J(\bar{\boldsymbol{\theta}}) &\geq \mathbb{E}_{i \sim \nu} \left[\mathbb{E}_{\substack{\mathbf{s} \sim d_{\boldsymbol{\theta}_i} \\ \mathbf{a} \sim \pi_{\boldsymbol{\theta}_i}(\cdot | \mathbf{s})}} \left[\frac{\pi_{\boldsymbol{\theta}}(\mathbf{a} | \mathbf{s})}{\pi_{\boldsymbol{\theta}_i}(\mathbf{a} | \mathbf{s})} A^{\pi_{\boldsymbol{\theta}_i}}(\mathbf{s}, \mathbf{a}) \right] \right] \\ &\quad - \frac{2\gamma C(\boldsymbol{\theta}, \bar{\boldsymbol{\theta}})}{(1-\gamma)^2} \mathbb{E}_{i \sim \nu} \left[\mathbb{E}_{\mathbf{s} \sim d_{\boldsymbol{\theta}_i}} [D_{\text{TV}}(\pi_{\boldsymbol{\theta}}(\cdot | \mathbf{s}), \pi_{\boldsymbol{\theta}_i}(\cdot | \mathbf{s}))] \right] \\ &\quad - \frac{4R_{\max}(1+\gamma)}{1-\gamma} \mathbb{E}_{i \sim \nu} \left[\sup_{\mathbf{s} \in \mathcal{S}} D_{\text{TV}}(\pi_{\bar{\boldsymbol{\theta}}}(\cdot | \mathbf{s}), \pi_{\boldsymbol{\theta}_i}(\cdot | \mathbf{s})) \right]. \end{aligned}$$

□

Theorem 3.1. Let $\omega_k \in \mathbb{N}_{\geq 1}$. Having access to $\{\pi_{\boldsymbol{\theta}_{k-i}}\}_{i=0}^{\omega_k-1}$, for any target $\pi_{\boldsymbol{\theta}}$, the following holds:

$$\begin{aligned} J(\boldsymbol{\theta}) - J(\boldsymbol{\theta}_k) &\geq \frac{1}{\omega_k} \sum_{i=0}^{\omega_k-1} \mathbb{E}_{\substack{\mathbf{s} \sim d_{\boldsymbol{\theta}_{k-i}} \\ \mathbf{a} \sim \pi_{\boldsymbol{\theta}_{k-i}}(\cdot | \mathbf{s})}} \left[\frac{\pi_{\boldsymbol{\theta}}(\mathbf{a} | \mathbf{s})}{\pi_{\boldsymbol{\theta}_{k-i}}(\mathbf{a} | \mathbf{s})} A^{\pi_{\boldsymbol{\theta}_{k-i}}}(\mathbf{s}, \mathbf{a}) \right] \\ &\quad - \frac{2\gamma C(\boldsymbol{\theta}, \boldsymbol{\theta}_k)}{(1-\gamma)^2 \omega_k} \sum_{i=0}^{\omega_k-1} \mathbb{E}_{\mathbf{s} \sim d_{\boldsymbol{\theta}_{k-i}}} [D_{\text{TV}}(\pi_{\boldsymbol{\theta}}(\cdot | \mathbf{s}), \pi_{\boldsymbol{\theta}_{k-i}}(\cdot | \mathbf{s}))] \\ &\quad - \frac{4R_{\max}(1+\gamma)}{(1-\gamma)\omega_k} \sum_{i=0}^{\omega_k-1} \sup_{\mathbf{s} \in \mathcal{S}} D_{\text{TV}}(\pi_{\boldsymbol{\theta}_k}(\cdot | \mathbf{s}), \pi_{\boldsymbol{\theta}_{k-i}}(\cdot | \mathbf{s})). \end{aligned}$$

Proof. The proof straightforwardly follows from Theorem B.4. In particular, setting $\Pi_{\omega} = \{\pi_{\boldsymbol{\theta}_{k-i}}\}_{i=0}^{\omega_k-1}$, $\bar{\boldsymbol{\theta}} = \boldsymbol{\theta}_k$, and by considering a uniform distribution ν over the policies in the window. Thus, we have the following:

$$\begin{aligned} J(\boldsymbol{\theta}) - J(\boldsymbol{\theta}_k) &\geq \frac{1}{\omega_k} \sum_{i=0}^{\omega_k-1} \mathbb{E}_{\substack{\mathbf{s} \sim d_{\boldsymbol{\theta}_{k-i}} \\ \mathbf{a} \sim \pi_{\boldsymbol{\theta}_{k-i}}(\cdot | \mathbf{s})}} \left[\frac{\pi_{\boldsymbol{\theta}}(\mathbf{a} | \mathbf{s})}{\pi_{\boldsymbol{\theta}_{k-i}}(\mathbf{a} | \mathbf{s})} A^{\pi_{\boldsymbol{\theta}_{k-i}}}(\mathbf{s}, \mathbf{a}) \right] \\ &\quad - \frac{2\gamma C(\boldsymbol{\theta}, \boldsymbol{\theta}_k)}{(1-\gamma)^2 \omega_k} \sum_{i=0}^{\omega_k-1} \mathbb{E}_{\mathbf{s} \sim d_{\boldsymbol{\theta}_{k-i}}} [D_{\text{TV}}(\pi_{\boldsymbol{\theta}}(\cdot | \mathbf{s}), \pi_{\boldsymbol{\theta}_{k-i}}(\cdot | \mathbf{s}))] \\ &\quad - \frac{4R_{\max}(1+\gamma)}{(1-\gamma)\omega_k} \sum_{i=0}^{\omega_k-1} \sup_{\mathbf{s} \in \mathcal{S}} D_{\text{TV}}(\pi_{\boldsymbol{\theta}_k}(\cdot | \mathbf{s}), \pi_{\boldsymbol{\theta}_{k-i}}(\cdot | \mathbf{s})). \end{aligned}$$

□

Theorem 3.2. Let $\omega_k \in \mathbb{N}_{\geq 1}$. Having access to $\{\pi_{\theta_{k-i}}\}_{i=0}^{\omega_k-1}$, let $d_{\omega_k}^{\text{BH}}(\mathbf{s}) := \frac{1}{\omega_k} \sum_{i=0}^{\omega_k-1} d_{\theta_{k-i}}(\mathbf{s})$ be the uniform mixture of the discounted state occupancies. Then, for any π_{θ} , the following holds:

$$\begin{aligned} J(\theta) - J(\theta_k) &\geq \\ &\frac{1}{\omega_k} \sum_{i=0}^{\omega_k-1} \mathbb{E}_{\substack{\mathbf{s} \sim d_{\theta_{k-i}} \\ \mathbf{a} \sim \pi_{\theta_{k-i}}(\cdot | \mathbf{s})}} \left[\frac{\pi_{\theta}(\mathbf{a} | \mathbf{s})}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} | \mathbf{s})} A^{\pi_{\theta_{k-i}}}(\mathbf{s}, \mathbf{a}) \right] - \frac{2\gamma C(\theta, \theta_k)}{(1-\gamma)^2 \omega_k} \sum_{i=0}^{\omega_k-1} \mathbb{E}_{\mathbf{s} \sim d_{\theta_{k-i}}} [D_{\text{TV}}(\pi_{\theta}(\cdot | \mathbf{s}), \pi_{\omega_k}^{\text{BH}}(\cdot | \mathbf{s}))] \\ &- \frac{4R_{\max}(1+\gamma)}{1-\gamma} \sum_{i=0}^{\omega_k-1} \sup_{\mathbf{s} \in \mathcal{S}} D_{\text{TV}}(\pi_{\theta_k}(\cdot | \mathbf{s}), \pi_{\theta_{k-i}}(\cdot | \mathbf{s})) - \left(\frac{4R_{\max}}{(1-\gamma)} + \frac{2\gamma C(\theta, \theta_k)}{(1-\gamma)^2 \omega_k} \right) \sum_{i=0}^{\omega_k-1} D_{\text{TV}}(d_{\omega_k}^{\text{BH}}, d_{\theta_{k-i}}). \end{aligned}$$

Proof. To start the proof, let us start from Lemma B.1:

$$\begin{aligned} J(\theta) - J(\theta_k) &= \mathbb{E}_{\substack{\mathbf{s} \sim d_{\theta} \\ \mathbf{a} \sim \pi_{\theta}(\cdot | \mathbf{s})}} [A^{\pi_{\theta_k}}(\mathbf{s}, \mathbf{a})] \pm \mathbb{E}_{\substack{\mathbf{s} \sim d_{\omega_k}^{\text{BH}} \\ \mathbf{a} \sim \pi_{\theta}(\cdot | \mathbf{s})}} [A^{\pi_{\theta_k}}(\mathbf{s}, \mathbf{a})] \\ &= \mathbb{E}_{\substack{\mathbf{s} \sim d_{\omega_k}^{\text{BH}} \\ \mathbf{a} \sim \pi_{\omega_k}^{\text{BH}}(\cdot | \mathbf{s})}} \left[\frac{\pi_{\theta}(\mathbf{a} | \mathbf{s})}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} | \mathbf{s})} A^{\pi_{\theta_k}}(\mathbf{s}, \mathbf{a}) \right] - \int (d_{\omega_k}^{\text{BH}}(\mathbf{s}) - d_{\theta}(\mathbf{s})) \mathbb{E}_{\mathbf{a} \sim \pi_{\theta}(\cdot | \mathbf{s})} [A^{\pi_{\theta_k}}(\mathbf{s}, \mathbf{a})] d\mathbf{s}, \end{aligned}$$

where we just changed, applying the needed importance weight correction, the state and action sampling distributions.

Now, let us introduce the BH state-action occupancy. Specifically, for every $\mathbf{s} \in \mathcal{S}$ and $\mathbf{a} \in \mathcal{A}$, let $\mu_{\omega_k}^{\text{BH}}(\mathbf{s}, \mathbf{a}) := \frac{1}{\omega_k} \sum_{i=0}^{\omega_k-1} d_{\theta_{k-i}}(\mathbf{s}) \pi_{\theta_{k-i}}(\mathbf{a} | \mathbf{s})$. Notice that $d_{\omega_k}^{\text{BH}}$ is the discounted state occupancy induced by $\mu_{\omega_k}^{\text{BH}}$ and that:

$$\int \mu_{\omega_k}^{\text{BH}}(\mathbf{s}, \mathbf{a}) d\mathbf{a} = \frac{1}{\omega_k} \sum_{i=0}^{\omega_k-1} \int d_{\theta_{k-i}}(\mathbf{s}) \pi_{\theta_{k-i}}(\mathbf{a} | \mathbf{s}) d\mathbf{a} = d_{\omega_k}^{\text{BH}}(\mathbf{s}).$$

Letting $\mu_{\omega_k}^{\text{BH}}(\mathbf{a} | \mathbf{s}) := \mu_{\omega_k}^{\text{BH}}(\mathbf{s}, \mathbf{a}) / d_{\omega_k}^{\text{BH}}(\mathbf{s})$ we note that, in general, we have the following:

$$\mu_{\omega_k}^{\text{BH}}(\mathbf{a} | \mathbf{s}) = \frac{\sum_{i=0}^{\omega_k-1} d_{\theta_{k-i}}(\mathbf{s}) \pi_{\theta_{k-i}}(\mathbf{a} | \mathbf{s})}{\sum_{i=0}^{\omega_k-1} d_{\theta_{k-i}}(\mathbf{s})} \neq \frac{1}{\omega_k} \sum_{i=0}^{\omega_k-1} \pi_{\theta_{k-i}}(\mathbf{a} | \mathbf{s}) = \pi_{\omega_k}^{\text{BH}}(\mathbf{a} | \mathbf{s}).$$

That being said, by applying the argument of [1], the following holds:

$$\begin{aligned} J(\theta) - J(\theta_k) &= \mathbb{E}_{\substack{\mathbf{s} \sim d_{\omega_k}^{\text{BH}} \\ \mathbf{a} \sim \pi_{\omega_k}^{\text{BH}}(\cdot | \mathbf{s})}} \left[\frac{\pi_{\theta}(\mathbf{a} | \mathbf{s})}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} | \mathbf{s})} A^{\pi_{\theta_k}}(\mathbf{s}, \mathbf{a}) \right] - \int (d_{\omega_k}^{\text{BH}}(\mathbf{s}) - d_{\theta}(\mathbf{s})) \mathbb{E}_{\mathbf{a} \sim \pi_{\theta}(\cdot | \mathbf{s})} [A^{\pi_{\theta_k}}(\mathbf{s}, \mathbf{a})] d\mathbf{s} \\ &\geq \mathbb{E}_{\substack{\mathbf{s} \sim d_{\omega_k}^{\text{BH}} \\ \mathbf{a} \sim \pi_{\omega_k}^{\text{BH}}(\cdot | \mathbf{s})}} \left[\frac{\pi_{\theta}(\mathbf{a} | \mathbf{s})}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} | \mathbf{s})} A^{\pi_{\theta_k}}(\mathbf{s}, \mathbf{a}) \right] - \frac{2\gamma C(\theta, \theta_k)}{(1-\gamma)^2} \mathbb{E}_{\mathbf{s} \sim d_{\omega_k}^{\text{BH}}} [D_{\text{TV}}(\pi_{\theta}(\cdot | \mathbf{s}), \mu_{\omega_k}^{\text{BH}}(\cdot | \mathbf{s}))] \\ &= \mathbb{E}_{\substack{\mathbf{s} \sim d_{\omega_k}^{\text{BH}} \\ \mathbf{a} \sim \pi_{\omega_k}^{\text{BH}}(\cdot | \mathbf{s})}} \left[\frac{\pi_{\theta}(\mathbf{a} | \mathbf{s})}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} | \mathbf{s})} A^{\pi_{\theta_k}}(\mathbf{s}, \mathbf{a}) \right] - \frac{2\gamma C(\theta, \theta_k)}{(1-\gamma)^2 \omega_k} \sum_{i=0}^{\omega_k-1} \mathbb{E}_{\mathbf{s} \sim d_{\theta_{k-i}}} [D_{\text{TV}}(\pi_{\theta}(\cdot | \mathbf{s}), \mu_{\omega_k}^{\text{BH}}(\cdot | \mathbf{s}))]. \end{aligned}$$

Now, by triangular inequality we have that:

$$D_{\text{TV}}(\pi_{\theta}(\cdot | \mathbf{s}), \mu_{\omega_k}^{\text{BH}}(\cdot | \mathbf{s})) \leq D_{\text{TV}}(\pi_{\theta}(\cdot | \mathbf{s}), \pi_{\omega_k}^{\text{BH}}(\cdot | \mathbf{s})) + D_{\text{TV}}(\pi_{\omega_k}^{\text{BH}}(\cdot | \mathbf{s}), \mu_{\omega_k}^{\text{BH}}(\cdot | \mathbf{s})).$$

Furthermore, note that:

$$\begin{aligned}
|\pi_{\omega_k}^{\text{BH}}(\mathbf{a} \mid \mathbf{s}) - \mu_{\omega_k}^{\text{BH}}(\mathbf{a} \mid \mathbf{s})| &= \frac{1}{\omega_k} \left| \sum_{i=0}^{\omega_k-1} \pi_{\theta_{k-i}}(\mathbf{a} \mid \mathbf{s}) - \frac{d_{\theta_{k-i}}(\mathbf{s}) \pi_{\theta_{k-i}}(\mathbf{a} \mid \mathbf{s})}{d_{\omega_k}^{\text{BH}}(\mathbf{s})} \right| \\
&\leq \frac{1}{\omega_k} \sum_{i=0}^{\omega_k-1} \left| 1 - \frac{d_{\theta_{k-i}}(\mathbf{s})}{d_{\omega_k}^{\text{BH}}(\mathbf{s})} \right| \pi_{\theta_{k-i}}(\mathbf{a} \mid \mathbf{s}) \\
&\leq \frac{1}{\omega_k} \sum_{i=0}^{\omega_k-1} \frac{|d_{\omega_k}^{\text{BH}}(\mathbf{s}) - d_{\theta_{k-i}}(\mathbf{s})|}{d_{\omega_k}^{\text{BH}}(\mathbf{s})} \pi_{\theta_{k-i}}(\mathbf{a} \mid \mathbf{s}), \tag{1}
\end{aligned}$$

which simply follows from the triangular inequality and from algebraic manipulations taking into account the $d_{\omega_k}^{\text{BH}}$ definition. Equipped with these two arguments, we have that:

$$\begin{aligned}
&\mathbb{E}_{\mathbf{s} \sim d_{\omega_k}^{\text{BH}}} [D_{\text{TV}}(\pi_{\omega_k}^{\text{BH}}(\cdot \mid \mathbf{s}), \mu_{\omega_k}^{\text{BH}}(\cdot \mid \mathbf{s}))] \\
&\leq \frac{1}{2\omega_k} \sum_{i=0}^{\omega_k-1} \int_{\mathcal{S}} d_{\omega_k}^{\text{BH}}(\mathbf{s}) \frac{|d_{\omega_k}^{\text{BH}}(\mathbf{s}) - d_{\theta_{k-i}}(\mathbf{s})|}{d_{\omega_k}^{\text{BH}}(\mathbf{s})} \int_{\mathcal{A}} \pi_{\theta_{k-i}}(\mathbf{a} \mid \mathbf{s}) d\mathbf{a} d\mathbf{s} \\
&= \frac{1}{\omega_k} \sum_{i=0}^{\omega_k-1} D_{\text{TV}}(d_{\omega_k}^{\text{BH}}, d_{\theta_{k-i}}),
\end{aligned}$$

thus leading to the following intermediate form of the performance improvement lower bound we aim to reach:

$$\begin{aligned}
J(\boldsymbol{\theta}) - J(\boldsymbol{\theta}_k) &\geq \mathbb{E}_{\substack{\mathbf{s} \sim d_{\omega_k}^{\text{BH}} \\ \mathbf{a} \sim \pi_{\omega_k}^{\text{BH}}(\cdot \mid \mathbf{s})}} \left[\frac{\pi_{\boldsymbol{\theta}}(\mathbf{a} \mid \mathbf{s})}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} \mid \mathbf{s})} A^{\pi_{\boldsymbol{\theta}_k}}(\mathbf{s}, \mathbf{a}) \right] \\
&\quad - \frac{2\gamma C(\boldsymbol{\theta}, \boldsymbol{\theta}_k)}{(1-\gamma)^2 \omega_k} \sum_{i=0}^{\omega_k-1} \mathbb{E}_{\mathbf{s} \sim d_{\theta_{k-i}}} [D_{\text{TV}}(\pi_{\boldsymbol{\theta}}(\cdot \mid \mathbf{s}), \pi_{\omega_k}^{\text{BH}}(\cdot \mid \mathbf{s}))] \\
&\quad - \frac{2\gamma C(\boldsymbol{\theta}, \boldsymbol{\theta}_k)}{(1-\gamma)^2 \omega_k} \sum_{i=0}^{\omega_k-1} D_{\text{TV}}(d_{\omega_k}^{\text{BH}}, d_{\theta_{k-i}}). \tag{2}
\end{aligned}$$

In order to carry on with the derivation, we introduce the following quantity:

$$\begin{aligned}
\mathbf{A} &:= \mathbb{E}_{(\mathbf{s}, \mathbf{a}) \sim \mu_{\omega_k}^{\text{BH}}} \left[\frac{\pi_{\boldsymbol{\theta}}(\mathbf{a} \mid \mathbf{s})}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} \mid \mathbf{s})} A^{\pi_{\boldsymbol{\theta}_k}}(\mathbf{s}, \mathbf{a}) \right] \\
&= \frac{1}{\omega_k} \sum_{i=0}^{\omega_k-1} \iint d_{\theta_{k-i}}(\mathbf{s}) \pi_{\theta_{k-i}}(\mathbf{a} \mid \mathbf{s}) \frac{\pi_{\boldsymbol{\theta}}(\mathbf{a} \mid \mathbf{s})}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} \mid \mathbf{s})} A^{\pi_{\boldsymbol{\theta}_k}}(\mathbf{s}, \mathbf{a}) d\mathbf{s} d\mathbf{a} \\
&= \frac{1}{\omega_k} \sum_{i=0}^{\omega_k-1} \mathbb{E}_{\substack{\mathbf{s} \sim d_{\theta_{k-i}} \\ \mathbf{a} \sim \pi_{\theta_{k-i}}(\cdot \mid \mathbf{s})}} \left[\frac{\pi_{\boldsymbol{\theta}}(\mathbf{a} \mid \mathbf{s})}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} \mid \mathbf{s})} A^{\pi_{\boldsymbol{\theta}_k}}(\mathbf{s}, \mathbf{a}) \right],
\end{aligned}$$

where we just exploited the $\mu_{\omega_k}^{\text{BH}}$ definition.

Notice that the next steps are to make $\mathbf{A}$ appear into Equation (2), then employing Lemma B.3 to finally have the stale advantages. That being said, we proceed by adding and subtracting $\mathbf{A}$ into

Equation (2), obtaining the following:

$$\begin{aligned}
J(\theta) - J(\theta_k) &\geq \frac{1}{\omega_k} \sum_{i=0}^{\omega_k-1} \mathbb{E}_{\substack{s \sim d_{\theta_{k-i}} \\ \mathbf{a} \sim \pi_{\theta_{k-i}}(\cdot | s)}} \left[\frac{\pi_{\theta}(\mathbf{a} | s)}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} | s)} A^{\pi_{\theta_k}}(s, \mathbf{a}) \right] \\
&\quad - \frac{2\gamma C(\theta, \theta_k)}{(1-\gamma)^2 \omega_k} \sum_{i=0}^{\omega_k-1} \mathbb{E}_{s \sim d_{\theta_{k-i}}} [D_{\text{TV}}(\pi_{\theta}(\cdot | s), \pi_{\omega_k}^{\text{BH}}(\cdot | s))] \\
&\quad - \frac{2\gamma C(\theta, \theta_k)}{(1-\gamma)^2 \omega_k} \sum_{i=0}^{\omega_k-1} D_{\text{TV}}(d_{\omega_k}^{\text{BH}}, d_{\theta_{k-i}}) \\
&\quad + \underbrace{\mathbb{E}_{\substack{s \sim d_{\omega_k}^{\text{BH}} \\ \mathbf{a} \sim \pi_{\omega_k}^{\text{BH}}(\cdot | s)}} \left[\frac{\pi_{\theta}(\mathbf{a} | s)}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} | s)} A^{\pi_{\theta_k}}(s, \mathbf{a}) \right] - \mathbb{E}_{(s, \mathbf{a}) \sim \mu_{\omega_k}^{\text{BH}}} \left[\frac{\pi_{\theta}(\mathbf{a} | s)}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} | s)} A^{\pi_{\theta_k}}(s, \mathbf{a}) \right]}_{=: \mathbf{B}}.
\end{aligned} \tag{3}$$

Next, we have to treat $\mathbf{B}$. To do so, we can exploit the fact that $\int \mu_{\omega_k}^{\text{BH}}(s, \mathbf{a}) d\mathbf{a} = d_{\omega_k}^{\text{BH}}(s)$ for every $s \in \mathcal{S}$. Then, the following holds:

$$\begin{aligned}
\mathbf{B} &= \mathbb{E}_{\substack{s \sim d_{\omega_k}^{\text{BH}} \\ \mathbf{a} \sim \pi_{\omega_k}^{\text{BH}}(\cdot | s)}} \left[\frac{\pi_{\theta}(\mathbf{a} | s)}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} | s)} A^{\pi_{\theta_k}}(s, \mathbf{a}) \right] - \mathbb{E}_{(s, \mathbf{a}) \sim \mu_{\omega_k}^{\text{BH}}} \left[\frac{\pi_{\theta}(\mathbf{a} | s)}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} | s)} A^{\pi_{\theta_k}}(s, \mathbf{a}) \right] \\
&= \mathbb{E}_{s \sim d_{\omega_k}^{\text{BH}}} \left[\mathbb{E}_{\mathbf{a} \sim \pi_{\omega_k}^{\text{BH}}(\cdot | s)} \left[\frac{\pi_{\theta}(\mathbf{a} | s)}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} | s)} A^{\pi_{\theta_k}}(s, \mathbf{a}) \right] - \mathbb{E}_{\mathbf{a} \sim \mu_{\omega_k}^{\text{BH}}(\cdot | s)} \left[\frac{\pi_{\theta}(\mathbf{a} | s)}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} | s)} A^{\pi_{\theta_k}}(s, \mathbf{a}) \right] \right] \\
&= \mathbb{E}_{s \sim d_{\omega_k}^{\text{BH}}} \left[\int (\pi_{\omega_k}^{\text{BH}}(\mathbf{a} | s) - \mu_{\omega_k}^{\text{BH}}(\mathbf{a} | s)) \frac{\pi_{\theta}(\mathbf{a} | s)}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} | s)} A^{\pi_{\theta_k}}(s, \mathbf{a}) d\mathbf{a} \right].
\end{aligned}$$

Because of the presence of the BH-corrected ratio $\frac{\pi_{\theta}(\mathbf{a} | s)}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} | s)}$, we cannot directly bound $|\mathbf{B}|$ with a standard TV argument. Thus, we first note that:

$$\begin{aligned}
\mathbf{B} &\leq \mathbb{E}_{s \sim d_{\omega_k}^{\text{BH}}} \left[\int |\pi_{\omega_k}^{\text{BH}}(\mathbf{a} | s) - \mu_{\omega_k}^{\text{BH}}(\mathbf{a} | s)| \frac{\pi_{\theta}(\mathbf{a} | s)}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} | s)} |A^{\pi_{\theta_k}}(s, \mathbf{a})| d\mathbf{a} \right] \\
&\leq \frac{2R_{\max}}{1-\gamma} \mathbb{E}_{\substack{s \sim d_{\omega_k}^{\text{BH}} \\ \mathbf{a} \sim \pi_{\theta}(\cdot | s)}} \left[\frac{|\pi_{\omega_k}^{\text{BH}}(\mathbf{a} | s) - \mu_{\omega_k}^{\text{BH}}(\mathbf{a} | s)|}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} | s)} \right],
\end{aligned}$$

where we exploited the fact that for every $s \in \mathcal{S}$, $\mathbf{a} \in \mathcal{A}$, and $\theta \in \Theta$ it holds that $|A^{\pi_{\theta}}(s, \mathbf{a})| \leq \frac{2R_{\max}}{1-\gamma}$ and we used π_{θ} as an action-sampling distribution.

Exploiting Equation (1), we can finally conclude the $|\mathbf{B}|$ bound as:

$$\begin{aligned}
\mathbf{B} &\leq \frac{2R_{\max}}{1-\gamma} \mathbb{E}_{\substack{s \sim d_{\omega_k}^{\text{BH}} \\ \mathbf{a} \sim \pi_{\theta}(\cdot | s)}} \left[\frac{|\pi_{\omega_k}^{\text{BH}}(\mathbf{a} | s) - \mu_{\omega_k}^{\text{BH}}(\mathbf{a} | s)|}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} | s)} \right] \\
&\leq \frac{2R_{\max}}{(1-\gamma)\omega_k} \sum_{i=0}^{\omega_k-1} \mathbb{E}_{\substack{s \sim d_{\omega_k}^{\text{BH}} \\ \mathbf{a} \sim \pi_{\theta}(\cdot | s)}} \left[\frac{|d_{\omega_k}^{\text{BH}}(s) - d_{\theta_{k-i}}(s)|}{d_{\omega_k}^{\text{BH}}(s)} \frac{\pi_{\theta_{k-i}}(\mathbf{a} | s)}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} | s)} \right] \\
&\leq \frac{2R_{\max}}{(1-\gamma)} \sum_{i=0}^{\omega_k-1} \mathbb{E}_{s \sim d_{\omega_k}^{\text{BH}}} \left[\frac{|d_{\omega_k}^{\text{BH}}(s) - d_{\theta_{k-i}}(s)|}{d_{\omega_k}^{\text{BH}}(s)} \right] \\
&= \frac{4R_{\max}}{(1-\gamma)} \sum_{i=0}^{\omega_k-1} D_{\text{TV}}(d_{\omega_k}^{\text{BH}}, d_{\theta_{k-i}}),
\end{aligned}$$

where: (i) we applied the previous bound on $|\pi_{\omega_k}^{\text{BH}}(\mathbf{a} \mid \mathbf{s}) - \mu_{\omega_k}^{\text{BH}}(\mathbf{a} \mid \mathbf{s})|$ from Equation (1); (ii) we exploited the fact that $\frac{\pi_{\theta_{k-i}}(\mathbf{a} \mid \mathbf{s})}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} \mid \mathbf{s})} \leq \omega_k$ for every $\mathbf{s} \in \mathcal{S}$, $\mathbf{a} \in \mathcal{A}$, and $i \in \llbracket 0, \omega_k - 1 \rrbracket$; and (iii) we exploited the TV definition. Notice that, when exploiting the specific structure of the BH importance weight with $\pi_{\theta_{k-i}}$ at the numerator, we lose the dependence on π_{θ} in the inner expectation term.

We can now plug this result into Equation (3), obtaining:

$$\begin{aligned}
J(\boldsymbol{\theta}) - J(\boldsymbol{\theta}_k) &\geq \frac{1}{\omega_k} \sum_{i=0}^{\omega_k-1} \overbrace{\mathbb{E}_{\substack{\mathbf{s} \sim d_{\boldsymbol{\theta}_{k-i}} \\ \mathbf{a} \sim \pi_{\boldsymbol{\theta}_{k-i}}(\cdot \mid \mathbf{s})}} \left[\frac{\pi_{\boldsymbol{\theta}}(\mathbf{a} \mid \mathbf{s})}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} \mid \mathbf{s})} A^{\pi_{\boldsymbol{\theta}_k}}(\mathbf{s}, \mathbf{a}) \right]}^{=: \mathbf{C}} \\
&\quad - \frac{2\gamma C(\boldsymbol{\theta}, \boldsymbol{\theta}_k)}{(1-\gamma)^2 \omega_k} \sum_{i=0}^{\omega_k-1} \mathbb{E}_{\mathbf{s} \sim d_{\boldsymbol{\theta}_{k-i}}} [D_{\text{TV}}(\pi_{\boldsymbol{\theta}}(\cdot \mid \mathbf{s}), \pi_{\omega_k}^{\text{BH}}(\cdot \mid \mathbf{s}))] \\
&\quad - \left(\frac{4R_{\max}}{(1-\gamma)} + \frac{2\gamma C(\boldsymbol{\theta}, \boldsymbol{\theta}_k)}{(1-\gamma)^2 \omega_k} \right) \sum_{i=0}^{\omega_k-1} D_{\text{TV}}(d_{\omega_k}^{\text{BH}}, d_{\boldsymbol{\theta}_{k-i}}). \tag{4}
\end{aligned}$$

The last thing to do to conclude the proof is to substitute in $\mathbf{C}$ the fresh advantage $A^{\pi_{\boldsymbol{\theta}_k}}(\mathbf{s}, \mathbf{a})$ with the stale one $A^{\pi_{\boldsymbol{\theta}_{k-i}}}(\mathbf{s}, \mathbf{a})$, for every $i \in \llbracket 0, \omega_k - 1 \rrbracket$. To do so, we will proceed as done in Theorem 3.1, adding and subtracting the expectation term with the stale advantage, then resorting to Lemma B.3. Specifically, the following holds

$$\begin{aligned}
\mathbf{C} &= \mathbb{E}_{\substack{\mathbf{s} \sim d_{\boldsymbol{\theta}_{k-i}} \\ \mathbf{a} \sim \pi_{\boldsymbol{\theta}_{k-i}}(\cdot \mid \mathbf{s})}} \left[\frac{\pi_{\boldsymbol{\theta}}(\mathbf{a} \mid \mathbf{s})}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} \mid \mathbf{s})} A^{\pi_{\boldsymbol{\theta}_k}}(\mathbf{s}, \mathbf{a}) \right] \\
&= \mathbb{E}_{\substack{\mathbf{s} \sim d_{\boldsymbol{\theta}_{k-i}} \\ \mathbf{a} \sim \pi_{\boldsymbol{\theta}_{k-i}}(\cdot \mid \mathbf{s})}} \left[\frac{\pi_{\boldsymbol{\theta}}(\mathbf{a} \mid \mathbf{s})}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} \mid \mathbf{s})} A^{\pi_{\boldsymbol{\theta}_{k-i}}}(\mathbf{s}, \mathbf{a}) \right] + \underbrace{\mathbb{E}_{\substack{\mathbf{s} \sim d_{\boldsymbol{\theta}_{k-i}} \\ \mathbf{a} \sim \pi_{\boldsymbol{\theta}_{k-i}}(\cdot \mid \mathbf{s})}} \left[\frac{\pi_{\boldsymbol{\theta}}(\mathbf{a} \mid \mathbf{s})}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} \mid \mathbf{s})} (A^{\pi_{\boldsymbol{\theta}_k}}(\mathbf{s}, \mathbf{a}) - A^{\pi_{\boldsymbol{\theta}_{k-i}}}(\mathbf{s}, \mathbf{a})) \right]}_{=: \mathbf{D}}.
\end{aligned}$$

So now, it suffices to apply Lemma B.3 to $\mathbf{D}$, thus having:

$$\begin{aligned}
\mathbf{D} &= \mathbb{E}_{\substack{\mathbf{s} \sim d_{\boldsymbol{\theta}_{k-i}} \\ \mathbf{a} \sim \pi_{\boldsymbol{\theta}_{k-i}}(\cdot \mid \mathbf{s})}} \left[\frac{\pi_{\boldsymbol{\theta}}(\mathbf{a} \mid \mathbf{s})}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} \mid \mathbf{s})} (A^{\pi_{\boldsymbol{\theta}_k}}(\mathbf{s}, \mathbf{a}) - A^{\pi_{\boldsymbol{\theta}_{k-i}}}(\mathbf{s}, \mathbf{a})) \right] \\
&= \mathbb{E}_{\substack{\mathbf{s} \sim d_{\boldsymbol{\theta}_{k-i}} \\ \mathbf{a} \sim \pi_{\boldsymbol{\theta}}(\cdot \mid \mathbf{s})}} \left[\frac{\pi_{\boldsymbol{\theta}_{k-i}}(\mathbf{a} \mid \mathbf{s})}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} \mid \mathbf{s})} (A^{\pi_{\boldsymbol{\theta}_k}}(\mathbf{s}, \mathbf{a}) - A^{\pi_{\boldsymbol{\theta}_{k-i}}}(\mathbf{s}, \mathbf{a})) \right] \\
&\leq \omega_k \mathbb{E}_{\substack{\mathbf{s} \sim d_{\boldsymbol{\theta}_{k-i}} \\ \mathbf{a} \sim \pi_{\boldsymbol{\theta}}(\cdot \mid \mathbf{s})}} [|A^{\pi_{\boldsymbol{\theta}_k}}(\mathbf{s}, \mathbf{a}) - A^{\pi_{\boldsymbol{\theta}_{k-i}}}(\mathbf{s}, \mathbf{a})|] \\
&\leq \omega_k \frac{4R_{\max}(1+\gamma)}{1-\gamma} \sup_{\mathbf{s} \in \mathcal{S}} D_{\text{TV}}(\pi_{\boldsymbol{\theta}_k}(\cdot \mid \mathbf{s}), \pi_{\boldsymbol{\theta}_{k-i}}(\cdot \mid \mathbf{s})),
\end{aligned}$$

where (i) we changed the action sampling distribution from $\pi_{\boldsymbol{\theta}_{k-i}}$ to $\pi_{\boldsymbol{\theta}}$; (ii) we exploited $\frac{\pi_{\boldsymbol{\theta}_{k-i}}(\mathbf{a} \mid \mathbf{s})}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} \mid \mathbf{s})} \leq \omega_k$ for every $\mathbf{s} \in \mathcal{S}$, $\mathbf{a} \in \mathcal{A}$, and $i \in \llbracket 0, \omega_k - 1 \rrbracket$; (iii) we used the fact that $\mathbf{D} \geq -|\mathbf{D}|$, then applying the Jensen's inequality; and (iv) we employed Lemma B.3.

We can finally put all together, thus having the following final policy improvement bound:

$$\begin{aligned}
J(\theta) - J(\theta_k) &\geq \frac{1}{\omega_k} \sum_{i=0}^{\omega_k-1} \mathbb{E}_{\substack{\mathbf{s} \sim d_{\theta_{k-i}} \\ \mathbf{a} \sim \pi_{\theta_{k-i}}(\cdot | \mathbf{s})}} \left[\frac{\pi_{\theta}(\mathbf{a} | \mathbf{s})}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a} | \mathbf{s})} A^{\pi_{\theta_{k-i}}}(\mathbf{s}, \mathbf{a}) \right] \\
&\quad - \frac{2\gamma C(\theta, \theta_k)}{(1-\gamma)^2 \omega_k} \sum_{i=0}^{\omega_k-1} \mathbb{E}_{\mathbf{s} \sim d_{\theta_{k-i}}} [D_{\text{TV}}(\pi_{\theta}(\cdot | \mathbf{s}), \pi_{\omega_k}^{\text{BH}}(\cdot | \mathbf{s}))] \\
&\quad - \frac{4R_{\max}(1+\gamma)}{1-\gamma} \sum_{i=0}^{\omega_k-1} \sup_{\mathbf{s} \in \mathcal{S}} D_{\text{TV}}(\pi_{\theta_k}(\cdot | \mathbf{s}), \pi_{\theta_{k-i}}(\cdot | \mathbf{s})) \\
&\quad - \left(\frac{4R_{\max}}{(1-\gamma)} + \frac{2\gamma C(\theta, \theta_k)}{(1-\gamma)^2 \omega_k} \right) \sum_{i=0}^{\omega_k-1} D_{\text{TV}}(d_{\omega_k}^{\text{BH}}, d_{\theta_{k-i}}).
\end{aligned}$$

□

C Experimental Details

To help with reproducibility, we provide all the necessary implementation and experimental details we used for conducting the experiments presented in Section 4 and Appendix D.

Environments. We conducted our numerical evaluation on six continuous control environments leveraging the MuJoCo [57] control suite. Specifically, we considered: *Reacher-v5*, *Swimmer-v5*, *Hopper-v5*, *Walker2d-v5*, *HalfCheetah-v5*, and *Ant-v5*. Observations are normalized online using a running mean and variance estimate.

Implementation. For implementing $\omega\text{PPO-U}$ and $\omega\text{PPO-BH}$ we extended the PPO implementation of [48] with the following modifications. First, the rollout buffer is extended to hold the most recent ω_k rollouts (see Algorithm 1 describing the data collection phase): at each iteration k , the oldest rollout $\mathcal{D}_{k-\omega_k}$ is discarded and the new one $\mathcal{D}_k$ is appended. Second, mini-batches $\{\mathcal{B}_j\}_{j=1}^{N_{\mathcal{B}}}$ are sampled uniformly at random from the full buffer $\mathcal{D}(\omega_k)$, with the constraint that each $\mathcal{B}_j$ contains the same number of transitions for every window index (B/ω_k in the fixed B setting, B in the fixed $N_{\mathcal{B}}$ setting). Third, $\omega\text{PPO-BH}$ additionally stores the parameters of the $\omega_k - 1$ most recent past policies, which are used to evaluate the BH mixture denominator on every stored transition in $\mathcal{D}(\omega_k)$. Specifically, after every data collection phase, $\omega\text{PPO-BH}$ requires to evaluate all the samples in $\mathcal{D}(\omega_k)$ coming from past policies $\{\pi_{\theta_{k-i}}\}_{i=1}^{\omega_k-1}$ under π_{θ_k} , as well as evaluating newer samples in $\mathcal{D}_k$ under the past policies $\{\pi_{\theta_{k-i}}\}_{i=1}^{\omega_k-1}$. This requires to perform additional $2D(\omega_k - 1)$ evaluations, once per data collection. Finally, the ratio in the actor loss is replaced by the per-policy ratio for $\omega\text{PPO-U}$ or the BH-corrected ratio $\omega\text{PPO-BH}$, as defined in Section 3. All remaining implementation details are shared with standard PPO . The pseudo-code of the considered PPO sample reuse variants is presented in Algorithm 3.

Hyperparameters. For each environment, in order to conduct a fair comparison, we adopt the tuned PPO hyperparameters from [47] as our baseline configuration, with the exception of *Hopper-v5*, for which the provided configuration did not yield the expected performance and hyperparameters were selected manually. The full set of hyperparameters for each environment is reported in Table 2. For the networks’ activation functions we used $\tanh$ ones. $\omega\text{PPO-U}$ and $\omega\text{PPO-BH}$ share all common hyperparameters with PPO , with only the parameters specific to each comparison setting varied as described in Section 4 and D. This choice reflects our goal of providing a controlled comparison: by evaluating the reuse variants against a strongly tuned PPO baseline, any observed improvement can be attributed to the data reuse mechanism itself rather than to a suboptimal baseline.

Evaluation Protocol. Each method is evaluated over 10 independent runs using the same set of fixed random seeds across all compared methods. Specifically, we employed seeds in $\llbracket 0, 9 \rrbracket$. Policy performance is assessed every 10000 environment steps by running 50 complete episodes under the current deterministic policy, and the evaluation metric is the mean episodic return, computed as the undiscounted sum of rewards over a complete episode. For plotting purposes, the resulting mean return is smoothed via a moving average with a window of 10 evaluation points, and all plots report

Algorithm 3: ω PPO-U and ω PPO-BH.

Input : Initial parameters θ_0 and w_0 , iterations K , epochs per iteration $\tilde{K}$, batch size B , environments $\{E_e\}_{e=1}^{N_E}$, GAE parameter λ , interaction length H , discount factor γ , loss coefficients c_1 and c_2 , learning rate α , clipping parameter ϵ , window size ω .

1 Initialize optimization variables: $\theta \leftarrow \theta_0$, $w \leftarrow w_0$; initialize buffer $\mathcal{D}(\omega_0) \leftarrow \emptyset$.

2 **for** $k \in \llbracket 1, K \rrbracket$ **do**

 // Data Collection

3 $\mathcal{D}_k \leftarrow \text{Run Algorithm 1}$;

 // Buffer Update

4 **if** $\omega_k < \omega$ **then**

5 $\mathcal{D}(\omega_k) \leftarrow \mathcal{D}(\omega_{k-1}) \cup \mathcal{D}_k$;

6 **else**

7 $\mathcal{D}(\omega_k) \leftarrow \mathcal{D}(\omega_{k-1}) \setminus \mathcal{D}_{k-\omega} \cup \mathcal{D}_k$;

8 **end**

 // [ω PPO-BH only] Precompute mixture denominator

9 $\pi_{\omega_k}^{\text{BH}}(\mathbf{a} \mid \mathbf{s}) \leftarrow \frac{1}{\omega_k} \sum_{i=0}^{\omega_k-1} \pi_{\theta_{k-i}}(\mathbf{a} \mid \mathbf{s})$ for all $(\mathbf{s}, \mathbf{a}) \in \mathcal{D}(\omega_k)$;

 // Multi-Epoch Optimization

10 **for** $l \in \llbracket \tilde{K} \rrbracket$ **do**

11 Shuffle $\mathcal{D}(\omega_k)$;

12 Create $\tilde{\mathcal{D}}_k = \{\mathcal{B}_z\}_{z=1}^{N_{\mathcal{B}}}$, where $N_{\mathcal{B}} = \left\lceil \frac{|\mathcal{D}(\omega_k)|}{B} \right\rceil$, such that $\forall z \in \llbracket N_{\mathcal{B}} \rrbracket : |\mathcal{B}_z| = B$ equally represents all ω_k iterations;

13 **for** $\mathcal{B} \in \tilde{\mathcal{D}}_k$ **do**

 // Actor Loss

14 $\omega\text{PPO-U: } \rho_{i,j} \leftarrow \frac{\pi_{\theta}(\mathbf{a}_{i,j} \mid \mathbf{s}_{i,j})}{\pi_{\theta_{k-i}}(\mathbf{a}_{i,j} \mid \mathbf{s}_{i,j})}$;

15 $\omega\text{PPO-BH: } \rho_{i,j} \leftarrow \frac{\pi_{\theta}(\mathbf{a}_{i,j} \mid \mathbf{s}_{i,j})}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a}_{i,j} \mid \mathbf{s}_{i,j})}$;

16
$$\hat{L}^{\text{actor}}(\theta, \mathcal{B}) = \frac{1}{|\mathcal{B}|} \sum_{(i,j) \in \mathcal{B}} \min \{ \rho_{i,j} A_{i,j}, \text{clip}(\rho_{i,j}, 1 - \epsilon, 1 + \epsilon) A_{i,j} \}$$

 // Critic Loss (on $\mathcal{D}_k$ samples only)

17
$$\hat{L}^{\text{vf}}(w, \mathcal{B}) = \frac{1}{|\mathcal{B} \cap \mathcal{D}_k|} \sum_{(0,j) \in \mathcal{B}} \frac{1}{2} (V_w(s_{0,j}) - R_{0,j})^2$$

 // Update

18
$$\theta \leftarrow \theta + \alpha \nabla_{\theta} \left[\hat{L}^{\text{actor}}(\theta, \mathcal{B}) + c_2 \frac{1}{|\mathcal{B}|} \sum_{(i,j) \in \mathcal{B}} H(\pi_{\theta}(\cdot \mid \mathbf{s}_{i,j})) \right]$$

$$w \leftarrow w - \alpha c_1 \nabla_w \hat{L}^{\text{vf}}(w, \mathcal{B})$$

19 **end**

20 **end**

21 **end**

Output : θ_K .

the mean and 95% confidence interval across runs. Final performance values reported in tables are computed as the mean over the last 10 evaluation points of each run, prior to smoothing.

Diagnostics. To monitor the distributional shift induced by data reuse, we track two complementary diagnostics at every iteration. Each diagnostic is computed independently for every window index $i \in \llbracket 0, \omega_k - 1 \rrbracket$ over the corresponding rollout set $\mathcal{D}_{k-i}$, and is evaluated before the optimization phase to distinguish the shift present at the start of each iteration. Specifically, the two diagnostics are:

Hyperparameter	Ant-v5	HalfCheetah-v5	Hopper-v5	Reacher-v5	Swimmer-v5	Walker2d-v5
<i>Environment</i>						
Total timesteps	3×10^6	1×10^6	1×10^6	1×10^6	1×10^6	1×10^6
Number of environments (N_E)	1	1	1	1	4	1
<i>PPO</i>						
Learning rate	1.91×10^{-5}	2.06×10^{-5}	3×10^{-4}	1.04×10^{-4}	6×10^{-4}	5.05×10^{-5}
Steps per rollout (H)	512	512	2048	512	1024	512
Batch size	32	64	64	32	256	32
Epochs per update	10	20	10	5	10	20
Discount factor γ	0.98	0.98	0.999	0.9	0.9999	0.99
GAE λ	0.8	0.92	0.95	1.0	0.98	0.95
Clip range ϵ	0.1	0.1	0.2	0.3	0.2	0.1
Value function coefficient	0.677	0.581	0.5	0.950	0.5	0.872
Entropy coefficient	4.96×10^{-7}	4.02×10^{-4}	0.0	7.53×10^{-8}	0.0	5.85×10^{-4}
Max gradient norm	0.6	0.8	0.5	0.9	0.5	1.0
Normalize advantage	✓	✓	✓	✓	✓	✓
<i>Policy Network</i>						
Architecture	64×64	256×256	64×64	64×64	64×64	64×64
Orthogonal init	✓	×	✓	✓	✓	✓
Log std init	0.0	-2.0	0.0	0.0	0.0	0.0

Table 2: Employed hyperparameters values for conducting the experimental validation shown in Section 4 and D.

- *Clip Fraction:*

$$\text{CF}_i^\square = \frac{1}{D} \sum_{j=1}^D \mathbb{1}[w_{i,j}^\square \notin [1 - \epsilon, 1 + \epsilon]],$$

where for $\omega\text{PPO-U}$ we have $w_{i,j}^\square = \frac{\pi_{\theta}(\mathbf{a}_{i,j}|\mathbf{s}_{i,j})}{\pi_{\theta_{k-i}}(\mathbf{a}_{i,j}|\mathbf{s}_{i,j})}$ while for $\omega\text{PPO-BH}$ we have $w_{i,j}^\square = \frac{\pi_{\theta}(\mathbf{a}_{i,j}|\mathbf{s}_{i,j})}{\pi_{\omega_k}^{\text{BH}}(\mathbf{a}_{i,j}|\mathbf{s}_{i,j})}$. $\text{CF}_i^\square$ is the fraction of transitions in $\mathcal{D}_{k-i}$ whose weight $w_{i,j}^\square$ falls outside $[1 - \epsilon, 1 + \epsilon]$. It measures how frequently the clipping mechanism is active for data from window index i , and therefore how often those samples are effectively excluded from the gradient signal. However, it is insensitive to the magnitude of deviations among clipped transitions.

- *(Normalized) Effective Sample Size (ESS):*

$$\text{ESS}_i^\square = \frac{\left(\sum_j w_{i,j}^\square\right)^2}{D \cdot \sum_j (w_{i,j}^\square)^2} \in [0, 1].$$

In a standard importance weighting setting, the *effective sample size* [ESS, 26] is the number of samples drawn directly from P for which the Monte Carlo estimator of the mean has approximately the same variance as the self-normalized importance weighting estimator employing D samples from Q [35]. So, the normalized ESS is 1 whenever $P = Q$, while it decreases as the dissimilarity of the two distributions enlarges.

Each metric is averaged across all iterations of a single training run, reported for the pre-optimization phase. Plots show the mean and 95% confidence interval over 10 independent runs.

Computational Resources All experiments were conducted on a machine equipped with an AMD Ryzen Threadripper PRO 9965WX CPU (24 cores, 48 threads) and an NVIDIA RTX A1000 GPU (8 GB VRAM), running Ubuntu 24.04 LTS. A single run of PPO in *HalfCheetah-v5* considering the hyperparameters of Table 2 takes ≈ 25 minutes. Additionally, in the same setting, a run of $\omega\text{PPO-BH}$ and $\omega\text{PPO-U}$ with $\omega = 4$ in the *fixed B* scenario takes ≈ 50 minutes, while for *fixed N_B* scenario it takes ≈ 35 minutes.

LLMs Usage. We employed LLMs mainly as aid for editing purposes, for checking proof correctness, and for assistance with code.

D Additional Experiments

In this section, we provide the complete experimental campaign conducted for this work, of which Section 4 presents a subset. Our experiments aim to empirically understand when and how PPO can

benefit from sample reuse. All experiments are designed to maintain fairness with respect to PPO, as detailed in Appendix C.

PPO Mechanics and Update Accounting. Before starting, we recall that at each iteration $k \in \llbracket K \rrbracket$, PPO collects a dataset $\mathcal{D}_k$ of $|\mathcal{D}_k| = N_E H =: D^4$ samples under π_{θ_k} . For $\tilde{K}$ epochs, $\mathcal{D}_k$ is shuffled and divided into N_B mini-batches of size B , each used to perform one parameter update, so PPO performs $\tilde{K} N_B$ actor and critic updates per iteration. Each datum therefore has an update life cycle of $\tilde{K}$ updates before being discarded.

ω PPO-U and ω PPO-BH Mechanics and Update Accounting. As detailed in Section 3, rather than discarding $\mathcal{D}_k$ after each iteration, ω PPO-U and ω PPO-BH maintain a sliding window of the $\omega_k := \min\{k, \omega\}$ most recent datasets, forming the composite buffer $\mathcal{D}(\omega_k) = \bigcup_{i=0}^{\omega_k-1} \mathcal{D}_{k-i}$ of total size $\omega_k D$,⁵ from which balanced mini-batches are drawn. As in PPO, $\tilde{K}$ epochs are performed per iteration, while the number and size of mini-batches depend on the comparison setting (see below and Table 3). Because each datum remains in the buffer for ω_k iterations and is replayed for $\tilde{K}$ epochs per iteration, its update life cycle extends to $\omega_k \tilde{K}$ epochs.

Comparison Settings. The mini-batch sampling strategy from $\mathcal{D}(\omega_k)$ governs how the reuse variants differ from PPO in terms of update count, mini-batch composition, and data collection cost. We consider three settings, each designed to isolate a distinct aspect of the reuse mechanism.

- *Fixed mini-batch size (Fixed B).* The reuse variants share the same B and $\tilde{K}$ as PPO. Since $|\mathcal{D}(\omega_k)| = \omega_k D$, this yields $\omega_k N_B$ mini-batches per epoch, so the reuse variants perform $\omega_k \tilde{K} N_B$ actor updates per iteration, exactly ω_k times as many as PPO, with each mini-batch containing B/ω_k samples per window index. This setting investigates whether a larger buffer, made of fresh and past data, can support more gradient steps without additional environment interaction, and whether keeping B fixed is sufficient to avoid update instability. To isolate the effect of reuse from the increased update count, we also run a PPO baseline with $\omega_k \tilde{K}$ epochs per iteration, matching the total actor updates of the reuse variants.
- *Fixed mini-batch count (Fixed N_B).* The reuse variants keep the same N_B as PPO, yielding larger mini-batches of size $\omega_k B$. All methods therefore perform $\tilde{K} N_B$ actor updates per iteration, with each update seeing the same amount of fresh data as in PPO while additionally drawing on past samples. This setting investigates whether keeping the update count fixed while enriching each mini-batch with historical data is sufficient to improve learning, and whether mixing current and past samples within a larger mini-batch provides a benefit over using fresh data alone.
- *Reduced data collection (H/ω Steps).* The reuse variants collect only H/ω_k new environment steps per iteration, keeping B , N_B , and $\tilde{K}$ fixed at their PPO values. The composite buffer therefore has the same total size D as PPO’s dataset, but only a fraction $1/\omega_k$ of it consists of fresh transitions, and all methods perform $\tilde{K} N_B$ actor updates per iteration. This setting investigates whether stored historical data can compensate for a reduced fresh signal, allowing the policy to advance more rapidly relative to environment interaction. Among the three settings, it poses the most stringent test of sample efficiency.

Table 3 summarizes the main characteristics of the discussed comparison settings. In the following, we analyze all three settings and use diagnostic tools to investigate when and how data reuse helps in PPO.

D.1 Fixed Mini-Batch Size and Increased PPO Epochs Settings

Here, we analyze the behavior of ω PPO-U and ω PPO-BH when sharing all hyperparameters with the tuned PPO baseline (Appendix C). Specifically, the reuse variants retain the same mini-batch size B , so each mini-batch contains B/ω_k samples per reuse window and the reuse variants perform ω_k times as many actor updates per iteration as PPO. To isolate the effect of sample reuse from the increased number of updates alone, we also include a PPO baseline running $\omega \tilde{K}$ epochs per iteration, matching the total update count of the reuse variants.

⁴Where N_E represents the number of parallel environments and H the number of steps performed in each environment.

⁵The buffers are considered to have the same size since at each iteration k policy π_k is used to collect $N_E H$ samples, as previously detailed.

Setting	Environment Steps	Update Epochs	Mini-Batch Size	Mini-Batch Count	Updates per Iteration
PPO	H	$\tilde{K}$	B	$N_{\mathcal{B}}$	$\tilde{K} N_{\mathcal{B}}$
PPO ($\omega \tilde{K}$ epochs)	H	$\omega \tilde{K}$	B	$N_{\mathcal{B}}$	$\omega \tilde{K} N_{\mathcal{B}}$
$\omega \text{PPO-U}$: Fixed B	H	$\tilde{K}$	B	$\omega N_{\mathcal{B}}$	$\omega \tilde{K} N_{\mathcal{B}}$
$\omega \text{PPO-U}$: Fixed $N_{\mathcal{B}}$	H	$\tilde{K}$	ωB	$N_{\mathcal{B}}$	$\tilde{K} N_{\mathcal{B}}$
$\omega \text{PPO-U}$: H/ω Steps	H/ω	$\tilde{K}$	B	$N_{\mathcal{B}}$	$\tilde{K} N_{\mathcal{B}}$

Table 3: Summary of the main features of the considered comparison setting. Highlighted cells mark values that differ from the PPO baseline.

Figures 4 and 5 show training evaluation curves across the considered MuJoCo [57] environments and Table 4 reports final performance achieved by all methods, clustered by environment. All experiments consider three window sizes, $\omega \in \{2, 4, 8\}$. Overall, this set of experiments supports five conclusions: (i) PPO may truly benefit from sample reuse; (ii) this benefit is not merely attributable to performing more gradient updates; (iii) in this setting, data reuse does not always accelerate convergence but leads to higher final performance in most cases; (iv) neither $\omega \text{PPO-U}$ nor $\omega \text{PPO-BH}$ consistently dominates the other, with both methods behaving similarly across environments; and (v) it appears that in this setting there exists a window value for which the reuse variants exhibit most of the benefit, although this window value is environment-dependent. We now detail the results for each environment in turn.

Figure 4a reports the evaluation curves on the *Reacher-v5* environment. Even a reuse window of $\omega = 2$ yields a clear improvement in sample efficiency, with most of the gain exhibited at $\omega = 4$. Further increasing to $\omega = 8$ continues to improve sample efficiency, though with diminishing returns relative to the $\omega = 2$ to $\omega = 4$ step. Notably, data reuse not only accelerates learning but also leads to higher final performance (Table 4), with $\omega \text{PPO-BH}$ consistently achieving the best results across window sizes. As expected, more updates also increase wall-clock training time. The comparison against the $\omega \tilde{K}$ -epoch PPO baseline is particularly informative. While this baseline initially matches the sample efficiency of the reuse variants, it becomes unstable at larger values of ω , a phenomenon that does not affect $\omega \text{PPO-U}$ or $\omega \text{PPO-BH}$. This suggests that the benefits of data reuse do not stem solely from performing more updates, but from the additional data itself.

Similar observations hold for the *Swimmer-v5* environment, shown in Figure 4b. Data reuse consistently benefits PPO, though the improvement at $\omega = 2$ is mild and the advantage over the $\omega \tilde{K}$ -epoch PPO baseline is not clearly separable at this window size. The benefit becomes more pronounced at $\omega = 4$ and $\omega = 8$, where the reuse variants reach higher final performances while the $\omega \tilde{K}$ -epoch PPO baseline saturates early. Notably, unlike in *Reacher-v5*, increasing the window size does not accelerate convergence. Instead, data reuse allows the agent to extract more information from the available samples, translating into higher final performance, as also confirmed by Table 4, rather than faster initial learning.

One of the most instructive scenarios is the *Hopper-v5* environment, shown in Figure 4c. Data reuse is beneficial for $\omega \in \{2, 4\}$: in both cases, $\omega \text{PPO-U}$ and $\omega \text{PPO-BH}$ behave similarly and outperform both PPO and its $\omega \tilde{K}$ -epoch variant, with the latter improving over PPO at $\omega = 2$ but collapsing at $\omega = 4$ due to the increased number of updates. For $\omega = 8$, however, both $\omega \text{PPO-U}$ and $\omega \text{PPO-BH}$ degrade significantly, with only $\omega \text{PPO-BH}$ eventually recovering to match PPO performance given its wide confidence interval. The $\omega \tilde{K}$ -epoch PPO baseline collapses early in this regime. This setting clearly demonstrates that an optimal window size exists and that excessive reuse can actively harm performance. We conjecture this results from several compounding factors, amplified by this environment’s configuration ($H = 2048$, $B = 64$, $\tilde{K} = 10$). First, at $\omega = 8$ the reuse variants perform 2560 updates per iteration against the 320 of PPO, amplifying the bias introduced by stale advantages across many gradient steps. Second, each window contributes only $B/\omega = 8$ samples per mini-batch, meaning that just 12.5% of each actor update is driven by fresh data and that each critic update employs only 8 samples. Third, since all hyperparameters are inherited from PPO, the larger number of updates may increase the drift between consecutive policies and hence the diversity of the policies in the window. Past samples falling outside the clipping range contribute no gradient. In

Algorithm		Reacher-v5	Walker2d-v5	Hopper-v5	HalfCheetah-v5	Swimmer-v5	Ant-v5
PPO		-4.35 ± 0.063	2430 ± 443	2530 ± 137	4330 ± 725	322 ± 13.6	3640 ± 204
PPO ($\omega\tilde{K}$ epochs)	2×	-4.70 ± 0.069	2510 ± 556	2790 ± 305	4500 ± 1120	340 ± 17.5	3610 ± 203
	4×	-6.76 ± 0.308	682 ± 138	1190 ± 442	4840 ± 861	309 ± 28.3	2740 ± 266
	8×	-7.97 ± 0.531	392 ± 32.4	415 ± 317	3940 ± 810	256 ± 49.3	1890 ± 236
ω PPO-U	$\omega = 2$	-4.29 ± 0.065	3350 ± 556	3240 ± 94.3	5560 ± 335	334 ± 16.2	3650 ± 227
	$\omega = 4$	-4.25 ± 0.075	3740 ± 546	3220 ± 192	5960 ± 729	339 ± 15.0	3840 ± 254
	$\omega = 8$	-4.15 ± 0.057	3650 ± 426	2130 ± 195	6010 ± 484	346 ± 7.73	3650 ± 481
ω PPO-BH	$\omega = 2$	-4.14 ± 0.047	3360 ± 491	3220 ± 94.6	5530 ± 550	343 ± 11.2	3800 ± 189
	$\omega = 4$	-3.99 ± 0.045	3230 ± 735	3110 ± 274	6170 ± 444	347 ± 6.76	3930 ± 210
	$\omega = 8$	-3.94 ± 0.052	2660 ± 815	2120 ± 476	5720 ± 586	343 ± 3.17	3760 ± 576

Table 4: Average of the last 10 final performance values of the curves shown in the *fixed B* setting (Figures 4 and 5). Each cell shows the mean $\pm 95\%$ C.I. edges (10 runs).

Appendix D.2, we investigate what happens when PPO and its reuse variants are instead constrained to perform the same number of updates per iteration while matching the number of mini-batches N_B .

Figure 4d shows the same evaluation conducted in the *Walker2d-v5* environment. The results are broadly consistent with the other environments: data reuse is beneficial, especially for $\omega \in \{2, 4\}$. The $\omega\tilde{K}$ -epoch PPO variant slightly improves over PPO at $\omega = 2$ but fails to match the reuse variants, while for $\omega \in \{4, 8\}$ it collapses entirely. Notably, the BH correction does not appear to offer a clear advantage over uniform weighting in this environment, particularly at $\omega = 8$. Finally, while larger windows do not consistently accelerate convergence, they allow the reuse variants to reach higher final performances, as confirmed by Table 4.

Figure 5a reports the evaluation curves on the *HalfCheetah-v5* environment. Data reuse consistently leads to higher final performance across all window sizes $\omega \in \{2, 4, 8\}$, as confirmed also by Table 4. There is a visible improvement from $\omega = 2$ to $\omega = 4$, while the gain from $\omega = 4$ to $\omega = 8$ is negligible, suggesting that most of the benefit is already captured at moderate window sizes. ω PPO-U and ω PPO-BH perform comparably throughout, with neither variant offering a clear advantage over the other. The $\omega\tilde{K}$ -epoch PPO baseline exhibits faster initial learning but fails to reach the final performance of the reuse variants, and shows increased instability, possibly due to overexploiting a fixed data batch. In contrast, ω PPO-U and ω PPO-BH perform the same number of updates while leveraging a larger and more diverse composite buffer, which appears to stabilize training and support more effective policy improvement.

Figure 5b reports the evaluation curves on the *Ant-v5* environment. Unlike the other environments, data reuse does not yield a clear improvement in final performance, as confirmed by Table 4; instead, the primary benefit lies in the speed at which the agent reaches that performance. At $\omega = 2$, the $\omega\tilde{K}$ -epoch PPO baseline achieves a convergence speed comparable to the reuse variants without exhibiting instability. However, at $\omega \in \{4, 8\}$, this baseline learns faster initially but then diverges, whereas ω PPO-U and ω PPO-BH remain stable and they reach the final performance level sooner than the baseline PPO.

D.2 Fixed Mini-Batch Count Setting

Here, we analyze the behavior of ω PPO-U and ω PPO-BH when sharing all hyperparameters with the tuned PPO baseline (Appendix C), but, in contrast to Appendix D.1, we keep fixed the number of mini-batches per epoch N_B rather than the mini-batch size B . By design, the reuse variants perform the *same* number of updates per epoch as PPO, but leverage mini-batches of size ωB , thereby preserving the same amount of fresh information per update, i.e., the same number of samples from the most recent policy π_{θ_k} .

Figures 6 and 7 show training evaluation curves across the considered MuJoCo [57] environments and Table 5 reports the final performance reached by the considered methods. All experiments consider three window sizes, $\omega \in \{2, 4, 8\}$. The results of this setting can be summarized as follows: (i) PPO consistently benefits from sample reuse; (ii) in this setting the benefit manifests in both faster convergence and higher final performance, as further confirmed by Table 5; (iii) ω PPO-BH tends

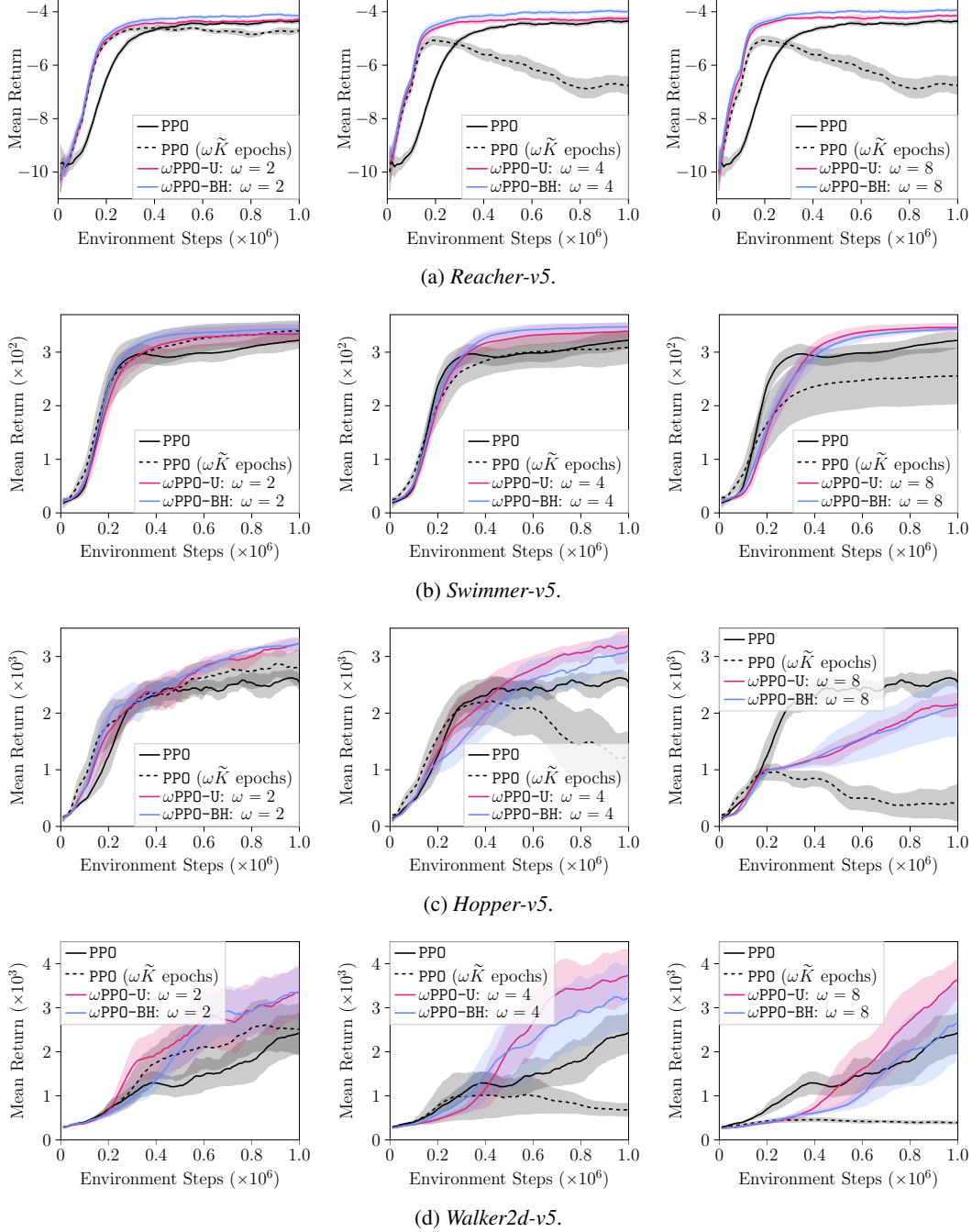


Figure 4: Comparison among PPO, ω PPO-BH, and ω PPO-U in various MuJoCo environments under different window values in the *fixed B* setting, showing the PPO version with increased epochs to match the number of updates performed by the reuse variants (10 runs, mean $\pm$ 95% C.I.).

to reach higher final performances with tighter confidence intervals, particularly for $\omega \in \{4, 8\}$; and (iv) across environments, $\omega = 4$ appears to offer the best trade-off, with larger windows providing diminishing returns.

In Figure 6a, we show results for the *Reacher-v5* environment, the simplest in our experimental campaign. The results are consistent with those of Appendix D.1, where the *baseline* setting is considered. The benefit of data reuse increases with ω : larger windows yield both faster convergence and higher final performance, with ω PPO-BH performing slightly better throughout. Importantly, all

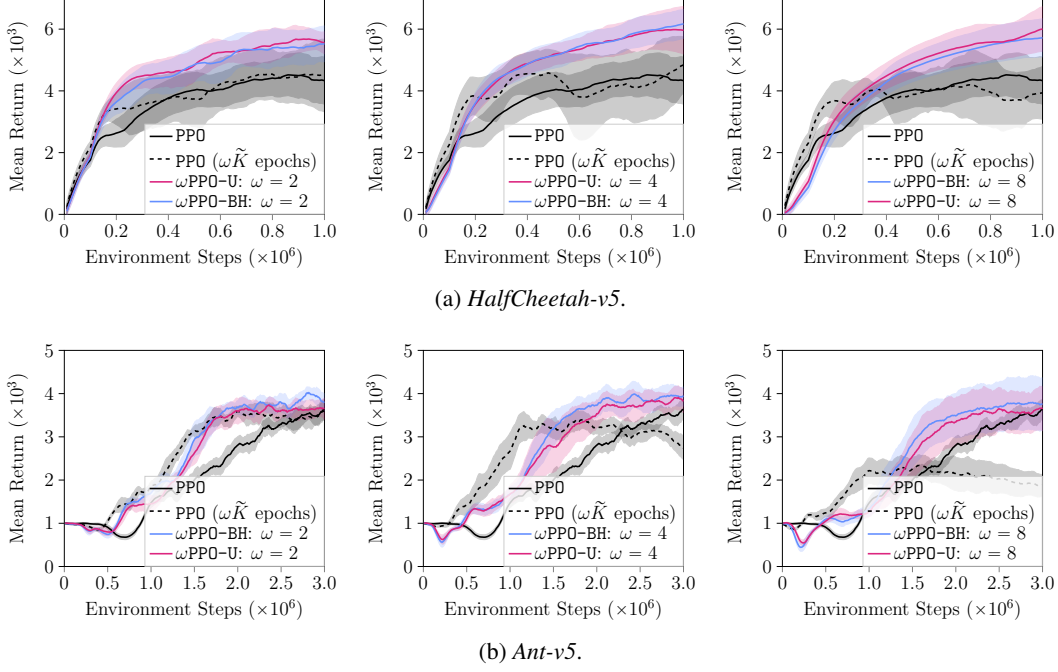


Figure 5: Comparison among PPO, ω PPO-BH, and ω PPO-U in various MuJoCo environments under different window values in the *fixed B* setting, showing the PPO version with increased epochs to match the number of updates performed by the reuse variants (10 runs, mean $\pm$ 95% C.I.).

Algorithm		Reacher-v5	Walker2d-v5	Hopper-v5	HalfCheetah-v5	Swimmer-v5	Ant-v5
PPO		-4.35 ± 0.063	2430 ± 443	2530 ± 137	4330 ± 725	322 ± 13.6	3640 ± 204
ω PPO-U	$\omega = 2$	-4.22 ± 0.060	3340 ± 535	2680 ± 163	5160 ± 515	327 ± 9.56	3910 ± 130
	$\omega = 4$	-4.17 ± 0.063	2820 ± 485	2810 ± 157	5760 ± 569	327 ± 11.2	3820 ± 491
	$\omega = 8$	-4.05 ± 0.057	3030 ± 430	2750 ± 131	5360 ± 384	334 ± 10.5	3970 ± 301
ω PPO-BH	$\omega = 2$	-4.12 ± 0.052	3250 ± 530	2740 ± 128	4580 ± 949	327 ± 7.69	3650 ± 136
	$\omega = 4$	-4.01 ± 0.045	3810 ± 642	2920 ± 159	5780 ± 558	334 ± 7.51	4030 ± 197
	$\omega = 8$	-3.96 ± 0.054	2990 ± 375	2910 ± 177	5240 ± 447	326 ± 12.6	3960 ± 357

Table 5: Average of the last 10 final performance values of the curves shown in the *fixed N_B* setting (Figures 6 and 7). Each cell shows the mean $\pm$ 95% C.I. edges (10 runs).

methods here perform the same number of updates, confirming that the gains are not an artifact of additional gradient steps.

For the other environments, similar conclusions can be drawn. At $\omega = 2$, data reuse provides a marginal improvement, while most of the benefit is observed at $\omega = 4$, particularly for ω PPO-BH. At $\omega = 8$, the pathological collapse seen in the baseline setting for *Hopper-v5* (Figure 4c) disappears, suggesting that larger mini-batches and reduced update count stabilize training. However, a hindered early convergence speed remains visible. We conjecture this is due to the larger policy changes occurring during early training: consecutive policies differ the most in this phase, so that, with a large reuse window, older samples carry the most stale advantages and come from the most distant behavioral policies.

The exception in this setting is the *Ant-v5* environment, the most complex among the ones considered. Unlike what is observed with fixed B (Appendix D.1), data reuse does not appear to help with convergence speed here and provides only a marginal improvement in final performance, with ω PPO-BH at $\omega = 4$ being the best-performing variant (Table 5), though the PPO baseline’s confidence interval slightly overlaps with it.

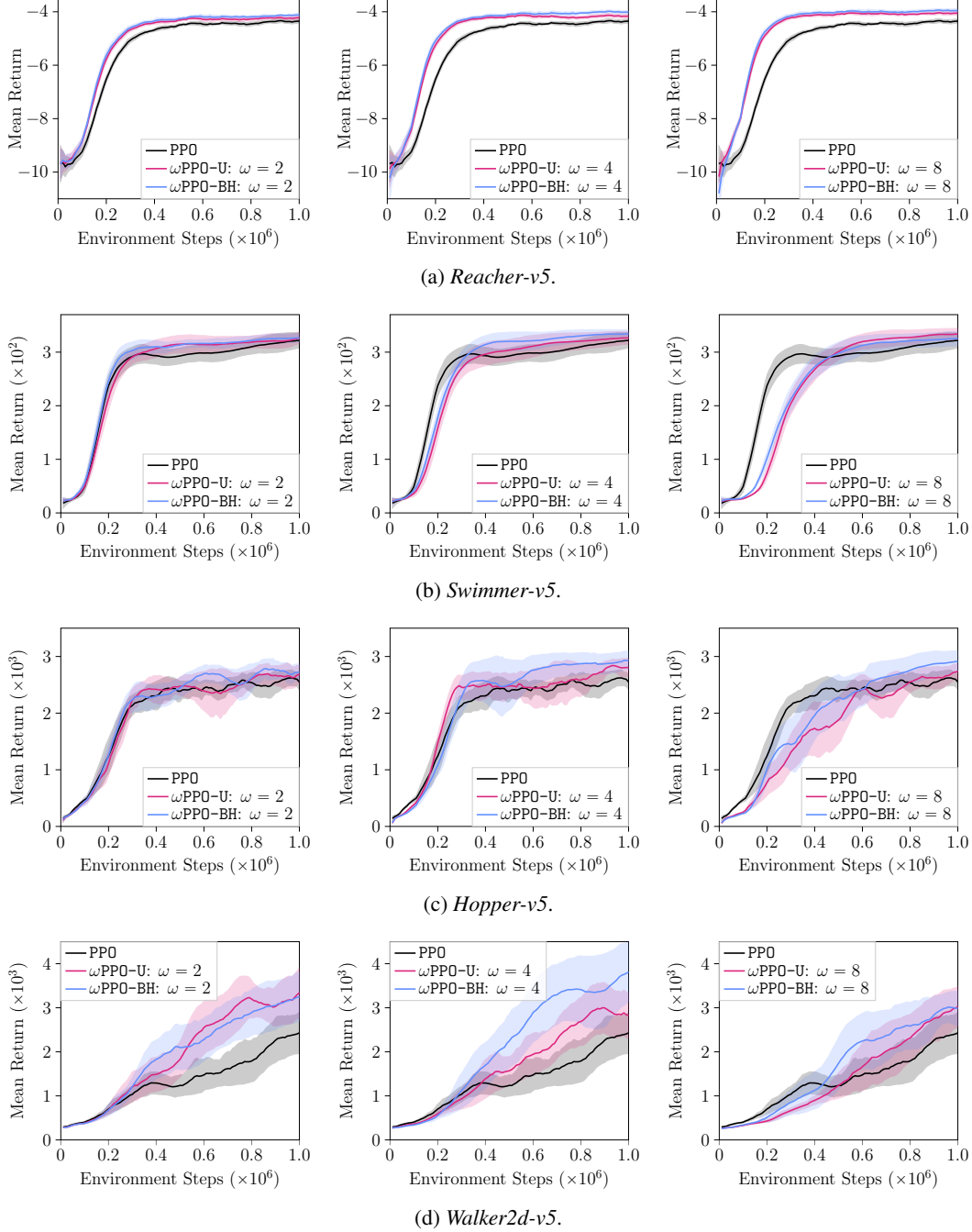


Figure 6: Comparison among PPO, ω PPO-BH, and ω PPO-U in various MuJoCo environments under different window values in the *fixed* N_B setting (10 runs, mean $\pm$ 95%).

D.3 Reduced Data Collection Setting

Here, we analyze ω PPO-U and ω PPO-BH under the most aggressive sample efficiency setting: in contrast to Appendices D.1 and D.2, we keep both N_B and B fixed at their PPO values but allow the reuse variants to collect only H/ω new samples per iteration. The reuse variants therefore perform the same number of updates as PPO over a composite dataset of the same total size, but with a fresh data fraction of $1/\omega$ per iteration.

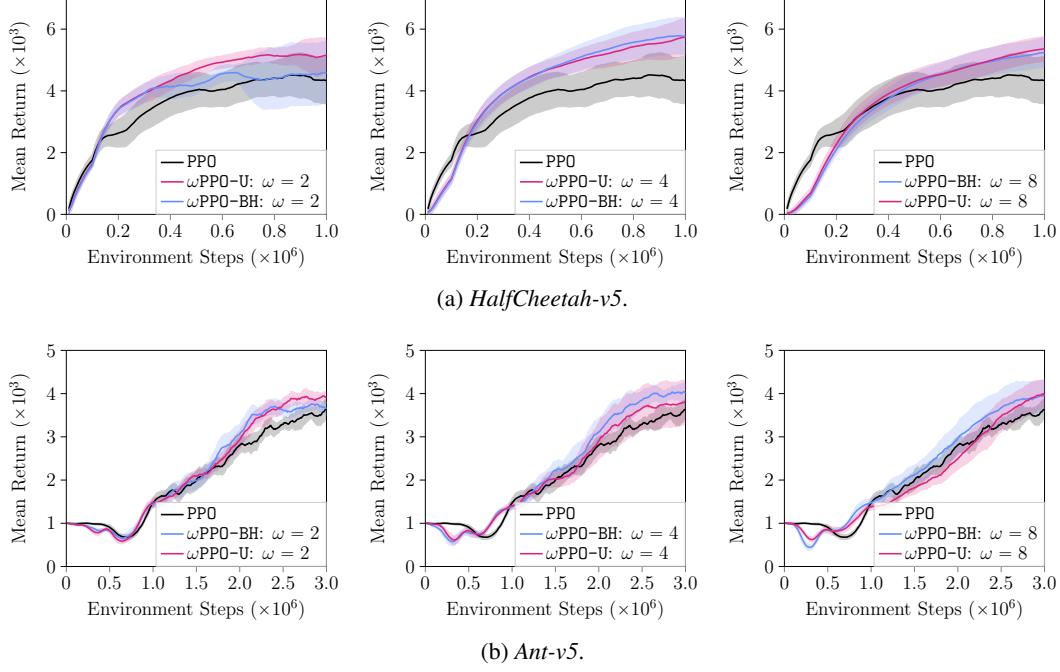


Figure 7: Comparison among PPO, ω PPO-BH, and ω PPO-U in various MuJoCo environments under different window values in the *Fixed N_B* setting (10 runs, mean $\pm$ 95%).

Algorithm		<i>Reacher-v5</i>	<i>Walker2d-v5</i>	<i>Hopper-v5</i>	<i>HalfCheetah-v5</i>	<i>Swimmer-v5</i>	<i>Ant-v5</i>
PPO		-4.35 ± 0.063	2430 ± 443	2530 ± 137	4330 ± 725	322 ± 13.6	3640 ± 204
ω PPO-U	$\omega = 2$	-4.86 ± 0.113	2300 ± 629	2920 ± 154	3960 ± 994	292 ± 51.6	3520 ± 222
	$\omega = 4$	-6.30 ± 0.342	1420 ± 300	1920 ± 553	3830 ± 373	251 ± 62.1	2670 ± 262
	$\omega = 8$	-7.78 ± 0.366	686 ± 157	438 ± 198	2710 ± 750	137 ± 70.1	1600 ± 216
ω PPO-BH	$\omega = 2$	-4.48 ± 0.084	2270 ± 463	3020 ± 163	5040 ± 711	269 ± 52.0	3500 ± 116
	$\omega = 4$	-5.06 ± 0.191	1160 ± 286	1770 ± 555	4070 ± 1080	207 ± 43.5	2150 ± 459
	$\omega = 8$	-6.63 ± 0.366	778 ± 126	777 ± 218	3440 ± 976	168 ± 57.9	865 ± 193

Table 6: Average of the last 10 final performance values of the curves shown in the H/ω steps setting (Figures 8 and 9). Each cell shows the mean $\pm$ 95% C.I. edges (10 runs).

Figures 8 and 9 show training evaluation curves and Table 6 reports final performance. All experiments consider $\omega \in \{2, 4, 8\}$. In this setting the results are unambiguous: the considered sample reuse does not help. We conjecture that the primary cause is the severely reduced amount of fresh data available to the critic. Since the critic is updated exclusively on the most recent dataset $\mathcal{D}_k$, reducing new interactions to H/ω may directly hinder it, possibly leading to degraded advantage estimates that in turn corrupt actor updates. We anticipate that this is not mitigated by fitting the critic on the whole $\mathcal{D}(\omega_k)$ (see Appendix D.5). The single exception is *Hopper-v5* at $\omega = 2$ (Figure 8c), where both ω PPO-U and ω PPO-BH still collect 1024 fresh samples per iteration, appearing to be sufficient to maintain critic quality in this environment. In all other environments, where PPO collects only $H = 512$ samples, the reuse variants are forced to operate with as few as 64 fresh samples at $\omega = 8$, falling below what appears to be the minimum threshold for stable learning.

D.4 ω PPO-U vs. ω PPO-BH

In this experiment, we aim to understand how the two importance weighting schemes adopted by ω PPO-U and ω PPO-BH differ in practice in their treatment of on and off-policy data. To this end, we present for all the considered environments and for all the comparison settings, the clip fraction and ESS diagnostics described in Appendix C. Each metric is evaluated before the optimization phase

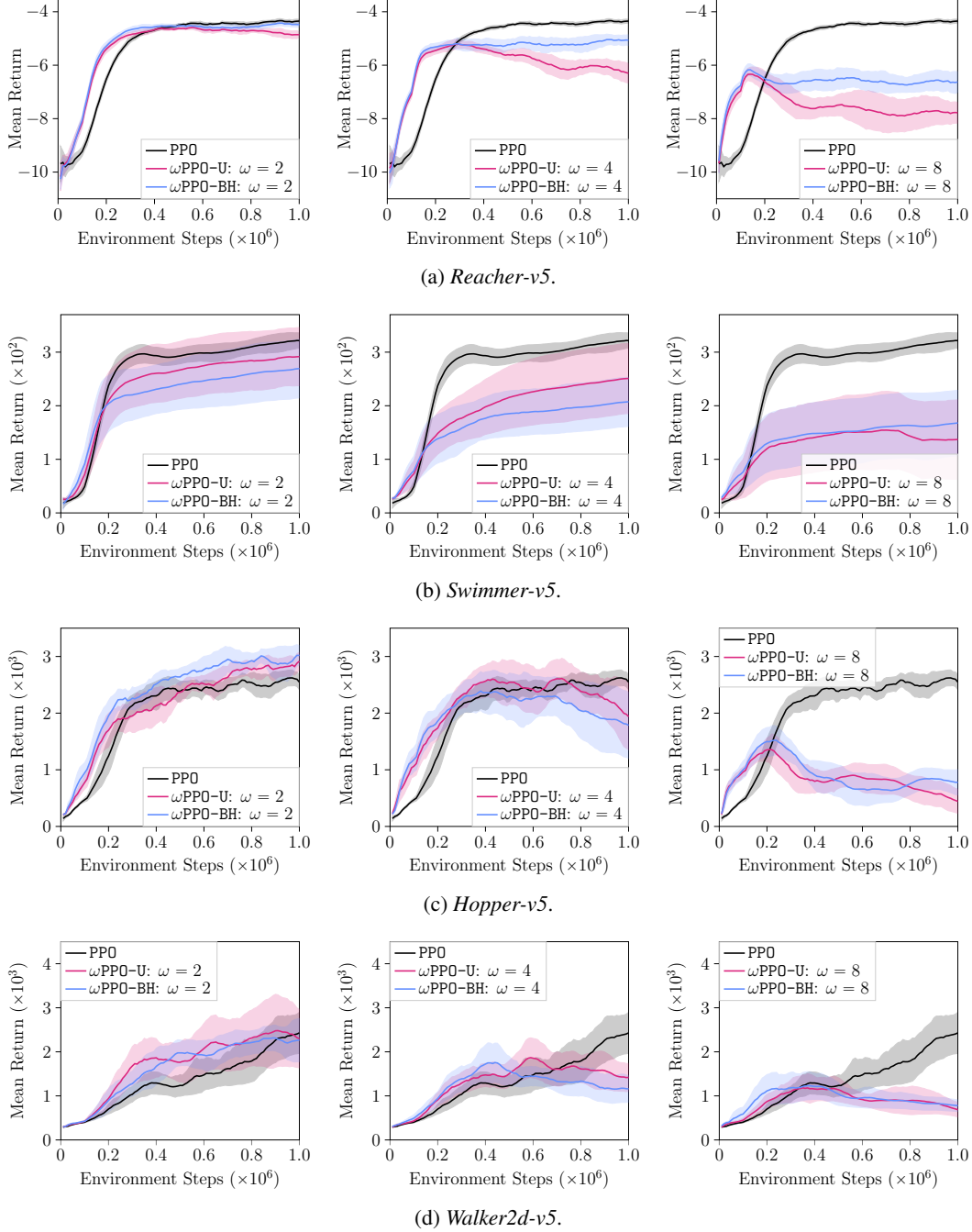


Figure 8: Comparison among PPO, ω PPO-BH, and ω PPO-U in various MuJoCo environments under different window values in the H/ω steps setting (10 runs, mean $\pm$ 95%).

at every iteration, allowing us to distinguish the distributional shift already present at the start of each iteration.

Before presenting the results, we recall that:

- The clip fraction $CF_i^\square$ ($\square \in \{U, BH\}$) captures how frequently the clipping mechanism is active for data from window index i . A high value indicates that a large fraction of transitions are effectively excluded from the gradient signal, suggesting a significant distributional shift between the current policy π_θ and the one at the denominator of the importance weight, i.e., either the policy under which samples have been collected or the uniform mixture policy.

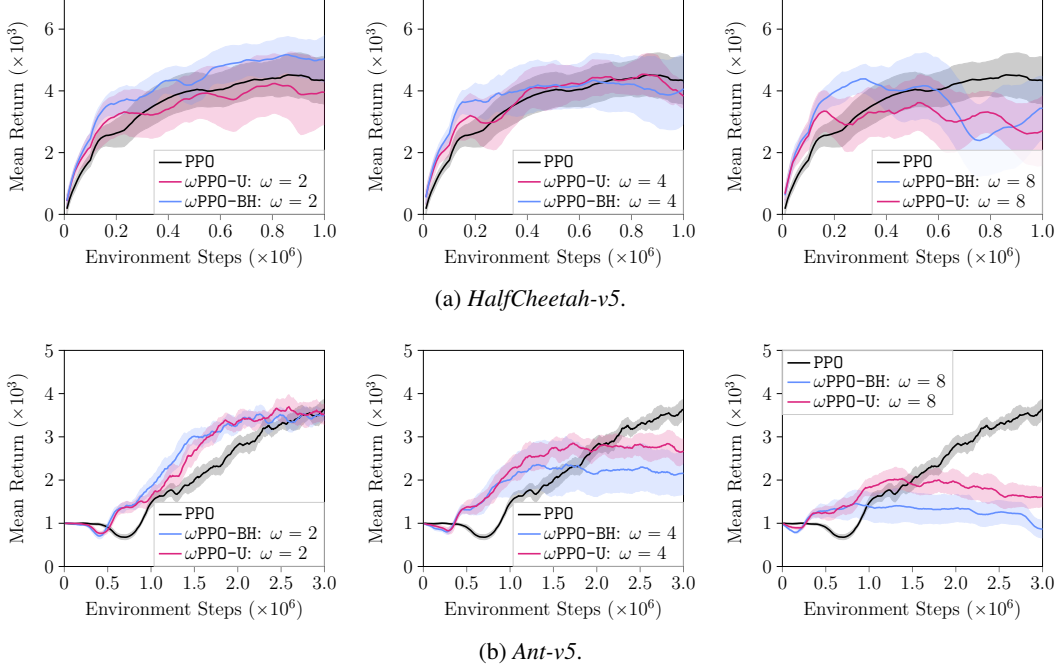


Figure 9: Comparison among PPO, ω PPO-BH, and ω PPO-U in various MuJoCo environments under different window values in the H/ω steps setting (10 runs, mean $\pm$ 95%).

- The normalized $\text{ESS}_i^\square \in [0, 1]$ ($\square \in \{\text{U}, \text{BH}\}$) measures the statistical efficiency of the importance weights: it equals 1 when the current policy matches the sampling distribution exactly, and decreases as the two distributions diverge.

Together, these two metrics provide complementary views of how stale the reused data is and how each method manages the resulting off-policy correction. Specifically, we group the plots of these metrics per environment:

- Figure 10 shows the metrics for the *Reacher-v5* environment;
- Figure 11 shows the metrics for the *Swimmer-v5* environment;
- Figure 12 shows the metrics for the *Hopper-v5* environment;
- Figure 13 shows the metrics for the *Walker2d-v5* environment;
- Figure 14 shows the metrics for the *HalfCheetah-v5* environment;
- Figure 15 shows the metrics for the *Ant-v5* environment.

Each figure reports the metric at hand averaged over the entire training and grouped by window index.

Window Index Effect Under ω PPO-U. Under the vanilla uniform weighting scheme, older data corresponding to larger window indices i consistently exhibit higher CF_i^U and lower ESS_i^U , before optimization. This is expected: since the policy has already been updated on those samples in previous iterations, the importance ratio $w_{i,j}^\text{U}$ for older transitions deviates more from 1 than for recently collected data, reflecting the growing distributional shift between the current policy and the one that generated them. This is also reflected by the ESS_i^U which decreases monotonically with window index.

The Effect of the BH Correction. Since ω PPO-BH weights all samples w.r.t. the mixture policy $\pi_{\omega_k}^\text{BH}$, both CF_i^BH and ESS_i^BH are approximately constant across all window indices before optimization, effectively equalizing the distribution of the IWs across window indices (the advantages of older samples, however, remain stale). Crucially, ω PPO-BH also maintains a substantially higher and more uniform ESS_i^BH across all window indices compared to ω PPO-U. This may allow ω PPO-BH to extract more reliable gradient signal from samples coming from older iterations. It is worth noting, however, that this equalization comes at a cost: since the BH ratios of the most recent data (window index $i = 0$) generally differ from 1, a fraction of new transitions already falls outside the clipping range before any gradient update, conversely to what happens in both PPO and ω PPO-U.

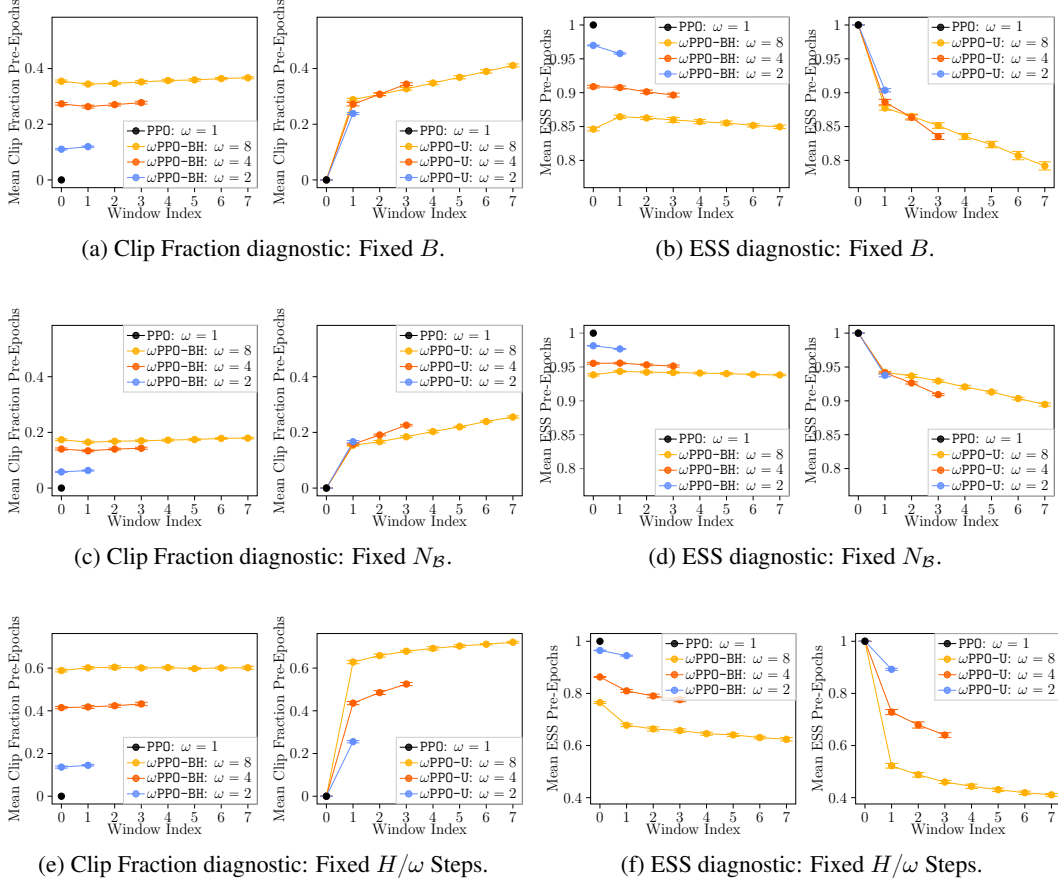


Figure 10: Clip Fraction and ESS diagnostics in *Reacher-v5* (10 runs, mean $\pm$ 95% C.I.).

D.5 Ablation: Training the Critic on $\mathcal{D}_k$ vs. on $\mathcal{D}(\omega_k)$

In this section, we conduct an ablation on whether to fit the critic on the full composite buffer $\mathcal{D}(\omega_k)$ rather than on the most recent dataset $\mathcal{D}_k$ alone, as done in all other experiments and motivated in Section 3. We consider *Reacher-v5*, *Swimmer-v5*, and *Hopper-v5* across the three comparison settings. Training evaluation curves are reported in Figures 16, 17, and 18, with overall performance in Table 7. We denote by $\omega\text{PPO-U Off}$ and $\omega\text{PPO-BH Off}$ the variants fitting the critic on $\mathcal{D}(\omega_k)$.

In general, by looking at Table 7 only and if the aim is just to find the best-performing configuration of $\omega\text{PPO-U}$ and $\omega\text{PPO-BH}$ (relative to our controlled scenarios), then one could treat the fitting of the critic of the whole buffer $\mathcal{D}(\omega_k)$ as an additional hyperparameter to explore. However, results appear to be tied to the specific starting hyperparameter configuration, considered control task, and considered comparison setting. For instance, in *Reacher-v5* fitting the critic on $\mathcal{D}(\omega_k)$ delivers the best performance across all the comparison settings, as well as in *Hopper-v5*, but for the sole *fixed* N_B scenario. Otherwise, results may vary and often there is not a clear distinction in achieved final performance and in many cases, as ω enlarges, final and training performances tends to degrade when fitting the critic on $\mathcal{D}(\omega_k)$. Next, we comment the comparison settings separately, considering to group them by ω .

In the *fixed B* setting (Figures 16a, 17a, and 18a), results are nearly identical at $\omega = 2$, but as ω increases, fitting on $\mathcal{D}(\omega_k)$ either worsens performance or leaves it unchanged across all environments. The only exception is *Reacher-v5*, where past data yields a marginal improvement in final performance, particularly for $\omega\text{PPO-BH}$, though the gain is negligible relative to fitting on $\mathcal{D}_k$ only.

In the *fixed N_B* setting (Figures 16b, 17b, and 18b), performance in *Reacher-v5* is comparable across both strategies for all window sizes, while in *Swimmer-v5* fitting on the full $\mathcal{D}(\omega_k)$ appears to

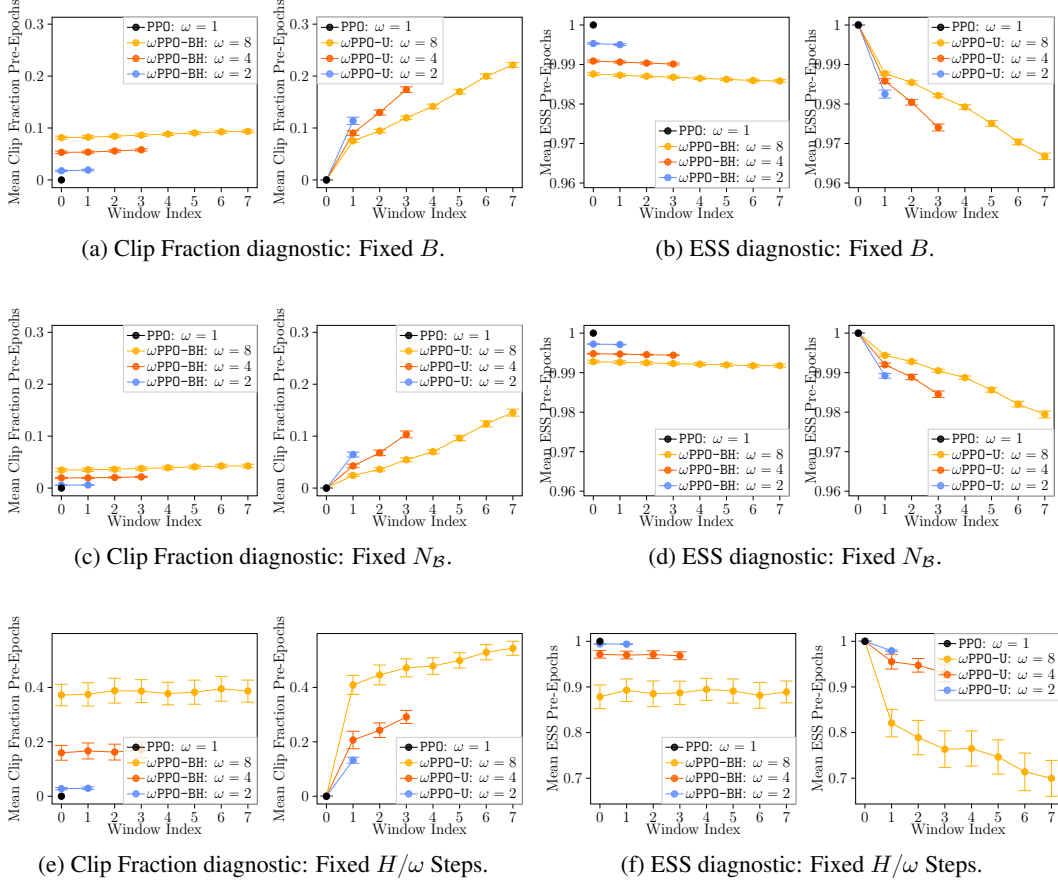


Figure 11: Clip Fraction and ESS diagnostics in *Swimmer-v5* (10 runs, mean $\pm$ 95% C.I.).

hinder training, particularly at $\omega \in \{4, 8\}$. A notable exception is *Hopper-v5*, where fitting on $\mathcal{D}(\omega_k)$ improves final performance for both variants and all window sizes.

In the H/ω steps setting (Figures 16c, 17c, and 18c), the choice of critic fitting strategy has no discernible effect across all window sizes, as all methods perform poorly regardless.

Overall, fitting the critic on, as ω enlarges, $\mathcal{D}(\omega_k)$ either shows no clear advantage or actively worsens learning, with the exceptions of *Reacher-v5* (all settings) and *Hopper-v5* in the *fixed N_B* setting only. As discussed in Section 3, past windows in $\mathcal{D}(\omega_k)$ contain return estimates computed under the critic available at collection time; without re-evaluation under the current critic [e.g., 46, 33, 69], stale returns may introduce a bias accumulating with the window size. This ablation confirms that such bias may be harmful, though its magnitude appears to depend on both the environment and the comparison setting and, as previously said, when the goal is just to find the best reuse configuration, then fitting the critic on the whole $\mathcal{D}(\omega_k)$ could be considered as an additional exploring dimension. However, for the purpose of this work and given this ablation, we decided to conduct the controlled study on reusing fitting the critic on $\mathcal{D}_k$ only.

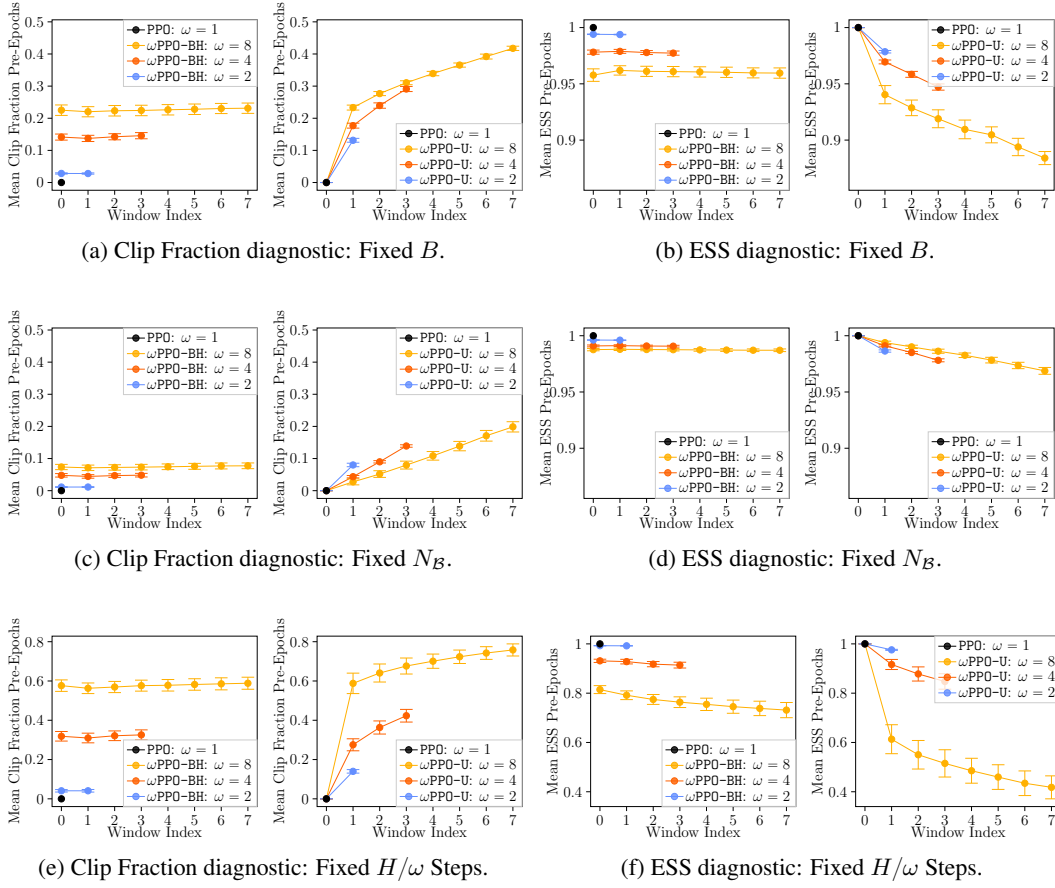


Figure 12: Clip Fraction and ESS diagnostics in *Hopper-v5* (10 runs, mean $\pm$ 95% C.I.).

Algorithm	Reacher			Hopper			Swimmer		
	Fixed B	Fixed N_B	H/ω Steps	Fixed B	Fixed N_B	H/ω Steps	Fixed B	Fixed N_B	H/ω Steps
PPO	-4.35 ± 0.063	-4.35 ± 0.063	-4.35 ± 0.063	2530 ± 137	2530 ± 137	2530 ± 137	322 ± 13.6	322 ± 13.6	322 ± 13.6
PPO ($\omega \tilde{K}$ epochs)	$2\times$ -4.70 ± 0.069	—	—	2790 ± 305	—	—	340 ± 17.5	—	—
	$4\times$ -6.76 ± 0.308	—	—	1190 ± 442	—	—	309 ± 28.3	—	—
	$8\times$ -7.97 ± 0.531	—	—	415 ± 317	—	—	256 ± 49.3	—	—
ω PPO-U	$\omega = 2$ -4.29 ± 0.065	-4.22 ± 0.060	-4.86 ± 0.113	3240 ± 94.3	2680 ± 163	2920 ± 154	334 ± 16.2	327 ± 9.56	292 ± 51.6
	$\omega = 4$ -4.25 ± 0.075	-4.17 ± 0.063	-6.30 ± 0.342	3220 ± 192	2810 ± 157	1920 ± 553	339 ± 15.0	327 ± 11.2	251 ± 62.1
	$\omega = 8$ -4.15 ± 0.057	-4.05 ± 0.057	-7.78 ± 0.366	2130 ± 195	2750 ± 131	438 ± 198	346 ± 7.73	334 ± 10.5	137 ± 70.1
ω PPO-U Off	$\omega = 2$ -4.23 ± 0.070	-4.16 ± 0.057	-4.63 ± 0.064	3200 ± 52.5	2910 ± 113	3050 ± 129	332 ± 15.3	338 ± 12.1	304 ± 41.2
	$\omega = 4$ -4.09 ± 0.045	-4.04 ± 0.040	-5.75 ± 0.201	2540 ± 649	3050 ± 77.9	1320 ± 395	306 ± 16.0	317 ± 17.4	221 ± 48.8
	$\omega = 8$ -4.01 ± 0.046	-3.94 ± 0.045	-6.93 ± 0.380	2470 ± 616	478 ± 223	260 ± 30.8	260 ± 30.8	270 ± 14.8	186 ± 47.8
ω PPO-BH	$\omega = 2$ -4.14 ± 0.047	-4.12 ± 0.052	-4.48 ± 0.084	3220 ± 94.6	2740 ± 128	3020 ± 163	343 ± 11.2	327 ± 7.69	269 ± 52.0
	$\omega = 4$ -3.99 ± 0.045	-4.01 ± 0.045	-5.06 ± 0.191	3110 ± 274	2920 ± 159	1770 ± 555	347 ± 6.76	334 ± 7.51	207 ± 43.5
	$\omega = 8$ -3.94 ± 0.052	-3.96 ± 0.054	-6.63 ± 0.366	2120 ± 476	2910 ± 177	777 ± 218	343 ± 3.17	326 ± 12.6	168 ± 57.9
ω PPO-BH Off	$\omega = 2$ -4.10 ± 0.051	-4.09 ± 0.034	-4.33 ± 0.049	3180 ± 85.9	3020 ± 126	2980 ± 159	342 ± 9.95	340 ± 13.4	304 ± 45.9
	$\omega = 4$ -3.90 ± 0.036	-3.94 ± 0.043	-4.68 ± 0.082	2700 ± 532	3070 ± 64.8	1330 ± 521	299 ± 18.3	324 ± 20.7	193 ± 48.3
	$\omega = 8$ -3.81 ± 0.056	-3.86 ± 0.044	-5.26 ± 0.162	2140 ± 652	569 ± 279	242 ± 33.2	242 ± 33.2	257 ± 23.5	170 ± 30.9

Table 7: Average of the last 10 final performance values of the curves shown in the *critic fitting ablation* (Figures 16, 17, and 18). Each cell shows the mean $\pm$ 95% C.I. edges (10 runs).

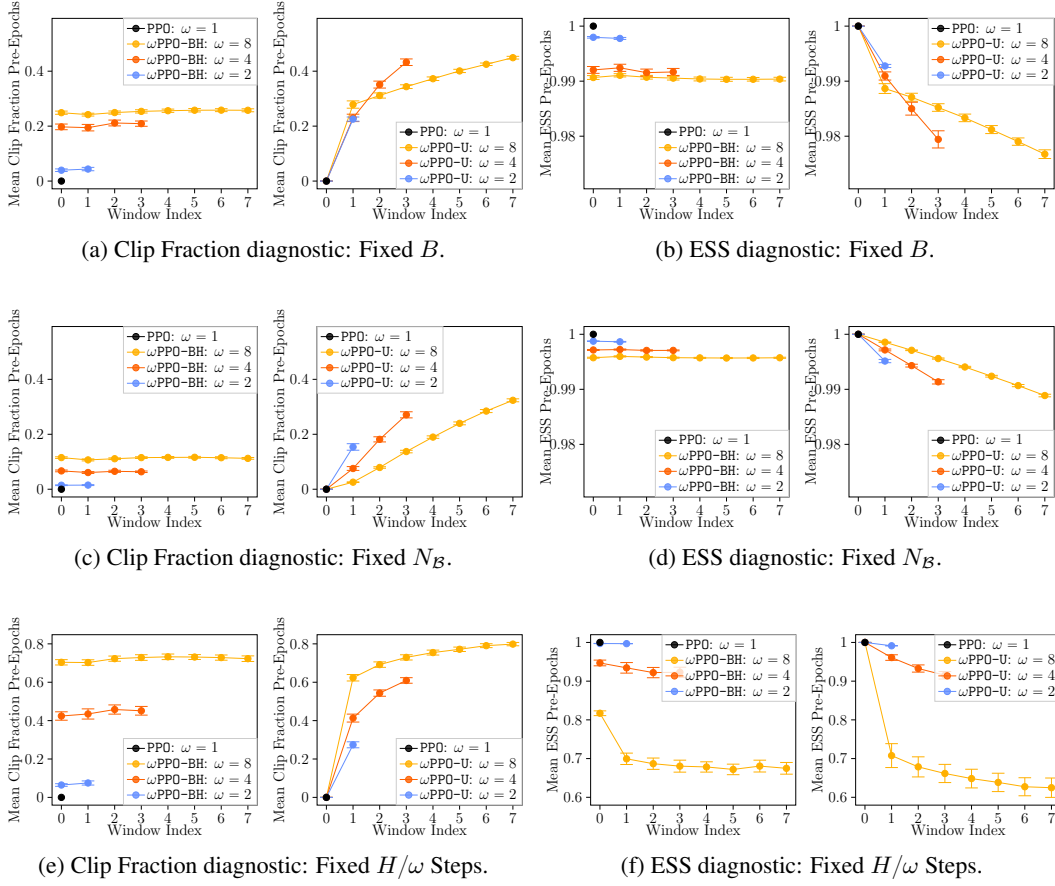


Figure 13: Clip Fraction and ESS diagnostics in *Walker2d-v5* (10 runs, mean $\pm$ 95% C.I.).

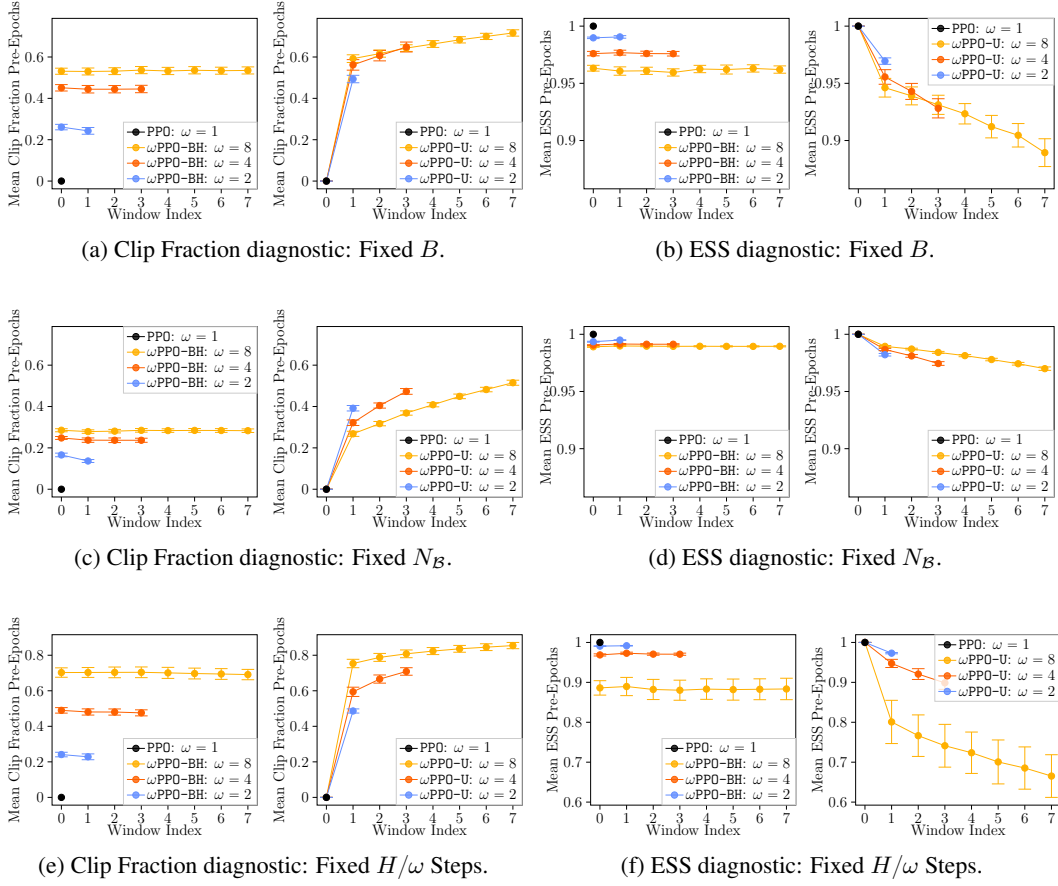


Figure 14: Clip Fraction and ESS diagnostics in *HalfCheetah-v5* (10 runs, mean $\pm$ 95% C.I.).

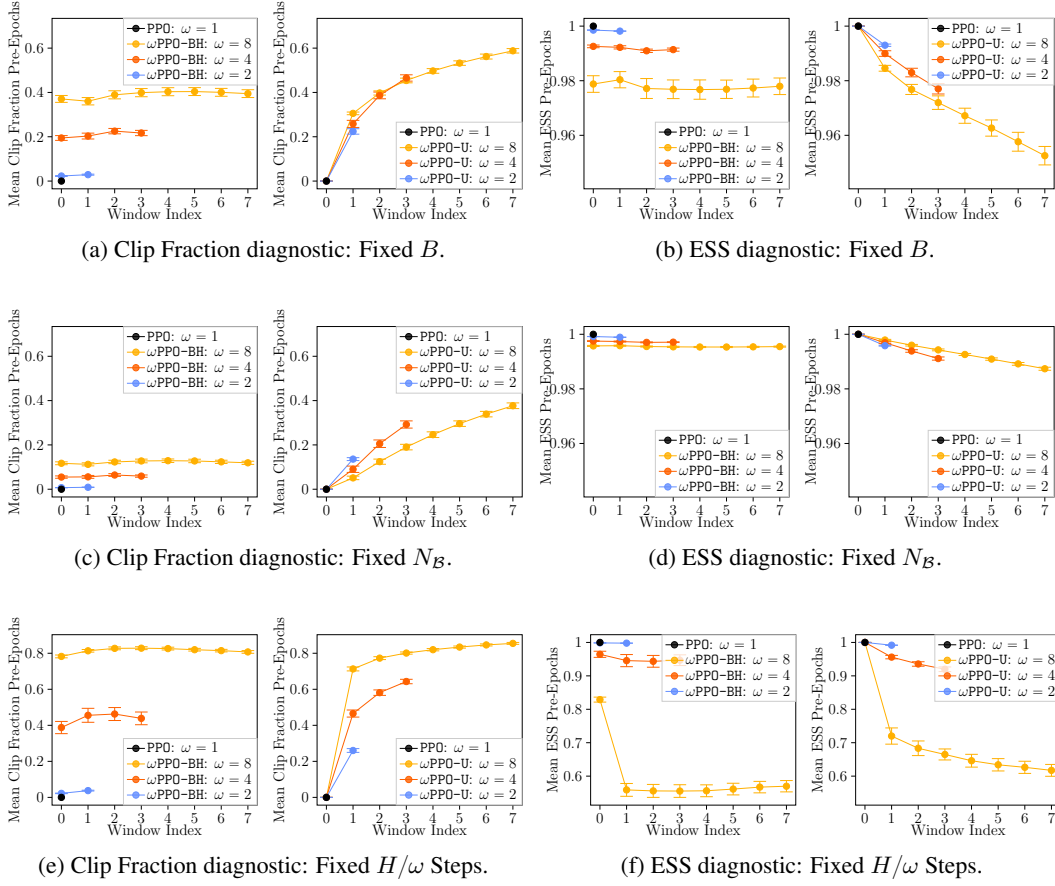


Figure 15: Clip Fraction and ESS diagnostics in *Ant-v5* (10 runs, mean $\pm$ 95% C.I.).

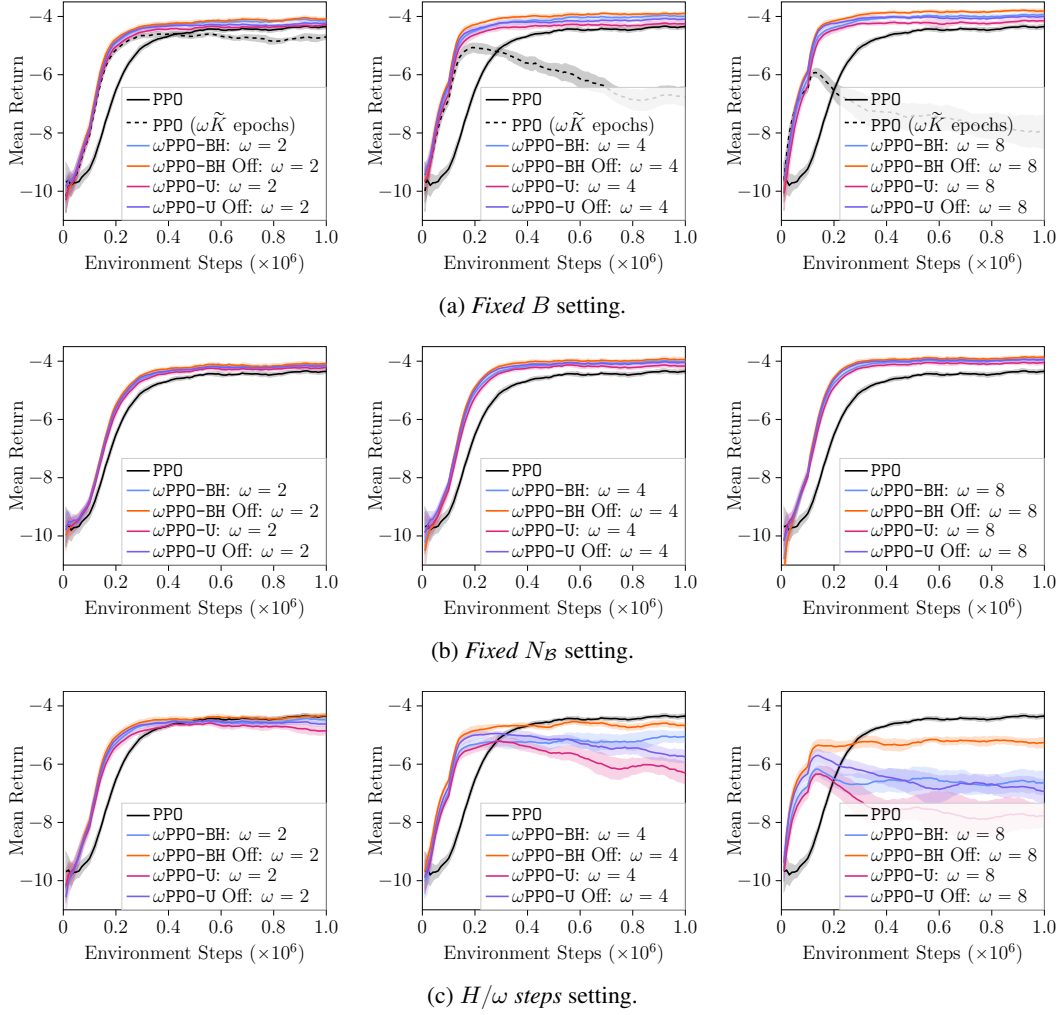


Figure 16: Ablation on training the critic on on-policy data only vs. on off-policy data too. Evaluation conducted in the *Reacher-v5* environment, considering all the presented comparison settings (10 runs, mean $\pm$ 95%).

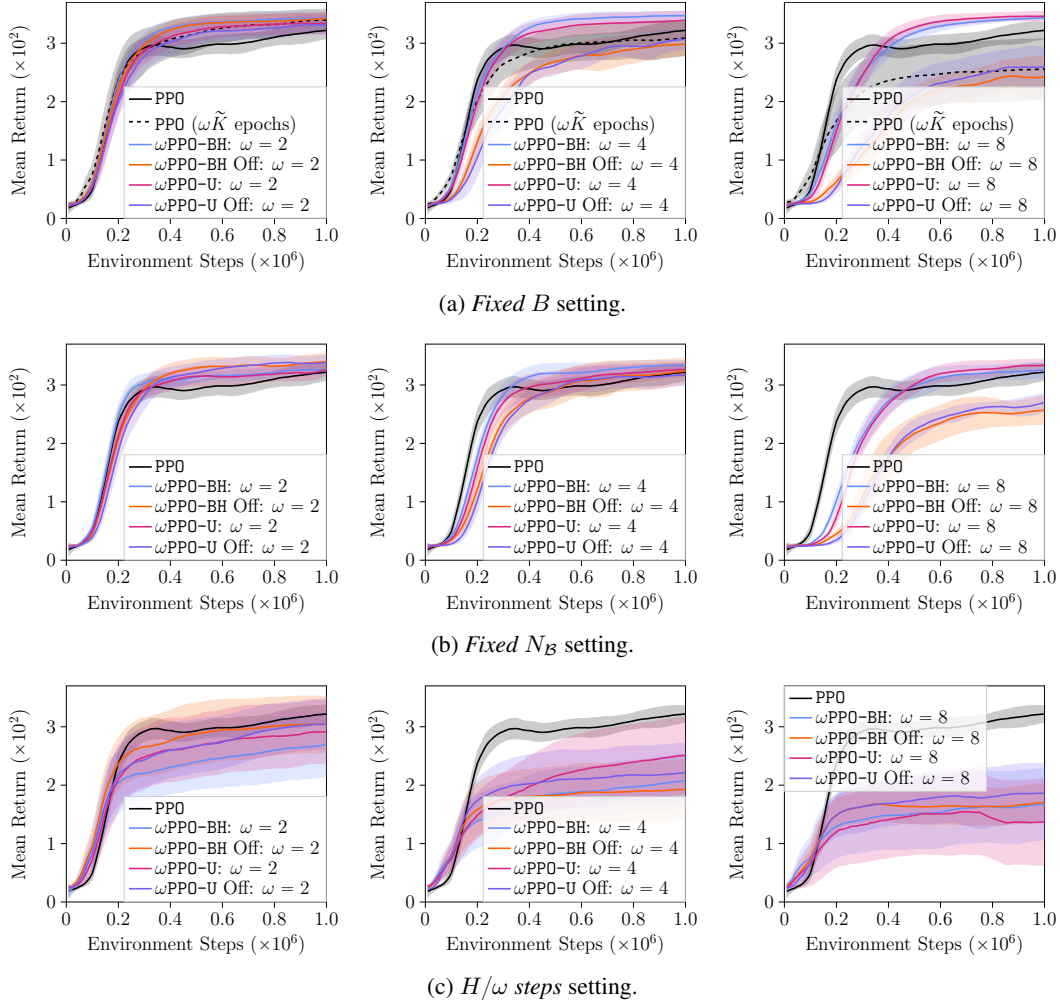


Figure 17: Ablation on training the critic on on-policy data only vs. on off-policy data too. Evaluation conducted in the *Swimmer-v5* environment, considering all the presented comparison settings (10 runs, mean $\pm$ 95%).

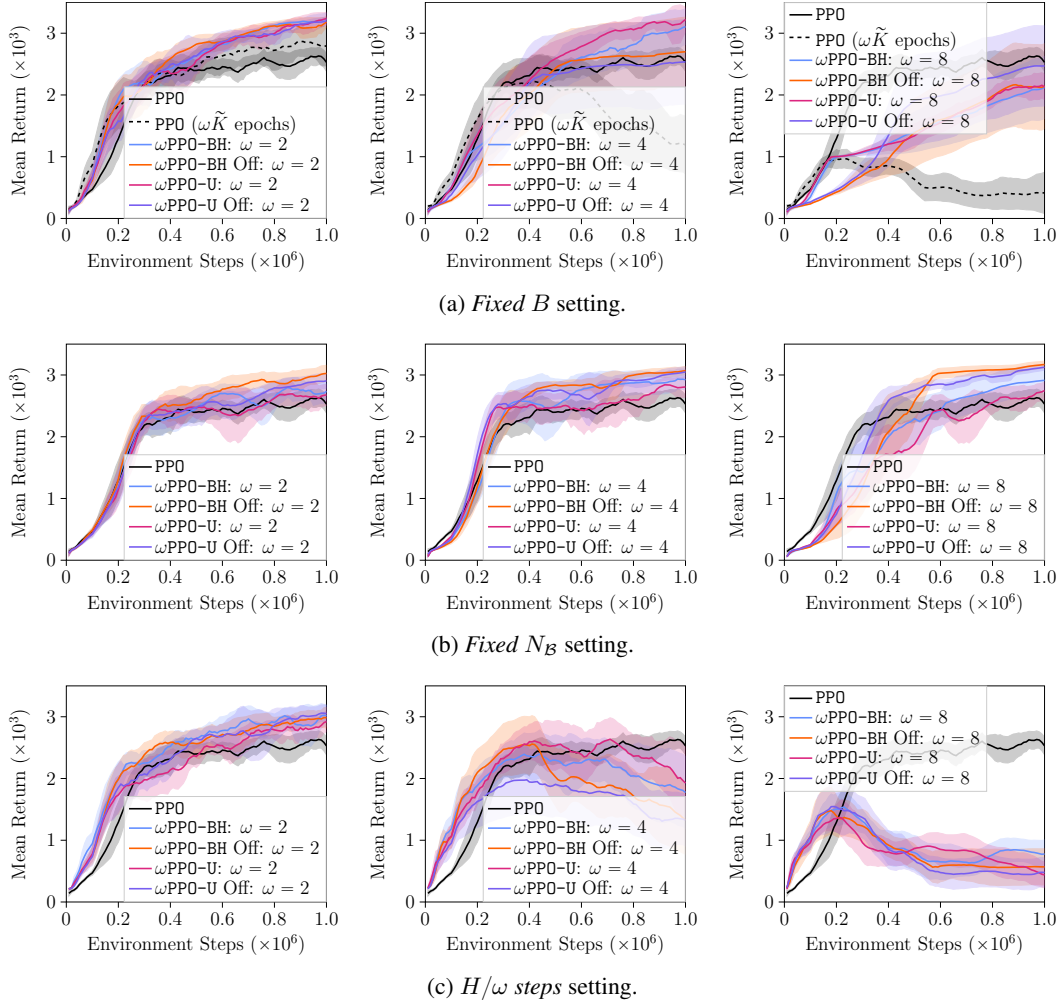


Figure 18: Ablation on training the critic on on-policy data only vs. on off-policy data too. Evaluation conducted in the *Hopper-v5* environment, considering all the presented comparison settings (10 runs, mean $\pm$ 95%).