ALTNET

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep learning methods have shown remarkable success when training from fixed datasets and in stationary environments. However, when models such as neural networks are sequentially trained on multiple tasks, their ability to learn progressively declines with each task. This phenomenon is known as plasticity loss. Previous work has shown that periodically resetting a neural network's parameters, in whole or in part, often helps restore plasticity. However, this comes at the cost of a temporary drop in performance, which can be risky in real-world settings. We introduce AltNet, a reset-based alternating network approach that mitigates plasticity loss without performance degradation by leveraging alternating twin networks. The use of twin networks anchors performance during resets and prevents performance collapse through a mechanism that allows networks to periodically switch roles: one learns as it interacts with the environment, while the other learns off-policy from the active agent's interactions and a replay buffer. At fixed intervals, the active network is reset and the passive network, having learned from the agent's prior (online and offline) experiences, becomes the new active network. We demonstrate that AltNet improves plasticity and sample efficiency, enables fast adaptation, and improves safety by preventing performance drops. In our experiments on challenging high-dimensional tasks from the DeepMind Control Suite, we show that AltNet outperforms baseline methods and various state-of-the-art reset-based techniques.

1 INTRODUCTION

Deep learning systems are often designed to learn and converge on a single task. In non-stationary environments, however, the goal being optimized by the model evolves over time. Success in such settings requires continual adaptation rather than the ability to identify a single solution. This need motivates the field of continual learning or lifelong learning, where an agent updates, accumulates, and exploits knowledge throughout its lifetime (Chen & Liu, 2018). A central obstacle in continual learning is *plasticity loss*—the progressive decline in an agent's ability to learn from new data over time (Nikishin et al., 2022; Lyle et al., 2022; Dohare et al., 2024; Kumar et al., 2020). Plasticity loss has been observed in non-stationary settings. For instance, Achille et al. (2017) showed that pre-training on blurred CIFAR images impaired subsequent learning of the original dataset. Similarly, Ash & Adams (2020) found that pre-training on half of a dataset and using the resulting model as a starting point when tackling a supervised learning task reduced accuracy compared to training on the full dataset from scratch. More broadly, Dohare et al. (2021) demonstrated that when neural networks are trained sequentially on multiple tasks, their ability to learn new tasks progressively declines with each additional task.

Reinforcement learning (RL) compounds this difficulty. Even if the task itself is stationary, RL agents face inherent sources of non-stationarity. First, agents collect their own data; as policies evolve, the distribution of encountered states and actions shifts, producing *input non-stationarity*. Second, many RL algorithms such as DQN, A2C, PPO, and SAC (Mnih et al., 2015; 2016; Schulman et al., 2017; Haarnoja et al., 2018) rely on bootstrapping, where predictions of future rewards serve as learning targets. As these predictions evolve, the targets themselves change, creating *target non-stationarity*. Together, these factors require agents to continually adapt to shifting data distributions even when tackling a single task, thereby amplifying plasticity loss. Finally, consistent with prior work (Nikishin et al., 2022; D'Oro et al., 2022), in this paper, we show that simply increasing the number of gradient updates per environment step (the replay ratio) can also exacerbate plasticity loss in RL (see Figure 1).

Various approaches have been proposed to mitigate plasticity loss (section 2). Among these, a particularly promising family of methods is based on periodically resetting network parameters (Nikishin et al., 2022; Kim et al., 2023; Dohare et al., 2024; Sokar et al., 2023). These methods leverage the observation that resets restore a network's ability to adapt. One explanation for why resets help in such settings, and the one we investigate in this work, is that plasticity is fundamentally tied to the initialization process of a neural network (Dohare et al., 2024). In particular, randomly initialized weights provide high plasticity at the start of training. However, when multiple tasks are encountered sequentially, weights learned on earlier tasks effectively become the initial weights used for solving subsequent ones. These task-specific weights often offer a worse starting point compared to random parameters, reducing the network's ability to adapt. Based on this empirical observation, one could expect that Standard Resets (Nikishin et al., 2022), which reinitialize the entire network, should in principle be the most effective way to restore plasticity. Although effective, full network resets come at a cost: they erase all information embedded in the network and cause immediate performance collapses (see Figure 3, orange curve). This renders Standard Resets impractical for real-world deployment. The central challenge we address in this paper is how to retain the benefits of full network resets in restoring plasticity while avoiding the performance instability they induce.

To address this challenge, we introduce *AltNet*, a reset-based alternating network approach that mitigates plasticity loss without inducing temporary and recurring performance collapses. AltNet maintains two networks that periodically switch roles. At any given time, the *active network* interacts with the environment, while the *passive network* learns off-policy from the active agent's experience and a shared replay buffer. At fixed intervals, the active network is fully reset and the passive network having learned from the agent's prior (online and offline) experiences becomes the new active agent. This alternating structure anchors performance across resets and prevents collapse. Importantly, AltNet successfully mitigates the negative impact of resets even in settings with a replay ratio of 1; in these cases, by contrast, vanilla resets (Nikishin et al., 2022) fail and more sophisticated methods such as RDE (Kim et al., 2023) still exhibit sharp post-reset drops (see Figure 3, blue curve). To understand what drives AltNet's gains, we systematically evaluate its robustness across design choices such as replay ratio, buffer size, and reset duration (section 5). Finally, we show that AltNet is not limited to off-policy algorithms; it improves performance in on-policy settings, as demonstrated by comparisons with the on-policy baseline, PPO (Schulman et al., 2017).

AltNet can also serve as a diagnostic tool for plasticity loss. The decline in performance of baseline SAC makes plasticity loss evident at high replay ratios (see Figure 1). However, at lower replay ratios, the effect is less obvious. Yet, AltNet's positive impact on performance (see Figure 3 and Figure 7, green curve) at lower replay ratios, suggests that plasticity loss still impacts learning in these settings. Viewed this way, AltNet's performance provides an indicator of, and to what extent, plasticity loss is constraining learning in particular lifelong learning settings. When it yields little or no benefit, baseline agent is not substantially limited. When AltNet delivers large gains, plasticity loss is the bottleneck. This diagnostic perspective parallels prior work on Plasticity Injection (Nikishin et al., 2023) in Arcade Learning Environment (Bellemare et al., 2013) and extends the idea to continuous-control domains, illustrating how full-network reset-based methods can provide empirical evidence of plasticity loss. We elaborate on the diagnostic role of AltNet in Section 3.

In summary, we (i) provide a thorough empirical analysis of plasticity loss in reinforcement learning by (ii) introduce AltNet, an alternating reset method that restores plasticity without the performance instability induced by prior approaches and achieves consistent gains in low sample complexity scenarios, such as when learning with low replay ratios, (iii) demonstrate it can extend to on-policy settings, and (iv) show that AltNet doubles as a diagnostic tool for identifying when plasticity loss is the primary factor constraining learning efficiency.

2 Related Work

Plasticity. Prior work uses the term *plasticity* to refer to the degree to which a network generalizes to unseen data Berariu et al. (2021) or to refer to its ability to continue improving performance on its training objective over time (Abbas et al., 2023; Nikishin et al., 2023; Kumar et al., 2020; Lyle et al., 2024). In this paper, we adopt the latter meaning. We say that a network has lost plasticity if it can no longer optimize its objective as effectively as a freshly initialized counterpart.

Plasticity loss in reinforcement learning. Several prior works point to the same underlying challenge: neural networks in reinforcement learning often lose their ability to adapt as training progresses. Lyle et al. (2022) observe a gradual loss of capacity to fit evolving targets even in the single task, while Kumar et al. (2020) attribute a similar effect to implicit under-parameterization. Nikishin et al. (2022) introduce the term *primacy bias*, referring to the tendency of agents to overfit to early experiences, which hinders subsequent learning. Although framed in different ways, these findings describe facets of the same phenomenon—plasticity loss. We choose the terminology *plasticity loss* because it captures the common thread across these phenomena: the gradual decline in an agent's ability to adapt to new information.

Causes of plasticity loss. The precise cause of plasticity loss remains unknown. Several correlates have been identified, such as inactive neurons, the growth of the network's average weight magnitude, the decrease in the expressivity of the network, and changes in the curvature of the loss landscape (Sokar et al., 2023; Dohare et al., 2021; Kumar et al., 2020; Lyle et al., 2023). No single correlate can, however, provides a consistent explanation across settings. For example, Lyle et al. (2023) show that for any proposed correlate, counterexamples can be constructed where the correlation disappears or even reverses. Since no single underlying cause of plasticity loss has been identified, it is difficult to determine directly whether a system has retained or lost plasticity. We therefore use performance as a practical proxy for plasticity. Following prior work (Nikishin et al., 2022; 2023; Kim et al., 2023), we evaluate plasticity loss by focusing directly on the agent's performance.

Given the difficulty of pinpointing a single cause, a wide range of algorithmic strategies have been proposed to mitigate plasticity loss. Broadly, these fall into two families: methods based on regularization, which constrain or perturb weights to preserve plasticity, and methods based on resets, which periodically reinitialize parts or the entire network. Below, we briefly review each in turn:

1. **Regularization-based strategies.** Prior work has explored regularization-based strategies to maintain plasticity. While L2 penalties can slow down weight growth, they sometimes aggravate rank collapse by biasing weights toward the origin (Dohare et al., 2021; Lyle et al., 2023). To address this, methods such as Shrink-and-Perturb (Ash & Adams, 2020) and L2 Init (Kumar et al., 2020) have been proposed, which encourage weight updates toward high-plasticity initializations while preserving feature diversity.

2. **Reset-based strategies.** Another family of methods directly resets the network in part or in whole. Nikishin et al. (2022) propose periodic full resets, relying on the replay buffer to transfer knowledge, but these often cause sharp performance drops. Igl et al. (2020) propose distilling a trained policy into a newly initialized network, which can be seen as a form of reset with distillation as the transfer mechanism. Continual Backprop (CBP) (Dohare et al., 2024) and ReDO (Sokar et al., 2023) reset subsets of neurons selected for low utility or persistent inactivity. Reset Deep Ensembles (RDE) (Kim et al., 2023) leverages full resets by maintaining an ensemble of networks, with each network reset in turn to induce plasticity. Actions are chosen through a Q-value–weighted voting scheme, where each proposed action is weighted by the critic of the oldest network in the ensemble. While RDE improves stability, it still suffers from significant post-reset performance drops (see Figure 3, orange curves) because a freshly reset, poorly trained network can still act in the environment.

3 Evidence for Plasticity Loss

In reinforcement learning, agents learn through direct interaction with the environment, which is often slow and costly in real-world domains such as robotics or healthcare applications. To maximize the utility of each interaction, practitioners often increase the *replay ratio* (RR)—the number of gradient updates performed per environment step (Fedus et al., 2020; Wang et al., 2016; D'Oro et al., 2022). In theory, larger replay ratios should accelerate learning by extracting more information from past experiences. However, prior work has shown that increasing the replay ratio exacerbates plasticity loss (Nikishin et al., 2022; D'Oro et al., 2022). We demonstrate the effect of increasing RR. At high RR, the decline in performance of baseline SAC, makes plasticity loss evident (see Figure 1). However, at lower replay ratios, the effect is less obvious. Later, we investigate whether plasticity loss persists in this regime and, if so, how it can be identified.

Plasticity Loss at High Replay Ratios We investigate how increasing the replay ratio (RR) affects the learning dynamics of the Soft Actor–Critic (SAC) algorithm in a continuous control task from the DeepMind Control Suite. We focus on the `hopper-hop` environment, a challenging locomotion domain in which the agent must learn to propel itself forward by hopping. Plasticity loss is defined as the diminishing ability of the agent to adapt to new tasks over time. In reinforcement learning, the data distribution evolves at every update, effectively presenting the agent with a new task and might causes the agent's plasticity to diminish much earlier relative to the number of samples collected. The central hypothesis is that as the number of gradient updates per environment step increases, the agent's plasticity diminishes. Concretely, if plasticity loss is present, we should observe that performance degrades once the replay ratio becomes large.

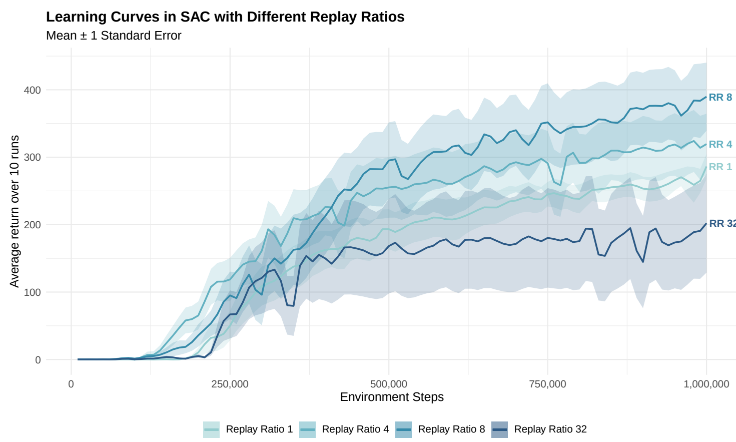


Figure 1: Learning curves of SAC in the `hopper-hop` environment (DMC) under different replay ratios (RR = 1, 4, 8, 32). Curves show mean episodic return over 10 seeds, with shaded regions denoting ± 1 standard error. Performance improves as RR increases up to 8, but collapses at RR = 32, providing direct evidence of plasticity loss.

In the `hopper-hop` environment, we trained SAC agents with replay ratios ranging from 1 to 32. As expected, performance initially improved with moderate increases in replay ratio (e.g., from RR = 1 to RR = 8), reflecting better sample efficiency. However, at RR = 32, performance decreased significantly (see Figure 1). This decline in performance arises solely from altered training dynamics, not from changes in the environment, providing direct evidence of plasticity loss. These findings align with explanations based on primacy bias (Nikishin et al., 2022), where the agent overfits to early experiences and loses adaptability as training progresses. Although prior reset-based approaches (Nikishin et al., 2022; D'Oro et al., 2022) have demonstrated recovery of plasticity under such extreme regimes, high replay ratios are computationally expensive. In `hopper-hop`, training for one million environment steps required approximately 5 GPU hours at RR = 1, but more than 200 GPU hours at RR = 32. Such scaling renders these settings impractical for real-world deployment, where rapid adaptation is crucial, motivating our shift of focus toward lower replay ratios.

Plasticity Loss at Low Replay Ratios At low RR, evidence of plasticity loss is less apparent, since standard learning curves across replay ratios do not clearly expose its presence. One natural approach to diagnose plasticity loss is to observe how a plastic system behaves. If it provides little or no benefit, the baseline agent is not substantially impaired. If it delivers substantial gains in performance, then plasticity loss is the bottleneck. Hence, such a method can serve as a diagnostic tool for plasticity loss. Though full network resets (Nikishin et al., 2022) make the system plastic, it erase accumulated knowledge and induce sharp drops in performance, making diagnosis difficult to interpret in this regime. Thus, in order to determine if a system is constrained by plasticity loss, we require a system that is both plastic and maintains stable performance.

AltNet provides precisely this combination—high plasticity without instability—making it a suitable tool for diagnosing plasticity loss. In Section 5, AltNet reveals agents at low replay ratios are constrained by plasticity loss (Figure 3). Nikishin et al. (2023) also demonstrated the ability of their proposed solution, Plasticity Injection, to be used as a diagnostic tool in the Arcade Learning Environment (Bellemare et al., 2013). In this paper, we extend this diagnostic perspective to continuous-control domains, leveraging full-network resets to demonstrate empirical evidence of plasticity loss.

4 ALTNET

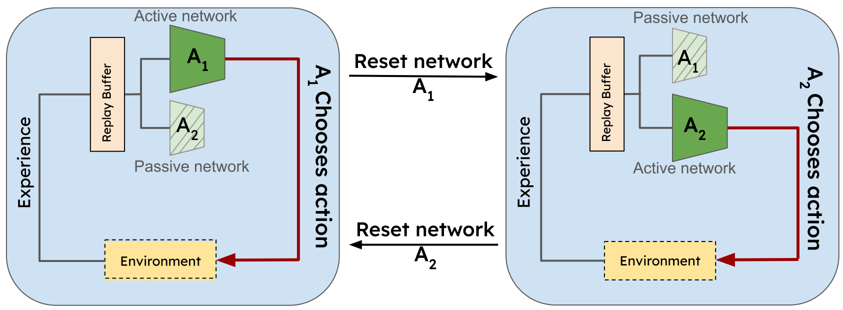


Figure 2: AltNet maintains two networks, A_1 and A_2, which share a replay buffer and alternate roles over time. Initially, A_1 (dark green) is active and collects experience by directly interacting with the environment, while A_2 (light green) remains passive and undergoes off-policy updates. At every `ResetFreq` steps, the active network is reset and becomes passive, while the previously passive network becomes active. This cyclic alternation enables frequent resets to maintain plasticity without sacrificing stability.

Central Hypothesis. Prior work has shown that full resets can restore plasticity (Nikishin et al., 2022), but they also cause sharp performance collapses when the reset policy acts immediately (see Figure 3, orange curve). We hypothesize that leveraging two insights can reconcile this plasticity-stability dilemma: (i) resetting a neural network initializes it to a highly plastic state, from which it may be able to learn a better policy, as compared to a trained network, and (ii) using well-trained networks for interaction with the environment prevents performance drops. To combine the benefits of both, AltNet introduces a dual-network architecture that allows frequent resets to occur while avoiding performance instability.

Architecture. AltNet is composed of two networks that alternate roles at a fixed interval, `ResetFreq` (Figure 2). At any given time, the active network interacts with the environment, while the passive network learns off-policy from the experiences of the active network and a replay buffer. The replay buffer is shared among the twin networks. Every `ResetFreq` steps the active network is reset and becomes the passive network and vice versa. This alternating cycle ensures that resets occur frequently enough to counter plasticity loss, yet performance remains stable because only trained networks interact with the environment.

To make the reset time comparable between methods, resets are scheduled in units of gradient updates, not environment steps. Following Kim et al. (2023); D'Oro et al. (2022), we define the reset frequency in terms of U, the number of gradient updates between resets (default $U = 200{,}000$), RR, the replay ratio (updates per environment step), and N, the number of networks, and then convert to environment steps. Formally, the effective reset interval is:

$$\texttt{ResetFreq}_{\text{(env steps)}} = \frac{U}{RR \times N} \tag{1}$$

Another crucial element of AltNet is that the shared replay buffer is preserved across resets. The size of the replay buffer is equal to the total number of interactions with the environment. If we were to apply resets to the entire system, including the buffer, that would be equivalent to training from scratch after every reset operation. What makes resets work is the preservation of data that has been collected so far in the buffer. It provides continuity of experience and serves as the medium of knowledge transfer between successive networks. While resetting the buffer is not practical, we experiment with reduced buffer sizes (Section 5.1) to test if *all* past experiences must be preserved.

Key Innovation. AltNet makes a structural departure from prior reset-based approaches. It prevents recently-reset networks from immediate interaction with the environment. In contrast, Standard Resets (Nikishin et al., 2022) expose the reset network directly to the environment, making performance collapse inevitable. RDE (Kim et al., 2023) employs ensembles with a Q-value–weighted gating policy to reduce the likelihood of a reset agent acting prematurely, although it still allows a recently reset agent to act. AltNet guarantees that only a trained network ever interacts with the environment. In AltNet reset networks first train passively on the shared buffer before taking over. This shift turns stability from a probabilistic outcome into a deterministic property. Empirically, the result is stronger and simpler: AltNet avoids post-reset performance drops across replay ratios, achieves higher and more stable returns (see Figure 3). Additionally, it offers a clearer method for diagnosing plasticity loss. More broadly, AltNet reframes reset-based learning: even with full network resets plasticity and stability can be simultaneously achieved.

5 Results and Analysis

We evaluate AltNet on continuous-control benchmarks from the DeepMind Control Suite. We compare it against SAC (Haarnoja et al., 2018), a widely used continuous-control algorithm, providing a strong and stable baseline for comparison. We also compare it with state-of-the-art methods like standard resets (Nikishin et al., 2022) and Reset Deep Ensembles (RDE) (Kim et al., 2023). All agents are trained for 1M environment interactions. All experiments use $U = 200{,}000$ updates between resets, with results reported for replay ratios of 1 and 4. Standard Reset employs a single network, whereas RDE and AltNet use two. To ensure fair comparison across methods, we define the `Reset Frequency` in terms of the number of gradient updates between resets (U), the replay ratio (RR), and the number of networks (N), following Equation 1. For example, when $RR = 1$, Standard Reset applies resets every 100k steps, while RDE and AltNet reset every 50k steps. When $RR = 4$, Standard Reset resets every 50k steps, while RDE and AltNet reset every 25k steps.

Table 1: Normalized AUC comparison of different methods across DMC environments. The best method in each environment is highlighted in bold. AltNet achieves the highest normalized AUC, outperforming SAC by ∼45%, SR by ∼10%, and RDE by ∼6% on average.

Environment	AltNet	RDE	SAC	SR
Cheetah (RR=1)	**658.27**	596.62	616.12	529.94
Hopper (RR=1)	248.68	245.69	156.69	**270.68**
Quadruped (RR=1)	**619.12**	609.36	377.27	568.36
Walker (RR=1)	**645.76**	643.22	570.08	617.06
Cheetah (RR=4)	**721.85**	619.15	535.80	620.08
Hopper (RR=4)	**313.78**	278.29	205.00	249.66
Quadruped (RR=4)	703.74	**717.24**	240.93	687.43
Walker (RR=4)	**728.49**	723.64	653.82	725.44
Average (RR=1)	**542.96**	523.22	430.04	496.51
Average (RR=4)	**616.47**	584.58	408.89	570.65

As can be seen in Figure 3, when $RR = 1$, performance of Standard Resets collapse almost immediately, and RDE experiences sharp post-reset performance drops. AltNet, by contrast, avoids these failures by anchoring resets with a passive network that learns before taking control. As a result, AltNet yields consistently higher average returns and stable learning curves across tasks. When $RR = 4$, AltNet continues to outperform SAC and Standard Resets, and matches the performance of RDE while exhibiting markedly greater stability (Appendix Figure 7). This stability is especially important in safety-sensitive domains, where abrupt failures are unacceptable.

Unlike Standard Resets (Nikishin et al., 2022), which require excessively high replay ratios and abundant compute, AltNet remains effective in the practical regime of low replay ratios. Thus, AltNet provides a scalable reset strategy that preserves plasticity, improves sample efficiency, and enhances stability under realistic computational budgets. It is the only full-network reset-based approach that eliminates post-reset drops even at $RR = 1$, demonstrating that resets can be both effective and safe in reinforcement learning. In Table 1, we report the normalized AUC, which

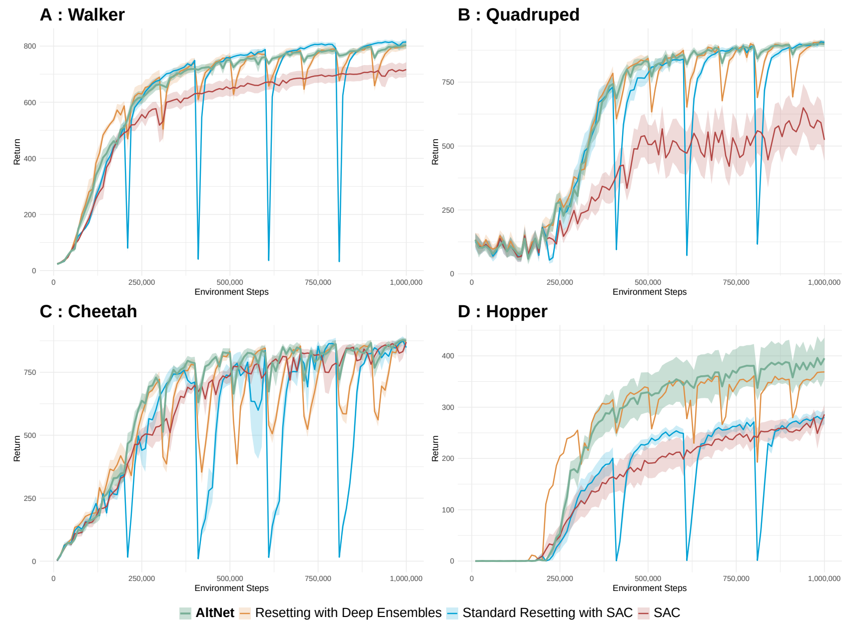


Figure 3: Learning curves across four DMC environments (Walker-run, Quadruped-run, Cheetah-run, Hopper-hop) with replay ratio = 1. Results are averaged over 10 seeds; shaded regions indicate ± 1 standard error. AltNet (green curve) avoids post-reset drops and achieves higher returns compared to SAC (red curve), Standard Resets (orange curve), and RDE (blue curve).

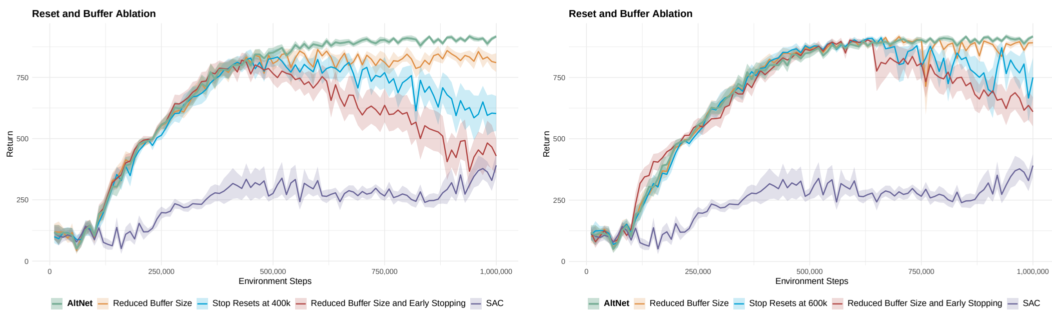
divides AUC (total return accumulated over training) by the step range to capture sample-efficiency. A higher normalized AUC indicates that the agent achieved stronger returns earlier, while a lower value reflects slower learning. AltNet achieves the highest normalized AUC, outperforming SAC by $\sim$45%, SR by $\sim$10%, and RDE by $\sim$6% on average.

These results highlight AltNet's stability and superior returns across replay ratios. A natural question is whether these benefits come at prohibitive computational cost, since AltNet maintains two networks instead of one. We find this may not always be the case. Although AltNet doubles the number of forward and backward passes, compute can be balanced simply by adjusting the replay ratio. For example, at $\mathrm{RR} = 1$, AltNet requires 12 GPU hours compared to 6 for SAC, yet it outperforms not only SAC at $\mathrm{RR} = 1$ but also SAC at $\mathrm{RR} = 4$, which takes 26 GPU hours (see Figure 9). Thus, AltNet achieves higher performance even with lower computational budget. Crucially, where prior reset-based methods demanded extreme replay ratios and high cost to remain effective, AltNet delivers plasticity and stability under realistic budgets, making it a scalable option for practice.

5.1 Buffer Size and Reset Duration

AltNet's performance relies on two interacting processes: (i) periodic resets that restore network plasticity, and (ii) preservation of a shared replay buffer that anchors performance by transferring experiences across resets. To examine the role of each component, we perform ablations that disrupt them individually and in combination (see Figure 4).

Reducing buffer size. Reset-based methods typically preserve the full replay buffer across resets (Nikishin et al., 2022; Kim et al., 2023). We investigate whether this is essential or whether older experiences can be discarded with negligible impact on AltNet's performance. Beginning with the default buffer size of 1M transitions, we reduced replay buffer capacity to 600k and 400k by replacing old samples in a FIFO manner. Reducing buffer capacity adversely affected performance (see Figure 4, orange curve): although AltNet continued to outperform non-reset baselines, performance declined relative to runs with the full buffer, showing that full replay buffer preservation is critical for stabilizing resets.



(a) Resets halted after 400k steps. Buffer size reduced to 400k.

(b) Resets halted after 600k steps. Buffer size reduced to 600k.

Figure 4: Analysis in the `Quadruped-run` environment (DMC). Curves show mean episodic return over 10 seeds, with shaded regions denoting ± 1 standard error.We compare standard AltNet (green), reduced buffer size (orange), resets halted (blue), both interventions combined (red), and the SAC baseline (purple). Results demonstrate that both preserving the full replay buffer and maintaining continuous resets are essential for AltNet's stability.

Reset Duration. Next, we investigate whether terminating resets in AltNet at some point in the training would disrupt performance. If plasticity loss accumulates over time, halting resets should cause learning progress to stall or decline once again. To test this hypothesis, we interrupted resets after 400k and 600k steps while preserving the full replay buffer. Performance declined sharply after resets were halted (see Figure 4, blue curve), indicating that plasticity loss re-emerges and that resets must be employed continuously in AltNet to sustain learning and stability.

Joint stress test. Finally, we investigate the impact of both interventions mentioned above. If resets and buffer preservation negatively impact plasticity and stability, then disrupting both should compound the decline in performance. Indeed, halting resets while also reducing buffer size produced the worst overall returns (see Figure 4, red curve). To ensure that these effects were not tied to a particular optimization setting, we repeated the procedures at a different learning rate and observed the same qualitative pattern (see Figure 8).

Together, these results show that AltNet depends on the interplay of two mechanisms: ongoing resets to restore plasticity, and replay-buffer preservation to provide continuity. Disrupting either weakens performance. These findings emphasize that AltNet must maintain both ingredients throughout training.

5.2 AltNet's Efficacy in on-policy settings

Research in plasticity loss has paid relatively little attention to on-policy reinforcement learning, even though, like off-policy methods, on-policy methods lose plasticity over long horizons and distribution shifts (Juliani & Ash, 2024; Dohare et al., 2024). Additionally, Juliani & Ash (2024) found that approaches originally developed to mitigate plasticity loss in off-policy settings such as Plasticity Injection (Nikishin et al., 2023) and CReLU (Abbas et al., 2023) are ineffective in on-policy settings. Motivated by this gap, we evaluate whether AltNet's alternating reset mechanism can also provide benefits to Proximal Policy Optimization (PPO) (Schulman et al., 2017), one of the most widely used on-policy algorithms

In on-policy reinforcement learning, agents collect trajectories by following their current policy, and updates are based solely on those trajectories. Unlike off-policy methods, these algorithms do not rely on a replay buffer for reusing past experience. Having shown that preserving the replay buffer is essential for AltNet's stability in off-policy training (see subsection 5.1), we investigate whether its benefits can extend to settings which lack a replay buffer.

Although, the absence of a replay buffer removes the explicit knowledge transfer mechanism, we find that AltNet continues to improve performance over the PPO baseline (see Figure 5). The key lies in a second, subtler form of transfer enabled by AltNet's twin-network design: while one network interacts with the environment, the other, recently reset network, learns in parallel from the same

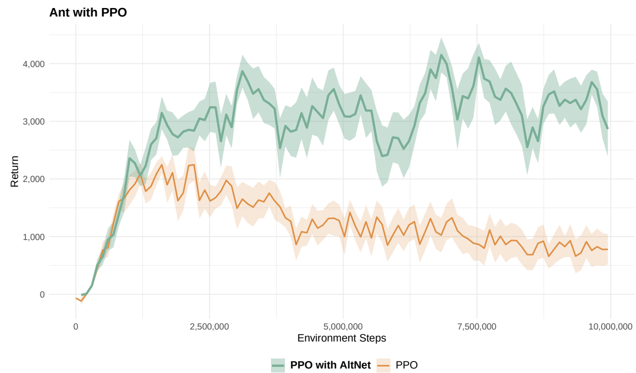


Figure 5: Training performance of PPO and PPO+AltNet in the MuJoCo `Ant` environment. Curves report mean episodic return over 10 seeds, with shaded regions denoting ± 1 standard error. AltNet provides consistent gains over PPO by anchoring resets without destabilizing learning, demonstrating its benefits extend to on-policy settings.

trajectories. Although it does not act directly, the passive network benefits from the active network's updates, enabling it to recover useful representations. This subtle knowledge transfer mechanism anchors performance across resets even without a replay buffer. Empirically, AltNet yields consistent improvements over the PPO baseline (Figure 5), demonstrating that its benefits extend beyond the off-policy setting.

6 Discussion

While our work focuses on plasticity loss, it is important to situate plasticity within the broader attributes of an effective reinforcement learning agent. Such an agent should not only remain plastic, retaining the capacity to update its predictions over time, but also adapt rapidly when distributions shift, make full use of past data, and achieve strong performance with limited interactions. This agent should also ensure performance stability that preserves prior learning progress. Plasticity underpins these other attributes: without the capacity to change, the ability to adapt quickly, exploit novel data, and learn with high sample efficiency are negatively impacted. Viewed through this lens, AltNet addresses more than plasticity loss. Through network resets, it restores plasticity in the system, enabling rapid adaptation, sustained exploitation of replay buffers, and efficient use of data at low replay ratios. And, by anchoring performance through a twin network, it is able to learn in a stable manner without performance degradation. Together, these results suggest that preserving plasticity is a foundational aspect for the broader qualities that define effective and practical RL agents.

Empirically, AltNet achieves higher performance without post-reset degradation at lower replay ratios, a setting where prior full-network reset methods lack. We further show that its benefits extend to on-policy regimes. Beyond raw performance, AltNet also functions as a more interpretable diagnostic tool for identifying plasticity loss, owing to the stability of its learning dynamics.

7 Limitations and Future Work

Although AltNet demonstrates strong empirical gains and stability across a range of continuous control tasks, our study has limitations. First, we focus our experiments on domains from the DeepMind Control Suite which are representative of challenging control problems used to evaluate state-of-the-art RL methods. That said, extending evaluation to more diverse environments could lead to further insights. Second, AltNet introduces additional compute overhead due to maintaining twin networks. While we show that this overhead is offset by lower replay ratios, real-world deployment may require tighter analysis of such computational trade-offs. Third, AltNet relies on the choice of reset frequency, which currently follows a standard schedule in our experiments; how this hyperparameter interacts with different environments and replay ratios is relevant future work.

References

Zaheer Abbas, Rosie Zhao, Joseph Modayil, Adam White, and Marlos C Machado. Loss of plasticity in continual deep reinforcement learning. In *Conference on lifelong learning agents*, pp. 620–636. PMLR, 2023.

Alessandro Achille, Matteo Rovere, and Stefano Soatto. Critical learning periods in deep neural networks. *arXiv preprint arXiv:1711.08856*, 2017.

Jordan Ash and Ryan P Adams. On warm-starting neural network training. *Advances in neural information processing systems*, 33:3884–3894, 2020.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of artificial intelligence research*, 47: 253–279, 2013.

Tudor Berariu, Wojciech Czarnecki, Soham De, Jorg Bornschein, Samuel Smith, Razvan Pascanu, and Claudia Clopath. A study on the plasticity of neural networks. *arXiv preprint arXiv:2106.00042*, 2021.

Zhiyuan Chen and Bing Liu. *Lifelong machine learning*. Morgan & Claypool Publishers, 2018.

Shibhansh Dohare, Richard S Sutton, and A Rupam Mahmood. Continual backprop: Stochastic gradient descent with persistent randomness. *arXiv preprint arXiv:2108.06325*, 2021.

Shibhansh Dohare, J Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A Rupam Mahmood, and Richard S Sutton. Loss of plasticity in deep continual learning. *Nature*, 632(8026): 768–774, 2024.

Pierluca D'Oro, Max Schwarzer, Evgenii Nikishin, Pierre-Luc Bacon, Marc G Bellemare, and Aaron Courville. Sample-efficient reinforcement learning by breaking the replay ratio barrier. In *Deep Reinforcement Learning Workshop NeurIPS 2022*, 2022.

William Fedus, Prajit Ramachandran, Rishabh Agarwal, Yoshua Bengio, Hugo Larochelle, Mark Rowland, and Will Dabney. Revisiting fundamentals of experience replay. In *International conference on machine learning*, pp. 3061–3071. PMLR, 2020.

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

Maximilian Igl, Gregory Farquhar, Jelena Luketina, Wendelin Boehmer, and Shimon Whiteson. Transient non-stationarity and generalisation in deep reinforcement learning. *arXiv preprint arXiv:2006.05826*, 2020.

Arthur Juliani and Jordan Ash. A study of plasticity loss in on-policy deep reinforcement learning. *Advances in Neural Information Processing Systems*, 37:113884–113910, 2024.

Woojun Kim, Yongjae Shin, Jongeui Park, and Youngchul Sung. Sample-efficient and safe deep reinforcement learning via reset deep ensemble agents. *Advances in Neural Information Processing Systems*, 36:53239–53260, 2023.

Aviral Kumar, Rishabh Agarwal, Dibya Ghosh, and Sergey Levine. Implicit under-parameterization inhibits data-efficient deep reinforcement learning. *arXiv preprint arXiv:2010.14498*, 2020.

Clare Lyle, Mark Rowland, and Will Dabney. Understanding and preventing capacity loss in reinforcement learning. *arXiv preprint arXiv:2204.09560*, 2022.

Clare Lyle, Zeyu Zheng, Evgenii Nikishin, Bernardo Avila Pires, Razvan Pascanu, and Will Dabney. Understanding plasticity in neural networks. In *International Conference on Machine Learning*, pp. 23190–23211. PMLR, 2023.

Clare Lyle, Zeyu Zheng, Khimya Khetarpal, Hado van Hasselt, Razvan Pascanu, James Martens, and Will Dabney. Disentangling the causes of plasticity loss in neural networks. *arXiv preprint arXiv:2402.18762*, 2024.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Volodymyr Mnih et al. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning*, pp. 1928–1937, 2016.

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

Evgenii Nikishin, Junhyuk Oh, Georg Ostrovski, Clare Lyle, Razvan Pascanu, Will Dabney, and André Barreto. Deep reinforcement learning with plasticity injection. *Advances in Neural Information Processing Systems*, 36:37142–37159, 2023.

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

Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. The dormant neuron phenomenon in deep reinforcement learning. In *International Conference on Machine Learning*, pp. 32145–32168. PMLR, 2023.

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.

A LEARNING CURVES IN DMC

In Section 5, we reported that AltNet consistently achieves higher normalized AUC than baselines across the DeepMind Control Suite. Here we provide the full learning curves to complement those summary statistics. These plots illustrate training dynamics over 1M environment steps under different replay ratios, highlighting both stability and return profiles.

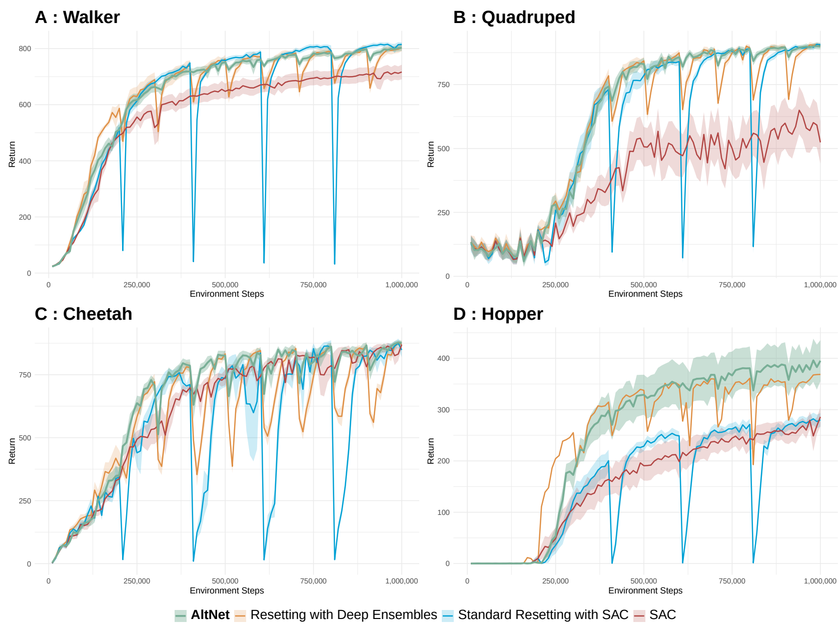


Figure 6: Learning curve with replay ratio = 1. Results are averaged over 10 seeds; shaded regions indicate ± 1 standard error.

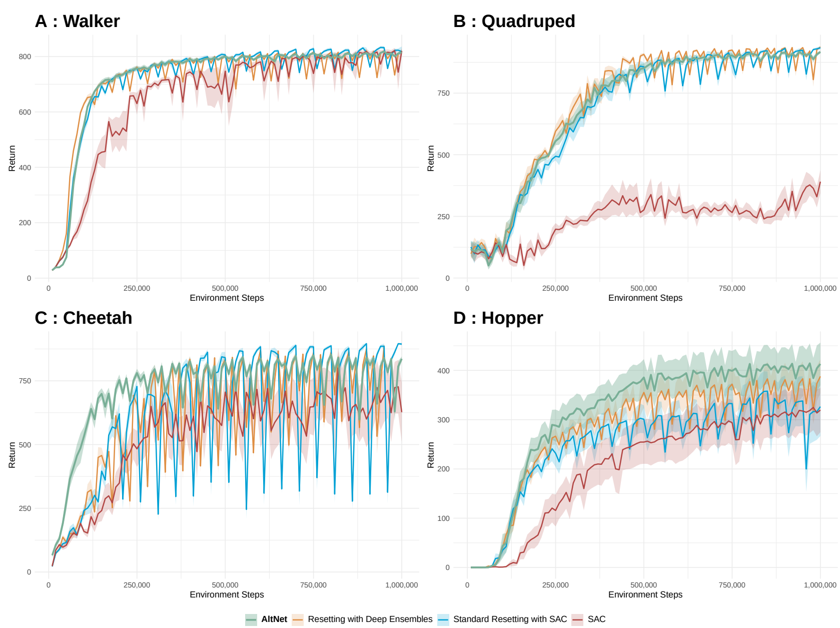


Figure 7: Learning curve with replay ratio = 4. Results are averaged over 10 seeds; shaded regions indicate ± 1 standard error.

The corresponding normalized AUC comparisons across environments are provided below to quantify sample efficiency. These results directly extend Table 1 in the main text by showing environment-level breakdowns at replay ratios of 1 and 4.

Table 2: Normalized AUC comparison of different methods across environments. The best method in each environment is highlighted in bold. (RR = 1)

Environment	AltNet	RDE	SAC	SR
Cheetah	**658.27**	596.62	616.12	529.94
Hopper	248.68	245.69	156.69	**270.68**
Quadruped	**619.12**	609.36	377.27	568.36
Walker	**645.76**	643.22	570.08	617.06

Table 3: Normalized AUC across environments for AltNet, RDE, SAC, and SR The best method in each environment is highlighted in bold. (RR=4).

Environment	AltNet	RDE	SAC	SR
Cheetah	**721.85**	619.15	535.80	620.08
Hopper	**313.78**	278.29	205.00	249.66
Quadruped	703.74	**717.24**	240.93	687.43
Walker	**728.49**	723.64	653.82	725.44

B HYPERPARAMETERS USED

The hyperparameters for AltNet and baseline agents follow standard practice in continuous-control reinforcement learning, with modifications only where required for resets. For reproducibility, we report the exact values used in the DMC environment.

Table 4: Hyperparameters on the `Hopper-Hop` domain.

Hyperparameters	**Value**
# of network (AltNet and RDE)	2
# of network (Baseline and Standard Reset)	1
Training steps	1×10^6
Discount factor	0.99
Warm up period	5000
Minibatch size	1024
Optimizer	Adam
Optimizer : learning rate	0.0003
Networks : activation	ReLU
Networks : n. hidden layers	2
Networks : neurons per layer	1024
Initial Temperature	1
Replay Buffer Size	1×10^6
Updates per step (Replay Ratio)	(1, 4)
Target network update period	1
τ (Polyak update)	0.005
Reset Frequency (gradient steps) for all	2×10^5
β (action select coefficient) for RDE	50

C BUFFER SIZE AND RESET DURATION

In Section 5.1, we argued that AltNet's stability depends jointly on two mechanisms: preserving the replay buffer across resets and maintaining resets throughout training. Here, we stress-test these design choices under different learning rates. The results confirm that both ingredients are indispensable; disrupting either one weakens AltNet's performance, and disrupting both leads to the largest degradation.

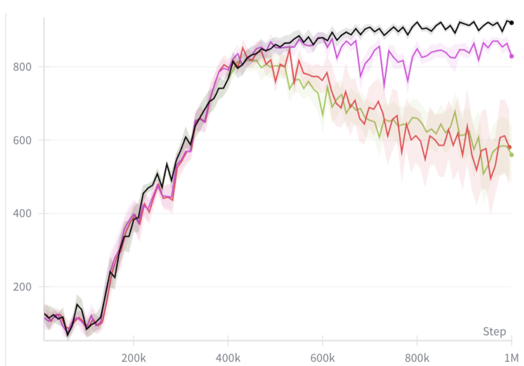


Figure 8: Consistent pattern observed under a different learning rate.

D ADDRESSING ALTNET'S COMPUTE COST

Section 5 highlighted that AltNet achieves higher returns with greater stability than SAC and other baselines, even at low replay ratios. A natural concern is whether these gains come at disproportionate computational expense, since AltNet doubles the number of networks. Here we compare wall-clock costs across methods and replay ratios.

Although AltNet doubles the number of forward and backward passes, compute can be balanced simply by adjusting the replay ratio. For example, at $\mathrm{RR} = 1$, AltNet requires 12 GPU hours compared to 6 for SAC, yet it outperforms not only SAC at $\mathrm{RR} = 1$ but also SAC at $\mathrm{RR} = 4$, which takes 26 GPU hours (see Figure 9). Thus, AltNet achieves higher performance even with lower computational budget.

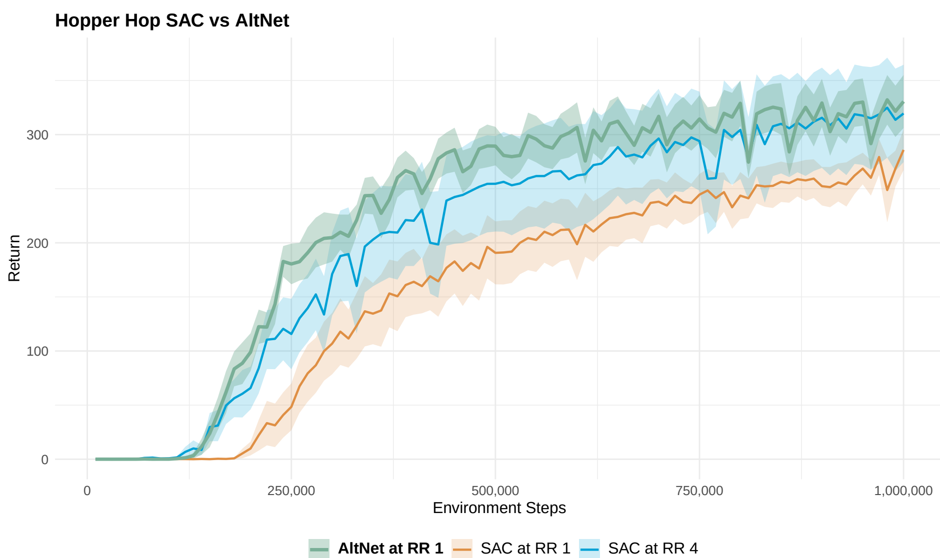


Figure 9: Compute–performance trade-off. AltNet at $\mathrm{RR} = 1$ requires fewer GPU hours than SAC at $\mathrm{RR} = 4$, yet delivers higher returns.