

# AltNet: Addressing the Plasticity-Stability Dilemma in Reinforcement Learning

Mansi Maheshwari\*  
University of Massachusetts  
Amherst, United States  
mmaaheshwari@umass.edu

John C. Raisbeck  
University of Massachusetts  
Amherst, United States  
jraisbeck@umass.edu

Bruno Castro da Silva  
University of Massachusetts  
Amherst, United States  
bsilva@umass.edu

## ABSTRACT

Artificial neural networks have shown remarkable success in supervised learning when trained on a single task using a fixed dataset. However, when neural networks are trained on a reinforcement learning task, their ability to continue learning from new experiences declines over time. This decline in learning ability is known as plasticity loss. To restore plasticity, prior work has explored periodically resetting the parameters of the learning network, a strategy that often improves performance. However, such resets come at the cost of a temporary drop in performance, which can be dangerous in real-world settings. To overcome this instability, we introduce AltNet, a reset-based approach that restores plasticity without performance degradation by leveraging a pair of “twin” networks. The use of twin networks anchors performance during resets through a mechanism that allows networks to periodically alternate roles: one network learns as it acts in the environment, while the other learns off-policy from the active network’s interactions through a replay buffer. At fixed intervals, the active network is reset and the passive network, having learned from prior experience, becomes the new active network. AltNet restores plasticity, improving sample efficiency and achieving higher performance, while avoiding performance drops that pose risks in safety-critical settings. We demonstrate these advantages in several high-dimensional control tasks from the DeepMind Control Suite, where AltNet outperforms various relevant baseline methods, as well as state-of-the-art reset-based techniques.

## KEYWORDS

Plasticity; Stability; Plasticity-Stability Dilemma; Continual Learning; Reinforcement Learning; Network Resets

## ACM Reference Format:

Mansi Maheshwari, John C. Raisbeck, and Bruno Castro da Silva. 2026. AltNet: Addressing the Plasticity-Stability Dilemma in Reinforcement Learning. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 7 pages. <https://doi.org/10.65109/KXRD5206>

\*Corresponding author.



This work is licensed under a Creative Commons Attribution International 4.0 License.

## 1 INTRODUCTION

Many deep learning systems are designed to be trained on a single task and to converge to a single solution. 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 fields of continual and lifelong learning, where an agent updates, accumulates, and exploits knowledge throughout its lifetime [5]. A central obstacle in continual learning is *plasticity loss*—the progressive decline in an agent’s ability to learn from new data over time [6, 14, 15, 21]. We say that a network has lost plasticity if its performance can no longer be improved as effectively as a freshly initialized counterpart [16]. Plasticity loss has been observed in non-stationary settings. For instance, Achille et al. [2] showed that pre-training on blurred CIFAR images impaired subsequent learning of the original dataset. Similarly, Ash and Adams [3] 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. [7] demonstrated that when neural networks are trained sequentially on multiple tasks, their ability to learn new tasks declines with each additional task.

Reinforcement learning (RL) compounds the difficulty of maintaining plasticity over time because, even when the task itself is stationary, RL agents face inherent sources of non-stationarity. First, in online RL, 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 [10, 18, 19, 22] rely on bootstrapping, where predictions of future rewards themselves 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.

To mitigate plasticity loss, various approaches have been proposed (Section 2). Among these, a particularly promising family of methods is based on periodically resetting network parameters [6, 13, 21, 23]. Resets are effective because they restore the network to a well-conditioned, highly plastic initialization that is gradually lost during training. As networks adapt to specific tasks or data distributions, they accumulate pathologies—such as dormant neurons, increasing weight magnitudes, and reduced rank—that impair their ability to learn from new data [6]. Resetting the parameters removes these accumulated effects and reinitializes the network to conditions resembling its original, plastic initialization. Nikishin et al. [21] empirically demonstrated that resetting a network can substantially improve performance by renewing its ability to learn

and exploit data. Although effective, full network resets come at a cost: they discard learned models and can cause an immediate performance drop (for an example, see Figure 2, orange curve). This makes Standard Resets [21] impractical for real-world deployment. In this paper, we address the challenge of retaining the benefits of full network resets in restoring plasticity while avoiding the performance instability they induce.

To address the plasticity-stability dilemma, we introduce *AltNet*, a reset-based alternating network approach that preserves plasticity without inducing recurring performance drops. 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 reset and the passive network, having learned from prior experience, becomes the new active network. This alternating structure anchors performance during resets and prevents performance collapse. Importantly, AltNet successfully leverages resets while significantly reducing performance instability even with little experience replay; in these cases, by contrast, Standard Resets [21] fail and more sophisticated methods such as Resets with Deep Ensembles (RDE) [13] still exhibit sharp post-reset performance drops (see Figure 2). To understand which factors contribute to AltNet’s superior performance, we systematically evaluate aspects such as model capacity, number of networks, replay ratio, buffer size, and reset duration (Subsection 4.2). Finally, we show that AltNet also improves performance in on-policy settings, as demonstrated by comparisons with relevant on-policy baselines (Subsection 4.3).

## 2 RELATED WORK

**Plasticity.** Prior work uses the term *plasticity* to refer to the degree to which a network generalizes to unseen data [4] or to refer to its ability to continue improving performance on its training objective over time [1, 14, 16, 20]. In this paper, we adopt the latter meaning. We say that a network has lost plasticity if its performance cannot be improved 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. [15] observe a gradual loss of capacity to fit evolving targets even in a single task, while Kumar et al. [14] attribute a similar effect to implicit under-parameterization. Nikishin et al. [21] 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 studies describe facets of the same phenomenon—plasticity loss. We choose the terminology *plasticity loss* because it captures the common thread: 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, growth of the network’s average weight magnitude, decrease in the expressivity of the network, and changes in the curvature of the loss landscape [7, 14, 17, 23]. None of the proposed correlates, however, provide a consistent explanation across

settings. For example, Lyle et al. [17] show that for any proposed correlate, counterexamples can be constructed where the correlation disappears or even reverses. Since the underlying cause of plasticity loss has not been identified, it is difficult to directly determine whether a system has retained or lost plasticity. Following prior work [13, 20, 21], we use performance as a proxy for plasticity.

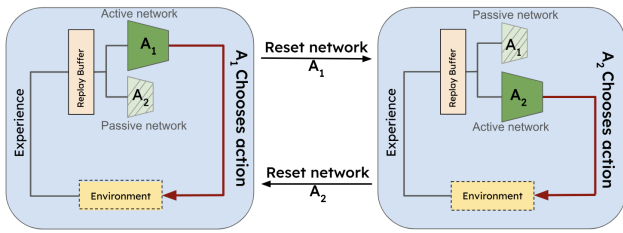
A wide range of strategies have been proposed to mitigate plasticity loss. Broadly, these fall into two families: methods based on regularization, which constrain or perturb weights to preserve plasticity, and methods based on resets, which periodically reinitialize parts or the entire network. Below,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

- (1) **Regularization-based strategies.** Prior work has explored regularization-based strategies to maintain plasticity. While  $\ell^1$  and  $\ell^2$  weight penalties can slow down weight growth, they sometimes aggravate rank collapse by biasing weights toward the origin [7, 17]. To address this, methods such as Shrink-and-Perturb [3] and L2 Init [14] 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 whole or in part. Continual Backprop [6] and ReDO [23] reset subsets of neurons selected for low utility or persistent inactivity. Igl et al. [11] 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. Nikishin et al. [21] propose periodic full network resets, relying on a replay buffer to transfer knowledge, but these cause sharp performance drops. In this paper, we refer to this approach as Standard Resets. Reset Deep Ensembles (RDE) [13] leverages full network resets by maintaining an ensemble of networks, resetting one network at a time 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. Although RDE improves stability over Standard Resets [21], it still suffers from significant post-reset performance drops because a freshly reset, untrained network can still act in the environment.

## 3 ALTNET

In this section, we introduce AltNet, a dual-network reset-based architecture designed to restore plasticity while maintaining stability in reinforcement learning.

**Central Hypothesis.** Prior work has shown that resets can restore plasticity [21], but they also cause sharp performance collapses when the reset policy acts immediately. We hypothesize that leveraging two insights can reconcile this plasticity-stability dilemma: (i) resetting a network initializes it to a highly plastic state, from which it may be possible 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 resets to occur while avoiding performance instability.



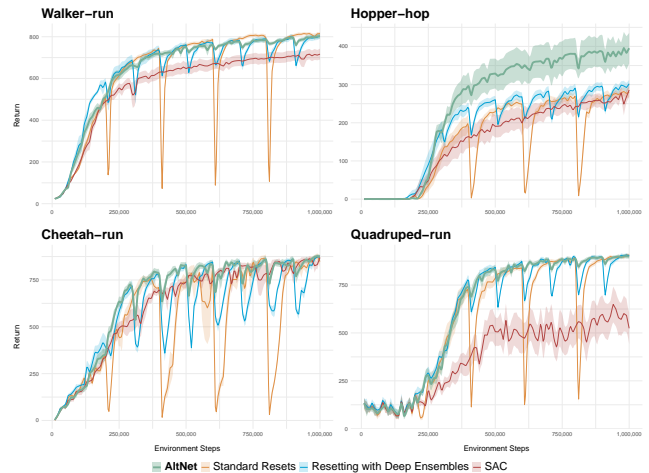
**Figure 1:** AltNet maintains two networks,  $A_1$  and  $A_2$ , which share a replay buffer and alternate roles over time. Initially,  $A_1$  (dark green) is active and collects experience by directly interacting with the environment, while  $A_2$  (light green) remains passive and undergoes off-policy 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.

**Architecture.** AltNet is composed of two networks that alternate roles at a fixed interval, `ResetFreq` (Figure 1). 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 between 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.

**Key Innovation.** AltNet makes a structural departure from prior reset-based approaches. It prevents recently-reset networks from acting in the environment until they have received sufficient training. In contrast, Standard Resets [21] expose the reset network directly to the environment, making immediate performance collapse inevitable. RDE [13] employs ensembles with a Q-value-weighted gating policy to reduce the likelihood that a reset agent acts prematurely, but still allows recently reset networks to act. AltNet, on the other hand, guarantees that only trained networks interact with the environment. In AltNet, reset networks first train passively before taking over. Empirically, the resulting mechanism is more robust and simpler: AltNet avoids post-reset performance drops across replay ratios and achieves higher and more stable returns (see, e.g., Figure 2). More broadly, AltNet shows that even with full network resets, plasticity and stability can be simultaneously achieved.

## 4 RESULTS AND ANALYSIS

In this section, we present empirical evidence that AltNet effectively addresses the plasticity–stability dilemma. We begin by contextualizing its performance through comparisons with established reset-based and non-reset baselines in diverse continuous-control environments. Next, we analyze the contributions of replay buffer preservation, network alternation, and periodic resets to higher



**Figure 2:** Learning curves in four DMC environments when replay ratio = 1. Results are averaged over 10 seeds; shaded regions indicate  $\pm 1$  standard error. AltNet (green) avoids post-reset drops and achieves higher returns compared to SAC (red), Standard Resets (orange), and RDE (blue).

performance. Finally, we extend our investigation to on-policy settings, where replay buffers are absent, to examine how alternating resets influence learning dynamics in these regimes.

We first evaluate AltNet on stationary continuous-control tasks from the DeepMind Control Suite [24]. We focus on the off-policy setting and use Soft Actor–Critic (SAC) [10] as the underlying algorithm for all methods, and as our baseline. SAC leverages a replay buffer to reuse past experiences. Later, we combine AltNet with PPO to assess whether our method’s benefits extend to on-policy settings, which do not rely on a replay buffer.

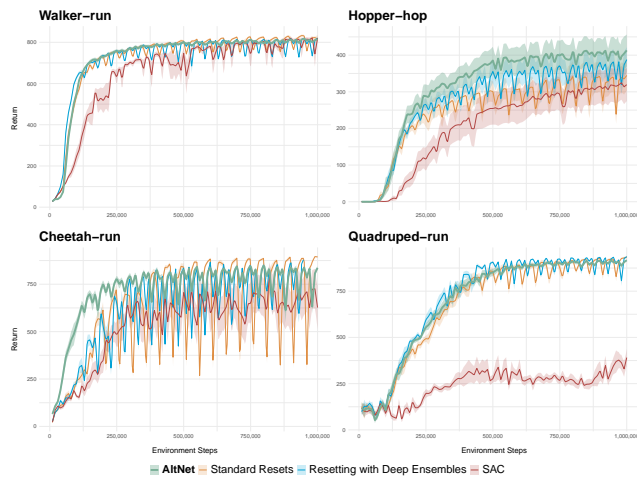
All agents are trained for 1M environment updates. The reset interval is fixed at 200,000 gradient updates ( $U$ ). We vary the replay ratio  $RR \in \{1, 4\}$ , defined as the number of gradient updates per environment step. To ensure a fair comparison, we adopt the reset frequency normalization proposed by Kim et al. [13], which accounts for both the replay ratio ( $RR$ ) and the number of networks ( $N$ ), as shown in Eq. (1). We compare AltNet against the non-reset-based baseline SAC [10], as well as state-of-the-art full network reset-based methods: Standard Resets [21] and Resets with Deep Ensembles (RDE) [13].

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

As shown in Figure 2, when  $RR = 1$ , Standard Resets’ performance collapses almost immediately, while RDE exhibits sharp post-reset drops in average return. In contrast, AltNet avoids these instabilities by anchoring performance during resets through a twin network that assumes control after each reset. Consequently, AltNet achieves higher average returns and smoother learning curves across tasks.

When  $RR = 4$ , AltNet continues to outperform SAC and Standard Resets while maintaining stable training dynamics and avoiding post-reset performance drops. In all environments except one, AltNet matches or exceeds the performance of RDE (see Table 1). In

Quadruped-run, RDE achieves a slightly higher normalized AUC, while AltNet yields more stable training dynamics (Figure 3).



**Figure 3: Learning curves in four DMC environments when replay ratio = 4. Results are averaged over 10 seeds; shaded regions indicate  $\pm 1$  standard error. AltNet (green) avoids post-reset drops and achieves higher returns compared to SAC (red), Standard Resets (orange), and RDE (blue).**

We also report, in Table 1, the mean normalized *area under the learning curve* (AUC), which summarizes the overall learning trajectory of an agent. The AUC integrates performance over time, capturing both higher returns and greater stability: agents that maintain strong performance throughout training achieve a higher AUC. We normalize the AUC to rescale the values into a consistent range, making comparisons more interpretable. As shown in Table 1, AltNet achieves the highest AUC in 7 out of 8 combinations of environments and replay ratios, outperforming SAC by approximately 38%, SR by 12%, and RDE by 6% on average.

### 4.1 AltNet Enhances Sample Efficiency

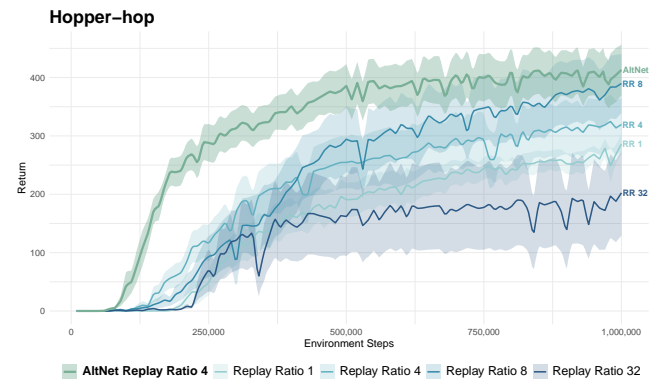
In this section, we evaluate whether AltNet improves performance and sample efficiency compared to an SAC baseline. We also compare AltNet against SAC trained at higher replay ratios—a common strategy to improve sample efficiency.

In reinforcement learning, agents learn through direct interaction with the environment, which is often slow and expensive in real-world domains such as robotics or healthcare applications. This makes *sample efficiency*—learning as much as possible from limited interactions—a central concern. A common strategy to improve sample efficiency is to increase the *replay ratio* (RR), defined as the number of gradient updates performed per environment step [8, 9, 25]. Higher replay ratios allow agents to reuse past experiences more extensively, thereby extracting additional learning signals from limited data. However, increasing RR also linearly increases computational cost and, beyond a point, can degrade performance due to overfitting to outdated experiences [8, 21]. As shown in Figure 4, increasing the replay ratio from 1 to 8 improves SAC’s performance by allowing more updates per sample, but performance

**Table 1: Normalized average AUC 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 approximately 38%, SR by 12%, and RDE by 6% on average.**

Environment	AltNet	RDE	SAC	SR
Cheetah (RR=1)	<b>658.27</b>	596.62	616.12	529.94
Hopper (RR=1)	<b>248.68</b>	189.79	156.69	154.52
Quadruped (RR=1)	<b>619.12</b>	609.36	377.27	568.36
Walker (RR=1)	<b>645.76</b>	643.22	570.08	617.06
Cheetah (RR=4)	<b>721.85</b>	619.15	535.80	595.13
Hopper (RR=4)	<b>313.78</b>	278.29	205.00	248.94
Quadruped (RR=4)	703.74	<b>717.24</b>	240.93	687.43
Walker (RR=4)	<b>728.49</b>	723.64	653.82	725.44
Average (RR=1)	<b>542.96</b>	509.75	430.04	467.47
Average (RR=4)	<b>616.97</b>	584.61	408.89	565.24

is substantially lower when RR = 32. AltNet, by contrast, achieves superior performance even at RR = 4 (see Figure 4), surpassing SAC trained at much higher replay ratios. This shows that unlike SAC, AltNet achieves higher performance and greater sample efficiency at lower replay ratios, and consequently reduced computational overhead. We further establish that AltNet’s improved performance is not due to increased capacity from the additional network: reducing the total number of parameters across the two networks to match a single SAC network yields nearly identical performance (Figure 5).



**Figure 4: Learning curves of SAC in the hopper-hop environment (DMC) under different replay ratios (RR = 1, 4, 8, 32) and AltNet at RR = 4. Curves show mean episodic return over 10 seeds, with shaded regions denoting  $\pm 1$  standard error. For SAC, performance improves as RR increases up to 8, but degrades at RR = 32. AltNet achieves the highest performance at RR = 4 and is the most sample efficient.**

To quantify sample efficiency directly, we measure the *average return at fixed interaction budgets* of 100k, 300k, and 500k steps. AltNet consistently (and more rapidly, i.e., for all interaction budgets) achieves higher returns than SAC across all evaluated replay

ratios. In particular, it outperforms the best SAC baselines by 52x at 100k interactions, 1.8x at 300k, and 1.3x at 500k (Table 2). This indicates that AltNet makes more effective use of each environment interaction and achieves higher sample efficiency than SAC at any replay ratio.

**Table 2: Fixed-budget returns at 100k, 300k, and 500k environment steps. AltNet achieves higher returns at every interaction budget, indicating greater sample efficiency compared to baseline SAC at various replay ratios.**

Method	100k Steps	300k Steps	500k Steps
AltNet (RR = 4)	43.7	312.8	385.7
SAC (RR = 1)	0.14	106.8	190.7
SAC (RR = 4)	0.84	171.2	254.6
SAC (RR = 8)	0.56	79.9	294.3
SAC (RR = 32)	0.12	120.9	161.9

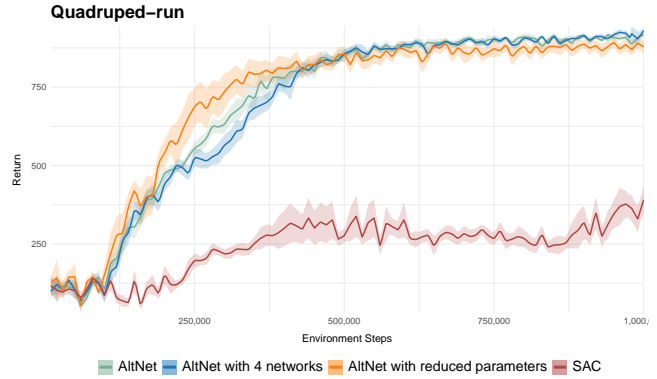
### 4.2 What Accounts for AltNet’s Success?

Having demonstrated AltNet’s advantages, we now investigate the mechanisms behind its success through five research questions (RQs). We first rule out model capacity as a primary factor and then isolate the contributions of alternating networks, resets, and replay-buffer preservation.

**RQ1. Does AltNet’s advantage arise from increased model capacity?** A natural hypothesis is that AltNet’s advantage arises simply from the increased number of trainable parameters, since it uses two networks. To test this, we reduced the size of AltNet’s networks so that the total number of trainable parameters matches that of a single-network baseline (SAC). AltNet’s performance remained largely unchanged (Figure 5, orange curve).

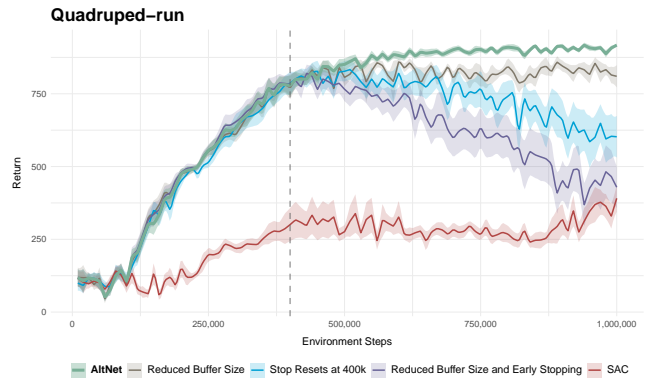
**RQ2. Does increasing the number of networks in AltNet provide additional benefits?** While AltNet’s alternating-network setup is primarily designed to anchor performance during resets, it may also encourage exploration by maintaining two policies that learn in parallel and take turns acting in the environment. This raises the question: can increasing the number of networks beyond two further improve performance? To test this, we modify AltNet to use four networks and adjust the reset frequency accordingly, as specified in Eq. (1). As shown in Figure 5 (blue curve), scaling beyond two networks does not yield additional gains. We therefore conclude that policy diversity is not a primary contributor to AltNet’s gains.

The above analyses rule out model size and the number of networks as the primary drivers of AltNet’s gains, so we shift our attention to its defining mechanisms. Specifically, AltNet’s performance when instantiated with SAC relies on two interacting processes: (i) preserving and reusing experience across resets via a shared replay buffer, and (ii) periodic alternating resets that restore network plasticity. To examine the role of each component, we perform experiments that disrupt them individually and in combination (Figure 6).



**Figure 5: We compare standard AltNet (green), a reduced-parameter version matching SAC’s parameter count (orange), and a scaled variant with four alternating networks (blue), against the SAC baseline (red). Curves report mean episodic return over 5 seeds, with shaded regions indicating  $\pm 1$  standard error. Despite reduced capacity or increased number of networks, AltNet achieves similar performance, ruling out model size and network count as drivers of its gains.**

**RQ3. Is the preservation of the replay buffer critical to AltNet’s stability?** The replay buffer plays a critical role in AltNet by providing information continuity during resets. In the experiments described so far, we have retained every agent interaction in the replay buffer. However, most off-policy algorithms do not preserve the full replay buffer indefinitely. To test the impact of buffer preservation on AltNet’s performance, we reduced the replay buffer capacity from the default 1M transitions to 400k, replacing old samples in a first-in, first-out manner. As shown in Figure 6



**Figure 6: We compare standard AltNet (green), AltNet with the replay buffer reduced to 400k transitions (gray), AltNet with resets halted after 400k steps (blue), AltNet with both interventions combined (red), and the SAC baseline (purple). Curves report mean episodic return over 10 seeds, with shaded regions indicating  $\pm 1$  standard error. Both preserving the full replay buffer and maintaining resets are essential for sustaining AltNet’s stability and performance.**

(gray curve), reducing replay buffer capacity led to a noticeable decline in performance compared to runs with the full buffer. This suggests that preserving the full replay buffer is critical for stabilizing resets and maintaining AltNet’s advantage.

**RQ4. Are alternating resets necessary once performance plateaus?** AltNet provides substantial performance gains during the early stages of training, after which progress slows as performance approaches the task’s achievable optimum. For example, in Quadruped-run, most improvements occur within the first 500k steps, after which performance stabilizes (see Figure 6). This raises a natural question: once learning has plateaued, can a single network—without alternating resets—maintain both plasticity and stability, continuing to perform well over time? To test this, we disabled resets after 400k steps while preserving the full replay buffer. As shown in Figure 6 (blue curve), performance declined once resets were discontinued. This indicates that resets remain essential even after apparent convergence, as stopping them may lead to plasticity loss and/or eventual decline in performance. Alternating network resets, by contrast, preserve both plasticity and stability, allowing AltNet to maintain high performance over time.

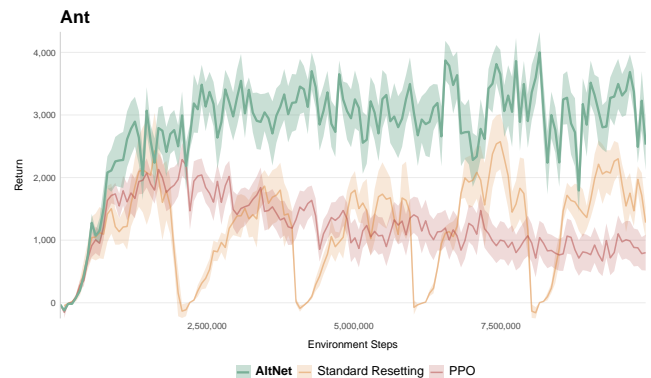
**RQ5. What happens when both mechanisms (buffer preservation and resets) are disrupted simultaneously?** Finally, we examine the combined effect of disrupting both mechanisms. If resets and buffer preservation are essential for plasticity and stability, then removing both should compound performance degradation. As expected, stopping resets while simultaneously reducing replay buffer size produced the lowest returns, aside from the SAC baseline (Figure 6, purple curve).

These findings demonstrate that AltNet’s gains are not driven by increased model capacity; it maintains its advantages even when the total number of parameters matches that of the SAC baseline. In addition, increasing the number of networks does not yield any further gains. Rather, AltNet’s improvements arise from the interplay between alternating resets and replay buffer preservation. Resets restore plasticity by mitigating failure modes such as persistently inactive neurons, rank collapse, or inflated weight norms; they reinitialize the network into a more plastic and learnable state, allowing AltNet to better exploit the increasingly representative replay buffer as training progresses. Resets also allow diverse policies to sequentially interact with the environment, which may help with exploration. Alternating networks and preserving the replay buffer maintain learning stability and continuity across resets. We find that disabling either of these components negatively impacts AltNet’s performance when paired with SAC. Next, we examine whether AltNet’s benefits extend to on-policy methods, which lack a replay buffer.

### 4.3 AltNet in On-Policy Settings

On-policy reinforcement learning is a critical test of a method’s generality and robustness. In on-policy RL, agents collect trajectories by following their current policy, and updates are performed only on newly collected samples. Agents do not have access to a replay buffer to use past data.

Existing full network reset methods, such as Standard Resets [21] and RDE [13], have been developed and evaluated primarily in off-policy settings. Juliani and Ash [12] showed that plasticity-preserving methods that are effective in off-policy settings often fail to provide benefits when paired with on-policy algorithms, where optimization dynamics differ substantially. Therefore, we now investigate whether AltNet’s benefits extend to on-policy settings, which lack a replay buffer.



**Figure 7: Performance of PPO (red curve), PPO augmented with Standard Resets (orange curve), and AltNet (green curve) in the MuJoCo Ant environment. Curves report mean episodic return over 10 seeds, except for Standard Resets, which was averaged over 5 seeds; shaded regions denote  $\pm 1$  standard error. AltNet provides consistent gains, demonstrating its efficacy in on-policy settings.**

We evaluate AltNet in MuJoCo’s Ant environment for 10M steps, a long-horizon setting where PPO suffers severe plasticity loss; in particular, after PPO reaches moderate performance, expected return declines since the agent is incapable of sustaining learning (Figure 7, red curve). In contrast, AltNet achieves nearly twice the performance of PPO and maintains it indefinitely (Figure 7, green curve). AltNet’s benefits persist due to an indirect form of knowledge transfer between its networks. While one network interacts with the environment, the other (recently reset) network learns in parallel from the same trajectories, enabling it to recover useful representations. Parallel learning anchors performance across resets and allows AltNet to maintain stability and plasticity in the on-policy regime.

We further compare AltNet with Standard Resets [21], which, without a replay buffer, suffer post-reset performance collapse (Figure 7, orange curve). Note that RDE [13] is not included in these experiments as it was not designed to tackle on-policy settings. In particular, it relies on a replay buffer and ranks actions according to their Q-values, estimated by off-policy critic. PPO, by contrast, relies on a critic that estimates state values rather than action values. AltNet thus emerges as the only full network reset method capable of restoring plasticity and achieving consistent performance gains across both off-policy and on-policy settings.

## 5 DISCUSSION

While our work focuses on mitigating plasticity loss, it is important to situate plasticity within the broader set of attributes of effective reinforcement learning agents. Ideally, RL agents should remain plastic (retaining their ability to update predictions over time) and adapt rapidly to distribution shift. They should also make full use of past data and achieve high performance with limited interactions, while maintaining stable performance. Plasticity underpins these goals: without the capacity to change, rapid adaptation and data efficiency are compromised. Viewed through this lens, AltNet addresses more than plasticity loss. It also enables rapid adaptation and efficient reuse of data, supporting sustained learning. Furthermore, by anchoring performance with a twin network, AltNet maintains stable learning dynamics and avoids performance degradation. Together, these capabilities are foundational for reinforcement learning agents that must continuously adapt over time while remaining stable and data-efficient.

## 6 LIMITATIONS AND FUTURE WORK

Although AltNet demonstrates strong empirical gains and stability across a range of continuous-control tasks, there are several directions for further study. Our experiments focus on challenging continuous-action problems from the DeepMind Control Suite, a widely used benchmark for modern reinforcement learning. Extending the evaluation to more diverse environments is likely to lead to further insights.

In addition, AltNet relies on a choice of predetermined reset frequency. An important direction for future work is to develop adaptive scheduling mechanisms that select reset frequencies automatically based on the environment and replay ratio.

## REFERENCES

- [1] Zaheer Abbas, Rosie Zhao, Joseph Modayil, Adam White, and Marlos C Machado. 2023. Loss of plasticity in continual deep reinforcement learning. In *Conference on lifelong learning agents*. PMLR, 620–636.
- [2] Alessandro Achille, Matteo Rovere, and Stefano Soatto. 2017. Critical learning periods in deep neural networks. *arXiv preprint arXiv:1711.08856* (2017).
- [3] Jordan Ash and Ryan P Adams. 2020. On warm-starting neural network training. *Advances in neural information processing systems* 33 (2020), 3884–3894.
- [4] Tudor Berariu, Wojciech Czarnecki, Soham De, Jorg Bornschein, Samuel Smith, Razvan Pascanu, and Claudia Clopath. 2021. A study on the plasticity of neural networks. *arXiv preprint arXiv:2106.00042* (2021).
- [5] Zhiyuan Chen and Bing Liu. 2018. *Lifelong machine learning*. Morgan & Claypool Publishers.
- [6] Shibhansh Dohare, J Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A Rupam Mahmood, and Richard S Sutton. 2024. Loss of plasticity in deep continual learning. *Nature* 632, 8026 (2024), 768–774.
- [7] Shibhansh Dohare, Richard S Sutton, and A Rupam Mahmood. 2021. Continual backprop: Stochastic gradient descent with persistent randomness. *arXiv preprint arXiv:2108.06325* (2021).
- [8] Pierluca D’Oro, Max Schwarzer, Evgenii Nikishin, Pierre-Luc Bacon, Marc G Bellemare, and Aaron Courville. 2022. Sample-efficient reinforcement learning by breaking the replay ratio barrier. In *Deep Reinforcement Learning Workshop NeurIPS 2022*.
- [9] William Fedus, Prajit Ramachandran, Rishabh Agarwal, Yoshua Bengio, Hugo Larochelle, Mark Rowland, and Will Dabney. 2020. Revisiting fundamentals of experience replay. In *International conference on machine learning*. PMLR, 3061–3071.
- [10] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*. Pmlr, 1861–1870.
- [11] M Igl, G Farquhar, J Luketina, W Boehmer, and S Whiteson. 2021. Transient non-stationarity and generalisation in deep reinforcement learning. In *Proceedings of the International Conference on Learning Representations*. OpenReview.
- [12] Arthur Juliani and Jordan Ash. 2024. A study of plasticity loss in on-policy deep reinforcement learning. *Advances in Neural Information Processing Systems* 37 (2024), 113884–113910.
- [13] Woojun Kim, Yongjae Shin, Jongeui Park, and Youngchul Sung. 2023. Sample-efficient and safe deep reinforcement learning via reset deep ensemble agents. *Advances in Neural Information Processing Systems* 36 (2023), 53239–53260.
- [14] Aviral Kumar, Rishabh Agarwal, Dibya Ghosh, and Sergey Levine. 2021. Implicit Under-Parameterization Inhibits Data-Efficient Deep Reinforcement Learning. In *International Conference on Learning Representations*.
- [15] Clare Lyle, Mark Rowland, and Will Dabney. 2022. Understanding and Preventing Capacity Loss in Reinforcement Learning. In *International Conference on Learning Representations*.
- [16] Clare Lyle, Zeyu Zheng, Khimya Khetarpal, Hado van Hasselt, Razvan Pascanu, James Martens, and Will Dabney. 2025. Disentangling the Causes of Plasticity Loss in Neural Networks. In *Conference on Lifelong Learning Agents*. PMLR, 750–783.
- [17] Clare Lyle, Zeyu Zheng, Evgenii Nikishin, Bernardo Avila Pires, Razvan Pascanu, and Will Dabney. 2023. Understanding plasticity in neural networks. In *International Conference on Machine Learning*. PMLR, 23190–23211.
- [18] Volodymyr Mnih et al. 2016. Asynchronous Methods for Deep Reinforcement Learning. In *Proceedings of the 33rd International Conference on Machine Learning*. 1928–1937.
- [19] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fiedelnd, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.
- [20] Evgenii Nikishin, Junhyuk Oh, Georg Ostrovski, Clare Lyle, Razvan Pascanu, Will Dabney, and André Barreto. 2023. Deep reinforcement learning with plasticity injection. *Advances in Neural Information Processing Systems* 36 (2023), 37142–37159.
- [21] Evgenii Nikishin, Max Schwarzer, Pierluca D’Oro, Pierre-Luc Bacon, and Aaron Courville. 2022. The primacy bias in deep reinforcement learning. In *International conference on machine learning*. PMLR, 16828–16847.
- [22] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).
- [23] Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. 2023. The dormant neuron phenomenon in deep reinforcement learning. In *International Conference on Machine Learning*. PMLR, 32145–32168.
- [24] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. 2018. Deepmind control suite. *arXiv preprint arXiv:1801.00690* (2018).
- [25] Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando de Freitas. 2017. Sample Efficient Actor-Critic with Experience Replay. In *International Conference on Learning Representations*.