PALR: Plasticity-Aware learning-rates for Big-World Continual Reinforcement Learning

Heng Zhang [1,2,†], **Pengsong Zhang**[3], **Qiyi Tong**[1,2], **Sheng Liu**[4], **Arash Ajoudani**[1]
heng.zhang@iit.it, pengsong.zhang@mail.utoronto.ca,
qiyi.tong@iit.it, sheng.liu@student.kit.edu, arash.ajoudani@iit.it

[1]**Human-Robot Interfaces and Interaction Lab, Istituto Italiano di Tecnologia, Genova, Italy.**
[2]**Ph.D. program of national interest in Robotics and Intelligent Machines (DRIM), Genoa, Italy.**
[3]**University of Toronto, Toronto, Canada.**
[4]**Karlsruhe Institute of Technology, Karlsruhe, Germany.**

Abstract

Loss of plasticity, the progressive inability of a neural network to adapt to new information, is a fundamental barrier to scaling reinforcement learning to "Big World" settings characterized by open-ended tasks and shifting objectives. While existing remedies like periodic resets or global perturbations can restore plasticity, they do so by indiscriminately discarding accumulated knowledge. We propose **PALR** (**P**lasticity-**A**ware **L**earning **R**ates), a lightweight augmentation that transforms plasticity maintenance into an online control problem. PALR continuously monitors two fine-grained diagnostics, the dead-neuron fraction and activation effective rank, to drive a dual-mechanism approach: (1) targeted neuron-level re-initialisation and (2) per-layer adaptive learning-rate scaling. Unlike prior meta-learning approaches, PALR employs a closed-form control loop that incurs near-zero ($< 0.1\%$) computational overhead.

We evaluate PALR across three diverse benchmarks: Continual CartPole, JellyBean-World, and ContinualWorld-10. Our results demonstrate that PALR consistently outperforms strong baselines, achieving a **+35.4%** mean reward improvement in robotic manipulation while reducing dead-neuron fractions by up to **8**$\times$. By restoring plasticity precisely where and when it degrades, PALR provides a principled mechanism for building lifelong-learning agents capable of robust, adaptive behaviour in complex, real-world environments without sacrificing the knowledge acquired over operational lifetimes.

1 Introduction

The *big world hypothesis* (Javed & Sutton, 2024) posits that real-world agents face a potentially unbounded stream of new information to learn, partial observability, limited computational resources, and no access to ground-truth environment state. A core requirement is *interactivity*: the sustained capacity to acquire new predictions across a lifetime of changing inputs (Lewandowski et al., 2026).

Modern deep RL agents are ill-suited to this regime despite their strong performance in arcade-style decision making (Mnih et al., 2015), continuous-control locomotion (Schulman et al., 2017), and specific robotic manipulation tasks (Zhang et al., 2024). Even on a single stationary task, ReLU networks gradually accumulate *dead neurons*, units whose pre-activations remain negative over the data distribution, and suffer *spectral collapse* of the Neural Tangent Kernel (NTK) Gram matrix (He et al., 2025). Under task switches, these pathologies compound: parameters tuned to an earlier distribution become obstacles, and the network's effective learning-rate falls toward zero, producing long "dead zones" immediately after each switch.

The limits of existing interventions. Standard weight-based remedies like L_2 regularisation prevent weight-norm explosion but are fundamentally unable to revive units that have already entered "dead zones" (Ash & Adams, 2020). Structural interventions, such as periodic resets (Nikishin et al., 2022) and shrink-and-perturb (Ash & Adams, 2020), can restore plasticity but often trigger catastrophic forgetting, as they apply global perturbations that indiscriminately discard valuable knowledge accumulated from prior tasks. More recent meta-learning approaches offer a principled way to adapt per-parameter step sizes (Goldie et al., 2024; Degris et al., 2024a), yet they introduce significant computational overhead due to the requirement of bi-level optimization or meta-gradients, making them difficult to scale in "big world" settings. Furthermore, while utility-based methods like Continual Backpropagation (Dohare et al., 2024) successfully replace individual dead units, they do not explicitly monitor or maintain the representation rank. Consequently, even with unit replacement, the network can still suffer from spectral collapse of the Neural Tangent Kernel (NTK) (He et al., 2025), leading to a systemic decline in the network's capacity to acquire new information.

Our approach. To address these challenges, we propose PALR, Plasticity-Aware learning-rates for continual RL. PALR takes a *surgical* approach: we re-initialise only the individual units detected as dead, leaving all other units unchanged. It intervenes only where plasticity has actually degraded, and only to the degree required. It continuously monitors the dead-neuron fraction $\delta^{(l)}$ and activation effective rank $\rho^{(l)}$ per layer, and combines them into a *plasticity deficit* that drives two coupled mechanisms: targeted dead-neuron re-initialisation and adaptive per-layer LR scaling. Our contributions are as below:

- A plug-and-play, agent-agnostic augmentation that addresses loss of plasticity without architectural modifications, employing a closed-form control loop based on explicit plasticity diagnostics (near-zero computational overhead, no bi-level meta-gradients).
- A targeted perturbation procedure that re-initialises only dead neurons (§3.3), a dead-neuron- and effective-rank-driven LR scheduler (§3.2), and a proactive task-switch hook (§3.4).
- Empirical validation on Continual CartPole (§5.1), JellyBeanWorld (§5.2), and ContinualWorld-10 (§5.3), covering discrete control, partially observable navigation, and high-dimensional robotic manipulation. An ablation confirming targeted perturbation as the primary load-bearing mechanism, with LR scaling providing complementary acceleration of post-revival weight integration (Appendix C).

2 Background and Related Work

Loss of plasticity in deep RL. The progressive decay of the ability of a neural network to adapt to new data, termed loss of plasticity, has emerged as a fundamental challenge in continual reinforcement learning (Klein et al.). While standard deep RL agents can perform well in stationary environments, they often fail under the non-stationarity inherent in lifelong learning (Javed & Sutton, 2024). Recent studies identify dead-neuron accumulation (Dohare et al., 2024) and representation collapse (Lyle et al., 2023) as the primary drivers of this failure. Mechanistically, this phenomenon is linked to the spectral collapse of the Neural Tangent Kernel (NTK) Gram matrix (He et al., 2025), which reduces the effective rank of the network and shrinks the space of learnable directions, effectively trapping the agent in a state of low trainability (Lewandowski et al., 2023).

Remedies. Existing interventions generally fall into two categories: weight-based regularisation and structural resets. Standard L2 regularisation prevents weight explosion but fails to revive units that have already entered "dead zones." Global interventions, such as shrink-and-perturb (Ash & Adams, 2020) or periodic resets (Nikishin et al., 2022), can restore plasticity but often induce catastrophic forgetting, discarding valuable knowledge accumulated from prior tasks. More recent approaches, such as continual backpropagation (Dohare et al., 2024), use utility scores to replace individual units, yet they do not explicitly account for representation rank. ReDo (Sokar et al., 2023) similarly recycles dormant neurons on a fixed schedule; we compare against both CBP and ReDo on CW10 in

§5.3. While meta-learning per-parameter step sizes offers a principled solution (Goldie et al., 2024), it introduces significant computational overhead due to bi-level optimization. In contrast, PALR closes the loop by using directly measured diagnostic signals, including the dead-neuron fraction and effective rank, to drive a closed-form learning-rate (LR) scaling rule, achieving meta-gradient-like benefits with near-zero overhead.

The big world hypothesis. The big world hypothesis (Javed & Sutton, 2024) posits that real-world agents must operate in environments that are far more complex than their function approximators (Li et al., 2023). In this regime, "lifetime tuning" is no longer feasible (Mesbahi et al., 2024), necessitating agents that can autonomously maintain their learning capacity. PALR operationalizes this requirement by monitoring the health of individual neurons and representation diversity online, ensuring that the network remains "plastic" even as the environment shifts (Yuan et al., 2025).

3 Method: Plasticity-Aware learning-rates (PALR)

In line with the big world hypothesis, PALR is a plug-and-play augmentation for any gradient-based RL agent, requiring no architectural modifications or privileged environment information. It continuously monitors the health of each hidden layer via two complementary metrics, dead-neuron fraction $\delta^{(l)}$ and activation effective rank $\rho^{(l)}$, which are combined into a *plasticity deficit* that drives adaptive per-layer LR scaling and targeted dead-neuron re-initialisation. We define the metrics in §3.1, the LR scaling rule in §3.2, the targeted perturbation in §3.3, the task-switch hook in §3.4, and the full algorithm in §3.5.

3.1 Plasticity Metrics

Let $\mathbf{a}^{(l)} \in \mathbb{R}^{B \times d_l}$ denote the post-ReLU activations of hidden layer l on a diagnostic batch of B observations.

Dead-neuron fraction. Unit j in layer l is *dead* if it outputs zero for every sample:

$$\delta^{(l)} := \frac{1}{d_l} \sum_{j=1}^{d_l} \mathbf{1}\left[\max_{i \in [B]} a_{ij}^{(l)} \leq 0\right]. \tag{1}$$

Effective rank. Following (Roy & Vetterli, 2007), we use the Shannon entropy of the normalised singular value spectrum $\{\sigma_k\}$:

$$\rho^{(l)} := \exp\left(-\sum_k p_k \log p_k\right), \qquad p_k := \frac{\sigma_k}{\sum_{k'} \sigma_{k'}}, \tag{2}$$

so $\rho^{(l)} \in [1, d_l]$. Higher effective rank indicates more diverse feature directions; this quantity is closely related to the NTK effective rank of (He et al., 2025).

Online control signal. Prior work uses effective rank as a post-hoc diagnostic (Lyle et al., 2023; 2024). Unlike prior work that uses effective rank only as a post-hoc diagnostic, we compute it online and feed it back into the optimiser. Specifically, $\rho^{(l)}$ is computed *online* every T_{measure} steps and fed directly into the optimiser, closing a *control loop* rather than serving as a passive indicator.

3.2 Plasticity Deficit and Adaptive LR Scaling

Every T_{measure} steps, PALR computes the *plasticity deficit*:

$$\phi^{(l)} := \underbrace{\left(\delta^{(l)} - \delta^*\right)_+}_{\text{dead deficit}} + \beta_\rho \underbrace{\left(\frac{\rho^* - \rho^{(l)}}{\rho^*}\right)_+}_{\text{rank deficit}}, \tag{3}$$

where $\delta^* = 0$ and $\rho^* = d_l/2$. The per-layer LR scale factor is:

$$s^{(l)} = \mathrm{clip}\Big(1 + \beta\,\phi^{(l)},\ 1,\ s_{\max}\Big)\,, \tag{4}$$

with $s_{\max} = 5.0$. Gradients for layer l are multiplied by $s^{(l)}$ before the Adam update. This approximates a one-step gradient-ascent update on the meta-objective $J(s^{(l)}) := \mathbb{E}[\rho^{(l)}(s^{(l)})]$, recovering the effect of bi-level step-size meta-learning (Degris et al., 2024b) at the cost of a single forward pass.

3.3 Targeted Neuron-Level Perturbation

When $\delta^{(l)} > \tau_{\text{perturb}}$, PALR re-initialises *only* the incoming weights and biases of dead units:

$$\mathbf{W}^{(l)}_{:,j} \leftarrow \mathcal{N}\left(0,\ \frac{2\,\sigma_p^2}{d_{l-1}}\right), \quad b^{(l)}_j \leftarrow 0, \quad \forall\, j : \delta^{(l)}_j = 1. \tag{5}$$

The bias reset is critical: a strongly negative bias keeps a unit dead even after weight re-initialisation. Healthy neurons are left unchanged, preserving consolidated representations.

Mechanistic necessity. LR scaling alone cannot revive dead neurons: for a ReLU unit with pre-activation $z_j \leq 0$, the gradient is zero, so $s^{(l)} \cdot 0 = 0$ regardless of scale. Targeted perturbation restores the gradient pathway; the LR boost maximises weight adjustment during each subsequent active window. LR scaling cannot revive a dead unit because its gradient is zero; re-initialisation restores a nonzero gradient, which is confirmed by ablation in all three settings.

3.4 Proactive Task-Switch Response

At each detected task switch, PALR intervenes *before* the first gradient update on the new task: it applies targeted perturbations across all layers and enforces a proactive learning–rate floor,

$$s^{(l)} \leftarrow \max\big(s^{(l)}, 2.0\big), \tag{6}$$

for T_{measure} steps. This removes the "dead zone" in which reactive methods remain idle until the next scheduled plasticity measurement, ensuring that every layer begins the new task with sufficient plasticity.

3.5 Full Algorithm

Algorithm 1 PALR Training Step (agent-agnostic)

Require: Network f_θ; base LR η, deficit sensitivity β, rank weight β_ρ, measure interval T_{measure}, perturb threshold τ, perturb scale σ_p, scale ceiling $s_{\max}$

Initialise $s^{(l)} \leftarrow 1$ for all hidden layers l
for each environment step t **do**
  Collect transition; push to replay buffer $\mathcal{D}$
  if task switch detected at t **then**
    Apply targeted perturbation (Eq. 5)
    Apply LR floor (Eq. 6)
  end if
  if $t \bmod T_{\text{measure}} = 0$ **and** $|\mathcal{D}| \geq B$ **then**
    Sample 256-transition diagnostic batch (no grad)
    Compute $\delta^{(l)}, \rho^{(l)}$ per layer (Eqs. 1–2)
    Compute $\phi^{(l)}$, update $s^{(l)}$ (Eqs. 3–4)
    for each l with $\delta^{(l)} > \tau$ **do**
      Apply targeted perturbation (Eq. 5)
    end for
  end if
  if $|\mathcal{D}| \geq$ batch size **then**
    Compute RL loss; compute gradients $\mathbf{g}$
    Scale: $\mathbf{g}^{(l)} \leftarrow s^{(l)} \cdot \mathbf{g}^{(l)}$; apply via Adam
  end if
end for

4 Experiments

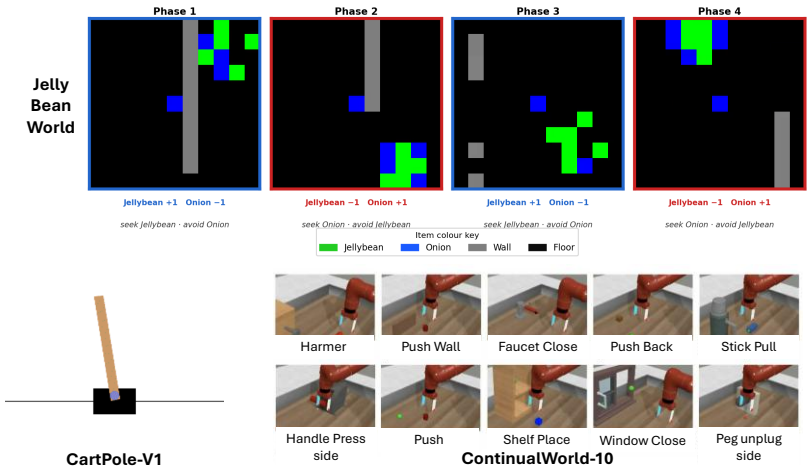

Figure 1: Tasks in the three benchmarks.

To evaluate PALR, we design experiments on three benchmarks that stress different aspects of continual RL in big worlds, covering a range of environment complexity, non-stationarity types, and reward structures. These benchmark experiments are intentionally designed from simpler to more complex settings, allowing us to trace the benefits of PALR across a spectrum of challenges and to identify the specific conditions under which it provides the greatest advantage. The tasks in these three benchmarks are shown in Figure 1.

Table 1: Dynamics parameters for the four-task Continual CartPole curriculum. Tasks switch every 200 episodes and repeat indefinitely in this order.

Task	**Gravity**	**Pole Mass**	**Pole Length**
Default	9.8	0.10	0.50
Heavy-Slow	14.0	0.20	0.70
Light-Fast	7.0	0.05	0.30
Default-Heavy	9.8	0.15	0.60

Table 2: JBW plasticity metrics (5-seed mean, averaged over all measurement steps).

Agent	Dead (L2) ↓	Eff. rank (L2) ↑
PALR **(ours)**	**0.056**	**63.8**
DQN-FixedLR	0.470	39.7
PALR-NoPerturb	0.503	36.9

4.1 Benchmark 1: Continual CartPole

We employ a modified CartPole-v1 environment (Brockman et al., 2016) comprising 10 sequential task phases, each lasting 200 episodes (2,000 in total). These phases cycle through four distinct physics configurations, varying gravity, pole mass, and pole length, such that phases 5–8 and 9–10 revisit the same dynamics as phases 1–4. This design induces repeated exposure to previously encountered regimes, thereby stressing long-horizon plasticity across multiple cycles. As summarized in Table 5, the Continual CartPole benchmark transitions from the standard CartPole dynamics to a heavier and slower variant, then to a lighter and faster configuration, and finally to a "Default-Heavy" regime that maintains nominal gravity while increasing pole mass and length. The environment switches tasks every 200 episodes and repeats this four-regime curriculum indefinitely, yielding a controlled yet non-stationary setting for evaluating continual adaptation.

Agent. PALR augments a standard DQN backbone with a two-hidden-layer MLP Q-network (128×128 units, ReLU activations), optimized using Adam with learning-rate $\eta = 10^{-3}$. We employ a replay buffer of size 50,000 and a minibatch size of 64.

Baselines. We compare against DQN-FixedLR, ShrinkAndPerturb, PeriodicReset, L2-Reg, PALR-NoPerturb, and PALR-NoScale. All agents are trained with 10 seeds and 200 episodes per phase.

Metrics. We report (i) overall mean reward across all phases (primary), (ii) final-50 reward as a measure of asymptotic performance within each phase, and (iii) recovery speed, defined as the number of episodes required to reach a reward of at least 200 after each task switch (lower is better).

4.2 Benchmark 2: JellyBeanWorld

JellyBeanWorld (JBW) (Platanios et al., 2020) is designed explicitly for big-world continual RL and never-ending learning. It simultaneously presents: (i) *partial observability*, where the agent receives only an 11×11 local visual patch together with a 3-dimensional scent vector; (ii) *infinite horizon*, with no natural episode boundaries; and (iii) *unannounced non-stationarity*, in which the reward polarity flips every 100 episodes without any external signal. This setting is strictly more challenging than Continual CartPole in 4.1, as the reward function itself changes, requiring the agent to detect and invert its policy purely from reward feedback.

Agent. We use the same DQN-based PALR architecture as in CartPole. Each agent is trained for 5 random seeds over 400 episodes (400,000 environment steps). At this scale, differences in cumulative reward are not yet statistically meaningful. Because full convergence in JBW typically requires 10^6–10^7 steps, therefore plasticity metrics constitute the primary evaluation signal.

4.3 Benchmark 3: ContinualWorld-10 (CW10)

ContinualWorld-10 (Wołczyk et al., 2021) is a sequential benchmark composed of 10 MetaWorld (Yu et al., 2020) robotic manipulation tasks executed back-to-back without a task oracle. The tasks span a diverse set of manipulation skills (Reach, Push, Pick-Place, Door-Open, Drawer-Open, Drawer-Close, Button-Press, Peg-Insert, Window-Open, Window-Close), each run for 200 episodes (2,000 total).

This benchmark stresses both plasticity maintenance and knowledge retention in a high-dimensional continuous-control setting.

Agent. PALR augments the SAC critic with a four-layer MLP (256×4 units, ReLU activations), optimized using Adam with learning rate $\eta = 3\times10^{-4}$. We use a replay buffer of size 500,000 and a minibatch size of 2,048. Baselines include SAC-FixedLR, SAC-L2Reg, PALR-NoPerturb, and PALR-NoScale. All agents are trained with 5 random seeds. We add two continual-learning baselines from the re-initialisation family: **CBP** (Continual Backpropagation, Dohare et al., 2024), which continually replaces the lowest-utility mature units at a fixed rate, and **ReDo** (Sokar et al., 2023), which recycles dormant neurons on a fixed schedule. We further add three PALR variants that isolate individual design choices: PALR-BetaOn (LR scaling enabled on CW10, $\beta = 3.0$), PALR-NoOracle (the proactive task-switch hook of §3.4 disabled entirely), and PALR-AutoOracle (the oracle replaced by a self-computed reward-drop detector: an intervention is triggered when the mean reward of the last 10 episodes falls below $0.5\times$ the mean of the preceding 30 episodes, with a 100-episode cooldown). Hyperparameters are listed in Table 7.

Metrics. We report overall mean reward (primary), early-task reward (tasks 1–5), late-task reward (tasks 6–10), adaptation speed (mean reward over the first 50 episodes of each task), and per-layer plasticity metrics.

5 Results

5.1 Benchmark 1: Continual CartPole

Table 3: Continual CartPole results (2 000 episodes, 10 phases, mean across 10 seeds). Recovery: episodes to reward ≥ 200 after a task switch (lower is better). L2-Dead: mean Layer-2 dead-neuron fraction (%) across all episodes (lower is better). Eff-Rank: mean effective rank at Phase-10 (higher is better). †PeriodicReset's low L2-Dead results from full-network resets, not maintained plasticity.

Agent	Overall ↑	Final-50 ↑	Phase-10 ↑	Recovery (ep) ↓	L2-Dead (%) ↓	Eff-Rank ↑
PALR **(ours)**	**319.2**	363.8	**362.0**	3.9 ± 11.9	7.4	12.32
DQN-FixedLR	303.5	294.7	285.6	6.5 ± 17.7	39.9	10.31
L2-Reg	309.9	316.2	318.3	5.6 ± 15.0	32.9	11.65
ShrinkAndPerturb	179.0	166.6	171.9	8.1 ± 15.8	23.5	6.46
PeriodicReset	38.0	32.7	33.7	91.7 ± 24.6	**4.9**†	7.83
PALR-NoPerturb	311.4	278.7	315.9	$\mathbf{3.6 \pm 7.9}$	39.1	11.61
PALR-NoScale	319.1	**385.0**	349.9	2.5 ± 7.2	7.3	**13.66**

Overall performance. PALR achieves the highest overall mean reward (319.2), surpassing DQN-FixedLR (303.5, +5.2%) and L2-Reg (309.9) across the full 10-phase curriculum. Critically, this advantage grows with time: **23%** higher final-50 reward (363.8 vs. 294.7 for DQN) and **27%** higher Phase-10 reward (362.0 vs. 285.6). DQN degrades on repeated dynamics, Phase-2 reward 339.5 falls to 285.6 by Phase-10 (same Heavy-Slow dynamics, seen three times), while PALR holds steady and improves (331.7 → 353.3 → 362.0), reflecting the compounding benefit of maintained plasticity.

Recovery speed. PALR recovers in 3.9 episodes on average, **40%** faster than DQN (6.5). PeriodicReset almost never recovers within 100 episodes (mean 91.7), confirming that whole-network resets destroy the replay-buffer knowledge needed for fast re-adaptation.

Plasticity dynamics. Without targeted perturbation (PALR-NoPerturb), Layer-2 accumulates 39.1% dead neurons, matching DQN (39.9%), while full PALR holds at 7.4% (**5× lower**). Effective rank grows from 9.51 in Phase-1 to 12.32 in Phase-10 for full PALR, confirming that maintained plasticity compounds over repeated cycles.

5.2 Benchmark 2: JellyBeanWorld

PALR maintains a Layer-2 dead-neuron fraction of only **5.6%** in JBW (Table 2), compared to **47.0%** for DQN-FixedLR (**8× reduction**) and **50.3%** for PALR-NoPerturb (LR scaling alone). Effective rank follows the same pattern: PALR achieves 63.8 vs. 39.7 for DQN (+61%) and 36.9 for PALR-NoPerturb.

These results replicate the CartPole plasticity findings in a strictly harder setting with partial observability, no task-switch oracle, and reward-function non-stationarity. The perturbation mechanism remains the critical component: without it, Layer-2 collapses to the same dead-neuron level as DQN, confirming that LR scaling alone cannot revive already-dead units regardless of environment complexity. Full plasticity dynamics over training are shown in Figure 5 (Appendix B).

5.3 Benchmark 3: ContinualWorld-10 (CW10)

Table 4: CW10 results (2 000 episodes, 10 tasks, mean across up to 5 seeds). Bold: best per column among the original submission agents. Adapt speed: mean reward over the first 50 episodes of each task. T indicates task. CBP and ReDo baselines (upper block) and PALR variants isolating the LR-scaling and task-switch-hook design choices (lower block).

Agent	Overall ↑	Early (T1–5) ↑	Late (T6–10) ↑	Adapt speed ↑
PALR **(ours)**	**583.7**	**729.2**	**438.2**	518.8
PALR-NoScale	555.3	703.2	407.5	**650.9**
CBP-SAC	487.9	558.5	417.3	526.4
SAC-L2Reg	440.5	545.8	335.2	463.8
SAC-FixedLR	431.2	491.7	370.7	391.9
ReDo-SAC	386.3	476.7	296.0	472.1
PALR-NoPerturb	334.5	395.8	273.2	464.9
PALR-AutoOracle	561.4	625.2	497.5	501.3
PALR-BetaOn (β=3)	556.8	605.7	507.9	511.7
PALR-NoOracle	395.2	576.7	213.6	389.7

Overall performance. PALR achieves the highest overall mean reward (583.7), surpassing SAC-FixedLR by **+35.4%** (431.2) and SAC-L2Reg by **+32.5%** (440.5). The advantage is consistent across early tasks (729.2 vs. 491.7, +48.4%) and late tasks (438.2 vs. 370.7, +18.2%), indicating that plasticity maintenance provides sustained benefit throughout the curriculum.

Adaptation speed. PALR (518.8) and PALR-NoScale (650.9) substantially outperform SAC-FixedLR (391.9) on adapt speed, confirming that dead-neuron revival accelerates recovery after each switch.

Plasticity dynamics. PALR maintains all four critic layers below 4% dead throughout training (**8–10× lower** than SAC-FixedLR in deep layers; Figure 6 in Appendix B). The scatter plot in Figure 7 (Appendix B) reveals a strong negative correlation between mean Layer-4 dead-neuron fraction and mean reward across all agents, directly linking the mechanism to performance differences.

Comparison with re-initialisation baselines. CBP-SAC reaches an overall mean reward of 487.9 (+13.2% over SAC-FixedLR), confirming that utility-based unit replacement partially preserves plasticity on CW10, but it remains below PALR (583.7; Figure 3). ReDo-SAC (386.3) falls below the fixed-LR baseline: recycling dormant units on a fixed 1,000-step schedule repeatedly perturbs a critic that is still integrating the current task. Both methods re-initialise units according to a fixed rate or schedule, whereas PALR intervenes only when the measured per-layer dead fraction exceeds a threshold and additionally intervenes at task boundaries; the comparison suggests that *when* units are re-initialised matters as much as *which* units are.

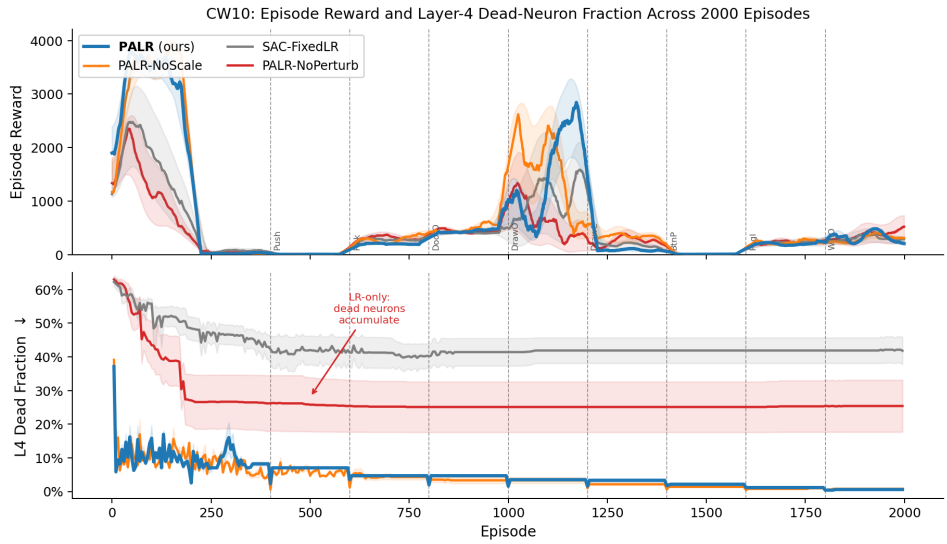

Figure 2: **CW10 hero: episode reward (top) and Layer-4 dead-neuron fraction (bottom), time-aligned over 2 000 episodes.** Dashed vertical lines mark task switches; shaded bands show ± 0.5 std. PALR (blue) leads in reward throughout and holds dead fraction near zero. PALR-NoPerturb (red dashed) accumulates dead neurons at the same rate as SAC-FixedLR, directly explaining its reward collapse: scaling a zero gradient achieves nothing ($s^{(l)} \cdot 0 = 0$).

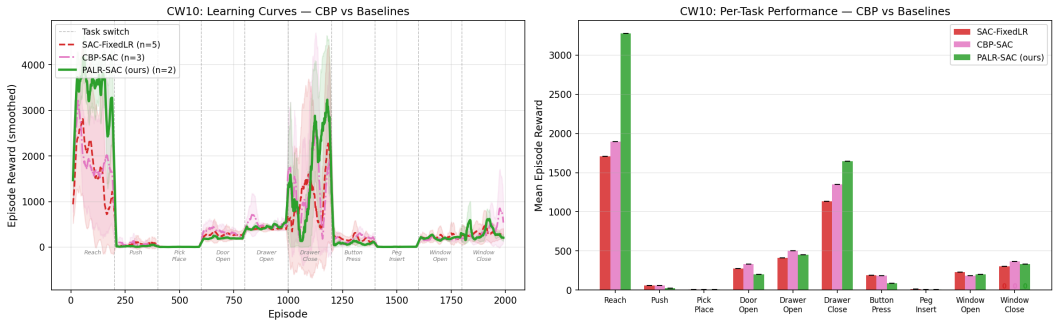

Figure 3: **CW10 comparison with CBP (Dohare et al., 2024).** Left: smoothed learning curves with task switches marked (dashed lines). Right: per-task mean episode reward. CBP-SAC improves over SAC-FixedLR (487.9 vs. 431.2 overall) but does not match PALR, with the largest gaps on Reach (T1) and Drawer Close (T6).

Effect of LR scaling on CW10 ($\beta > 0$). Our CW10 configuration sets the deficit sensitivity $\beta = 0$, disabling LR scaling (Table 7). To make this an evidence-based choice rather than an unexplained inconsistency, we ran PALR-BetaOn, identical to full PALR but with $\beta = 3.0$ (the value used on Continual CartPole). It attains 556.8, within 5% of the $\beta = 0$ configuration (583.7) and well above all non-PALR baselines: on CW10, LR scaling neither helps nor is needed, so we keep it disabled there. The mechanism is benchmark-dependent rather than vestigial: on Continual CartPole ($\beta = 3.0$), scaling contributes to the best sustained reward (Table 3).

Isolating the task-switch signal. Only the proactive hook of §3.4 uses task-switch information; the measurement-driven re-initialisation and LR scaling run without any oracle. To quantify how much of the CW10 gain depends on this assumption, we ran PALR-NoOracle (hook disabled): performance drops to 395.2, showing that on CW10's abrupt task switches the proactive intervention carries a large share of the benefit. Crucially, the oracle can be replaced by a signal the agent computes itself:

PALR-AutoOracle detects switches from a reward-drop heuristic and reaches 561.4, recovering 96% of the full method's performance with no privileged task information. Automatic switch detection from the plasticity signals themselves remains future work.

Overall, on Continual CartPole, PALR achieves a 23% improvement in final-50 reward, a $5\times$ reduction in Layer-2 dead-neuron fraction, and a 40% faster recovery rate compared to DQN-FixedLR over a 10-phase, 2,000-episode curriculum. On JellyBeanWorld, it yields an $8\times$ reduction in Layer-2 dead-neuron fraction and a 61% increase in effective rank under partial observability and unannounced reward-function shifts. On ContinualWorld-10, PALR improves overall mean reward by 35.4% relative to SAC-FixedLR while maintaining 8–$10\times$ lower deep-layer dead-neuron fractions across ten sequential manipulation tasks, and outperforms re-initialisation baselines CBP (487.9) and ReDo (386.3). A mechanism-hierarchy ablation (targeted perturbation as the primary driver, LR scaling as a complementary contribution, and neuron-redeath dynamics motivating continuous monitoring) is provided in Appendix C. For a component-wise decomposition of these gains and the underlying revival-productivity dynamics, we refer the reader to Appendix C.

6 Conclusion

We introduced PALR, a lightweight and plug-in plasticity-aware method for continual reinforcement learning. By monitoring dead-neuron fraction and activation effective rank on a per-layer basis, and by combining adaptive learning-rate scaling with targeted neuron-level perturbation, PALR consistently outperforms strong baselines across three qualitatively distinct benchmarks.

Three design principles underpin PALR's effectiveness. First, effective rank serves as an *online control signal* that closes a feedback loop between representational collapse and the optimiser. Second, the learning-rate scaling rule provides a closed-form meta-gradient approximation for maximising NTK rank, enabling principled and computationally efficient adaptation. Third, interventions operate at the *granularity of individual neurons*, matching the locus of plasticity failure without disrupting healthy representations. Notably, PALR requires no architectural modifications, no privileged environment information, and no network resets, making it a practical and easily deployable component for continual-learning agents. These properties are essential for scaling reinforcement learning to "big-world" settings: domains characterized by open-ended tasks, shifting objectives, and extended operational lifetimes. As such, PALR provides a principled and general mechanism for building lifelong-learning agents capable of robust, adaptive behaviour in complex real-world environments.

References

David Abel, André Barreto, Benjamin Van Roy, Doina Precup, Hado van Hasselt, and Satinder Singh. A definition of continual reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

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

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. *arXiv preprint arXiv:1606.01540*, 2016.

Thomas Degris, Khurram Javed, Arsalan Sharifnassab, Yuxin Liu, and Richard Sutton. Step-size optimization for continual learning. *arXiv preprint arXiv:2401.17401*, 2024a.

Thomas Degris, Khurram Javed, Arsalan Sharifnassab, Yuxin Liu, and Richard Sutton. Step-size optimization for continual learning. *arXiv preprint arXiv:2401.17401*, 2024b.

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.

Alexander D Goldie, Chris Lu, Matthew T Jackson, Shimon Whiteson, and Jakob N Foerster. Can learned optimization make reinforcement learning less difficult? *Advances in Neural Information Processing Systems*, 37:5454–5497, 2024.

Naicheng He, Kaicheng Guo, Arjun Prakash, Saket Tiwari, Ruo Yu Tao, Tyrone Serapio, Amy Greenwald, and George Konidaris. Spectral collapse drives loss of plasticity in deep continual learning. *arXiv preprint arXiv:2509.22335*, 2025.

Maximilian Igl, Gregory Farquhar, Jelena Luketina, Wendelin Boehmer, and Shimon Whiteson. Transient non-stationarity and generalisation in deep reinforcement learning. In *International Conference on Learning Representations*, 2021.

Khurram Javed and Richard S. Sutton. The big world hypothesis. In *Proceedings of the Workshop on Continual Reinforcement Learning*, 2024.

Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. Towards continual reinforcement learning: A review and perspectives. *Journal of Artificial Intelligence Research*, 75:1401–1476, 2022.

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

Alex Lewandowski, Haruto Tanaka, Dale Schuurmans, and Marlos C Machado. Directions of curvature as an explanation for loss of plasticity. *arXiv preprint arXiv:2312.00246*, 2023.

Alex Lewandowski, Aditya Ramesh, Edan Meyer, Dale Schuurmans, and Marlos C Machado. The world is bigger! a computationally-embedded perspective on the big world hypothesis. *Advances in Neural Information Processing Systems*, 38:28210–28237, 2026.

Bojiang Li, Wen Chen, Zhendong Li, Qingqing Wu, Nan Cheng, Changle Li, and Linglong Dai. Robust weighted sum-rate maximization for transmissive ris transmitter enabled rsma networks. *IEEE Communications Letters*, 27(10):2847–2851, 2023.

Lu Lu, Yeonjong Shin, Yanhui Su, and George Em Karniadakis. Dying ReLU and initialization: Theory and numerical examples. *Communications in Computational Physics*, 28:1671–1706, 2020.

Clare Lyle, Zeyu Zheng, Evgenii Nikishin, Bernardo Avila Pires, Razvan Pascanu, and Marc G. Bellemare. Understanding plasticity in neural networks. In *Proceedings of the 40th International Conference on Machine Learning*, ICML, 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.

Golnaz Mesbahi, Parham Mohammad Panahi, Olya Mastikhina, Steven Tang, Martha White, and Adam White. Position: Lifetime tuning is incompatible with continual reinforcement learning. *arXiv preprint arXiv:2404.02113*, 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.

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron Courville. The primacy bias in deep reinforcement learning. In *Proceedings of the 39th International Conference on Machine Learning*, ICML, 2022.

Emmanouil Antonios Platanios, Abulhair Saparov, and Tom Mitchell. Jelly bean world: A testbed for never-ending learning. *arXiv preprint arXiv:2002.06306*, 2020.

Mark B. Ring. *Continual Learning in Reinforcement Environments*. PhD thesis, University of Texas at Austin, 1994.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. *Proceedings of the 15th European Signal Processing Conference (EUSIPCO)*, 2007.

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 (ICML)*, 2023.

Maciej Wołczyk, Michał Zając, Razvan Pascanu, Łukasz Kuciński, and Piotr Miłoś. Continual world: A robotic benchmark for continual reinforcement learning. *Advances in Neural Information Processing Systems*, 34:28496–28510, 2021.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020.

Mingqi Yuan, Qi Wang, Guozheng Ma, Caihao Sun, Bo Li, Xin Jin, Yunbo Wang, Xiaokang Yang, Wenjun Zeng, Dacheng Tao, et al. Plasticine: Accelerating research in plasticity-motivated deep reinforcement learning. *arXiv preprint arXiv:2504.17490*, 2025.

Heng Zhang, Gokhan Solak, Gustavo J. G. Lahr, and Arash Ajoudani. Srl-vic: A variable stiffness-based safe reinforcement learning for contact-rich robotic tasks. *IEEE Robotics and Automation Letters*, 9(6):5631–5638, 2024. DOI: 10.1109/LRA.2024.3396368.

Supplementary Materials

The following content was not necessarily subject to peer review.

Appendix

A Extended Benchmark Setup

This appendix provides the detailed benchmark descriptions that were summarised in §4.

A.1 Benchmark 1: Continual CartPole (full description)

We employ a modified CartPole-v1 environment (Brockman et al., 2016) comprising 10 sequential task phases, each lasting 200 episodes (2,000 in total). These phases cycle through four distinct physics configurations, varying gravity, pole mass, and pole length, such that phases 5–8 and 9–10 revisit the same dynamics as phases 1–4. This design induces repeated exposure to previously encountered regimes, thereby stressing long-horizon plasticity across multiple cycles. As summarized in Table 5, the Continual CartPole benchmark transitions from the standard CartPole dynamics to a heavier and slower variant, then to a lighter and faster configuration, and finally to a "Default-Heavy" regime that maintains nominal gravity while increasing pole mass and length. The environment switches tasks every 200 episodes and repeats this four-regime curriculum indefinitely, yielding a controlled yet non-stationary setting for evaluating continual adaptation.

Table 5: Dynamics parameters for the four-task Continual CartPole curriculum. Tasks switch every 200 episodes and repeat indefinitely in this order.

Task	Gravity	Pole Mass	Pole Length
Default	9.8	0.10	0.50
Heavy-Slow	14.0	0.20	0.70
Light-Fast	7.0	0.05	0.30
Default-Heavy	9.8	0.15	0.60

Agent. PALR augments a standard DQN backbone with a two-hidden-layer MLP Q-network (128×128 units, ReLU activations), optimized using Adam with learning-rate $\eta = 10^{-3}$. We employ a replay buffer of size 50,000 and a minibatch size of 64.

Baselines. We compare against DQN-FixedLR, ShrinkAndPerturb, PeriodicReset, L2-Reg, PALR-NoPerturb, and PALR-NoScale. All agents are trained with 10 random seeds and 200 episodes per phase.

Metrics. We report (i) overall mean reward across all phases (primary), (ii) final-50 reward as a measure of asymptotic performance within each phase, and (iii) recovery speed, defined as the number of episodes required to reach a reward of at least 200 after each task switch (lower is better).

A.2 Benchmark 2: JellyBeanWorld (full description)

JellyBeanWorld (JBW) (Platanios et al., 2020) is designed explicitly for big-world continual RL and never-ending learning. It simultaneously presents: (i) *partial observability*, where the agent receives only an 11×11 local visual patch together with a 3-dimensional scent vector; (ii) *infinite horizon*, with no natural episode boundaries; and (iii) *unannounced non-stationarity*, in which the reward polarity flips every 100 episodes without any external signal. This setting is strictly more challenging than Continual CartPole, as the reward function itself changes, requiring the agent to detect and invert its policy purely from reward feedback.

Agent. We use the same DQN-based PALR architecture as in CartPole. Each agent is trained for 5 random seeds over 400 episodes (400,000 environment steps). At this scale, differences in cumulative reward are not yet statistically meaningful, full convergence in JBW typically requires 10^6–10^7 steps, therefore plasticity metrics constitute the primary evaluation signal.

A.3 Benchmark 3: ContinualWorld-10 (full description)

ContinualWorld-10 (Wołczyk et al., 2021) is a sequential benchmark composed of 10 MetaWorld (Yu et al., 2020) robotic manipulation tasks executed back-to-back without a task oracle. The tasks span a diverse set of manipulation skills (Reach, Push, Pick-Place, Door-Open, Drawer-Open, Drawer-Close, Button-Press, Peg-Insert, Window-Open, Window-Close), each run for 200 episodes (2,000 total). This benchmark stresses both plasticity maintenance and knowledge retention in a high-dimensional continuous-control setting.

Agent. PALR augments the SAC critic with a four-layer MLP (256×4 units, ReLU activations), optimized using Adam with learning-rate $\eta = 3\times10^{-4}$. We use a replay buffer of size 500,000 and a minibatch size of 2,048. Baselines include SAC-FixedLR, SAC-L2Reg, PALR-NoPerturb, and PALR-NoScale, CBP (Dohare et al., 2024), ReDo (Sokar et al., 2023), PALR-BetaOn, PALR-NoOracle, and PALR-AutoOracle (see §4.3). All agents are trained with 5 random seeds.

Metrics. We report overall mean reward (primary), early-task reward (tasks 1–5), late-task reward (tasks 6–10), adaptation speed (mean reward over the first 50 episodes of each task), and per-layer plasticity metrics.

B Supplementary Figures

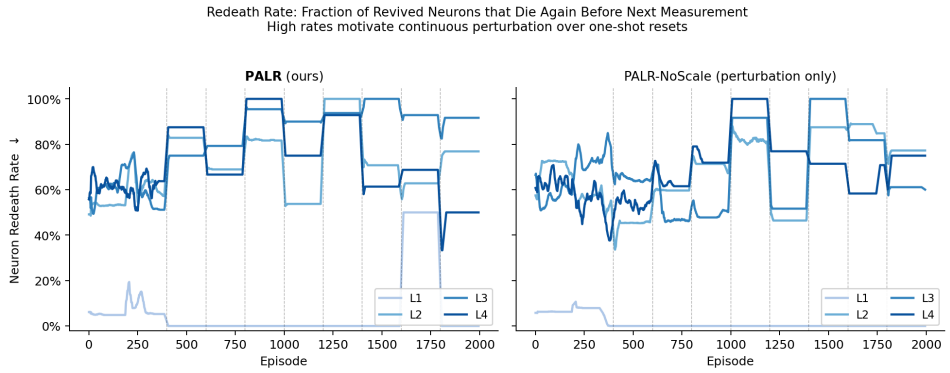

Figure 4: Re-death rate in CW10 deep layers (Layers 3–4) for PALR-SAC and PALR-SAC-NoScale. Both deep layers show persistently high re-death rates (0.68–0.84 mean), indicating structural re-collapse drives the need for continuous monitoring.

C Ablation: Mechanism Hierarchy Across Benchmarks

The ablation results across all three benchmarks establish a consistent hierarchy:

Targeted perturbation is the primary mechanism. Across all benchmarks, PALR-NoPerturb (LR scaling only) either performs below the fixed-LR baseline (CW10: 334.5 vs. 431.2) or accumulates dead neurons at the same rate as the baseline (JBW: 50.3% vs. 47.0%). This is mechanistically explained: LR scaling multiplies a zero gradient, achieving nothing for dead units regardless of scale factor.

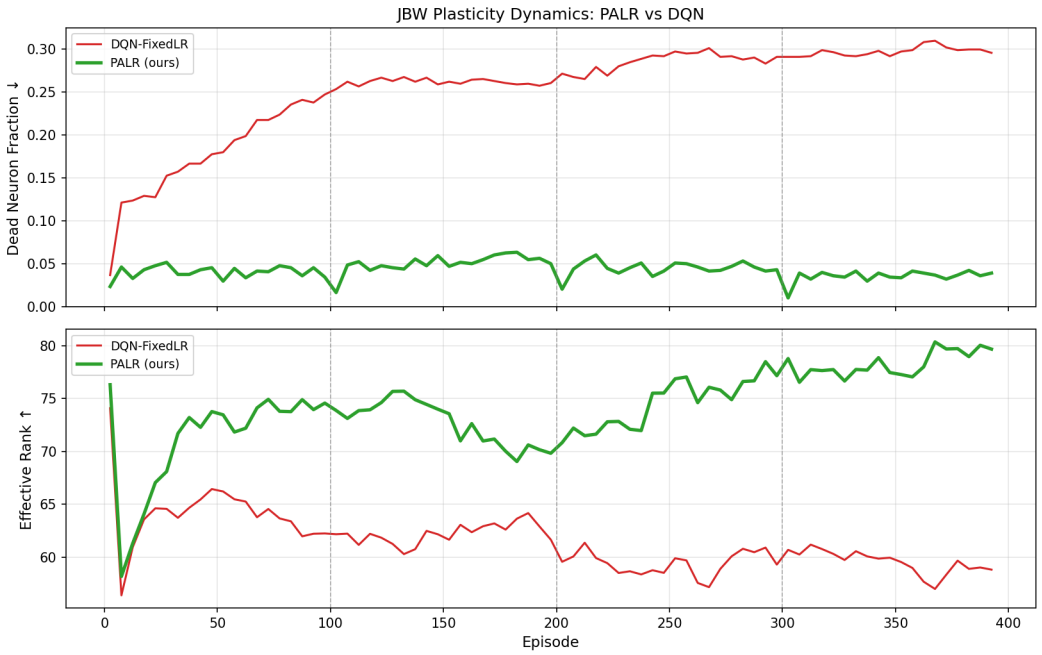

Figure 5: **JBW plasticity dynamics (dead-neuron fraction and effective rank).** PALR (blue) maintains low dead fraction throughout, even under unannounced reward-polarity flips (dashed lines). PALR-NoPerturb (red) converges to the same dead-fraction level as DQN-FixedLR (grey), confirming LR scaling alone cannot revive dead units.

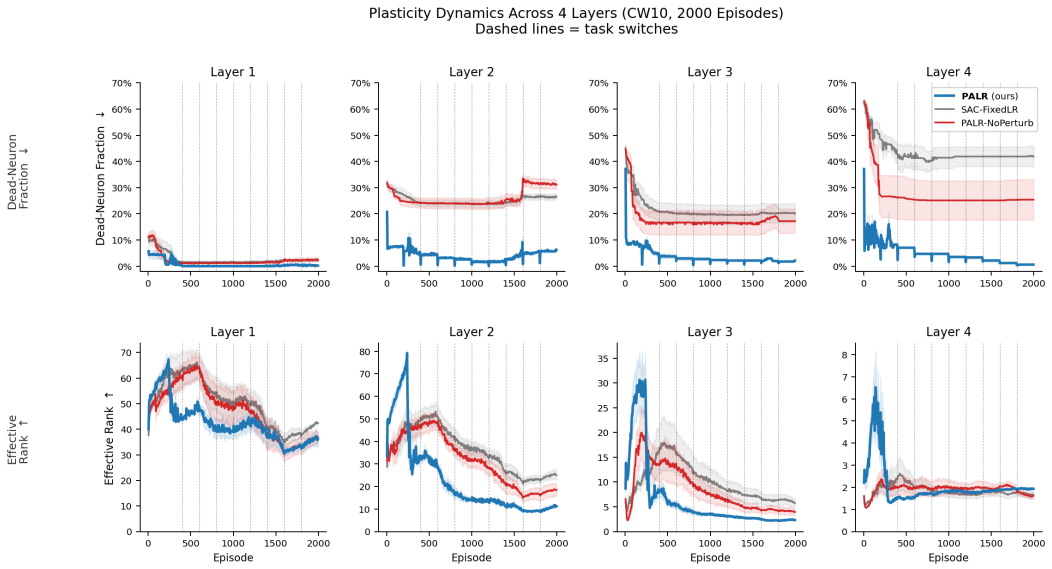

Figure 6: CW10 plasticity dynamics across all 4 critic layers. Top row: dead-neuron fraction (lower is better). Bottom row: activation effective rank (higher is better). PALR (blue) and PALR-NoScale maintain near-zero dead fractions. PALR-NoPerturb (red) matches SAC-FixedLR in dead-neuron accumulation, confirming LR scaling alone cannot revive dead units.

LR scaling provides a complementary but secondary contribution. PALR-NoScale (perturbation only) achieves 28.8% gain on CW10 and near-zero dead fractions on CartPole, but full PALR adds a further 5.1% on CW10 and the highest sustained overall reward on CartPole. The LR boost

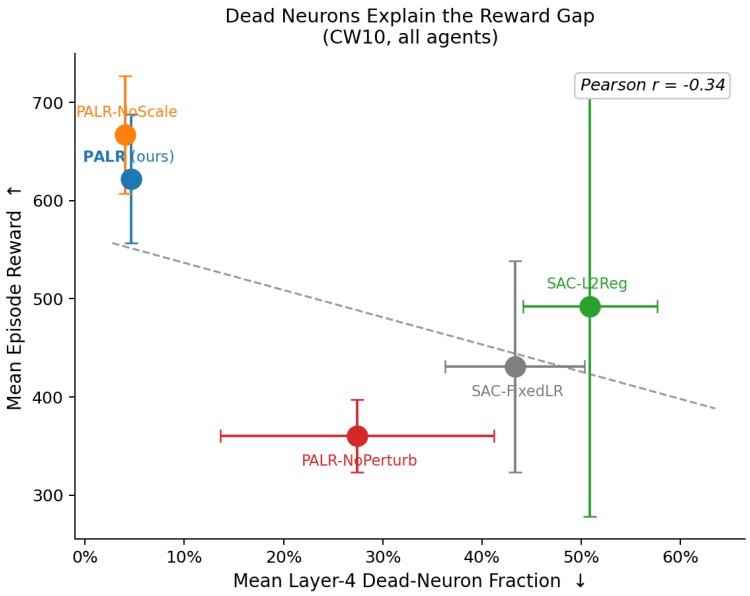

Figure 7: **Dead neurons explain the reward gap (CW10).** Each point is one agent; x: mean Layer-4 dead-neuron fraction; y: mean episode reward. The strong negative correlation (Pearson r shown) confirms dead-neuron accumulation as the primary driver of performance differences.

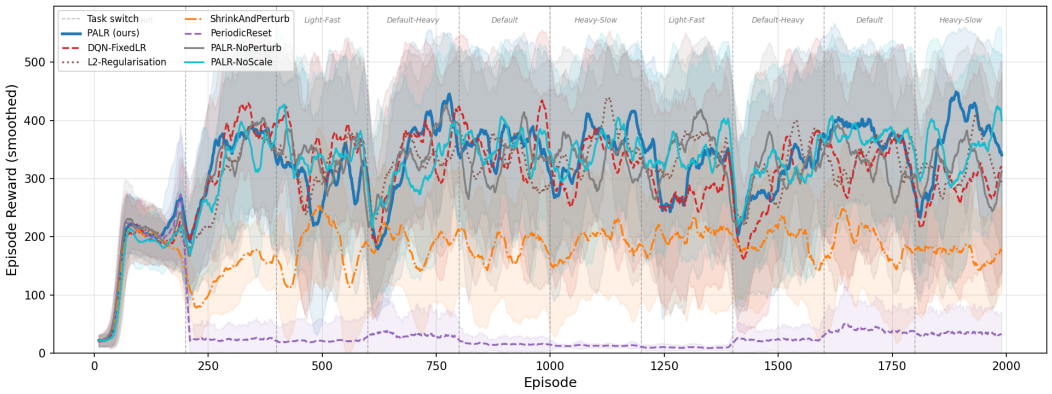

Figure 8: **Continual CartPole learning curves, all agents.** Smoothed per-episode reward across the 10-phase curriculum (10 seeds); dashed vertical lines mark task switches. PALR sustains and improves reward on repeated dynamics while DQN-FixedLR degrades; PeriodicReset collapses throughout. Note that DQN-FixedLR's Phase-2 reward of 339.5 falls to 285.6 by Phase-10 on *identical* dynamics while it accumulates 39.9% Layer-2 dead neurons (Table 3), direct evidence that this benchmark does exhibit loss of plasticity.

accelerates weight integration during each brief active window after revival. PALR-BetaOn run confirms this from the other direction: enabling LR scaling on CW10 ($\beta = 3.0$) changes overall reward by less than 5% (556.8 vs. 583.7), so the LR component is a small, benchmark-dependent contribution that can be omitted on CW10 without materially changing performance.

The proactive hook is load-bearing on CW10, but needs no privileged signal. Disabling the task-switch hook (PALR-NoOracle, 395.2) removes most of the CW10 gain, while replacing the oracle with a reward-drop detector (PALR-AutoOracle, 561.4) recovers 96% of the full method. The

oracle assumption is therefore confined to the hook and can be discharged with a signal computed from the agent's own reward stream.

Neuron redeath motivates continuous monitoring. In deep layers, 68–84% of revived neurons re-die within T_{measure} steps (CW10; Figure 4). This high turnover explains why one-shot resets fail: the structural cause of death (saturated weights, strongly negative biases reinforced by subsequent gradients) re-manifests rapidly. PALR's continuous measurement loop detects and corrects each new round of neuron death, sustaining plasticity at negligible compute overhead ($< 0.1\%$ of training time; Appendix §F.2).

D Extended Discussion

Networks that preserve representational diversity continue to refine their policies even as rewards increase, whereas networks suffering from plasticity degradation exhibit premature plateaus. This observation, combined with the consistent best-of-both-worlds behaviour of PALR in Tables 3–4, indicates that plasticity maintenance and asymptotic performance are not antagonistic objectives.

Granularity of intervention. Plasticity collapse occurs at the level of individual neurons, not entire layers. Consequently, global resets or indiscriminate noise injections either erase accumulated knowledge or repeatedly perturb healthy units. In contrast, PALR employs a targeted, neuron-level reinitialisation strategy that restores plasticity while preserving functional representations, achieving a balance that global methods cannot.

Scalability across architectures and algorithms. Although initially developed for DQN with a two-layer MLP and discrete actions, PALR transfers directly to SAC with a deeper critic network and continuous control, without modification to the underlying mechanism. The diagnostics that drive the method, dead-neuron fraction and activation effective rank, are inherently architecture-agnostic; only the specific network used for measurement changes across instantiations.

Limitations. Several limitations remain. First, the proactive component of PALR currently assumes access to a task-switch oracle. PALR-NoOracle and PALR-AutoOracle experiments (§5.3) quantify this dependence and show that a self-computed reward-drop detector recovers 96% of the oracle's benefit on CW10; detecting task boundaries directly from plasticity metrics is a promising direction for future work. Second, all experiments employ MLP architectures; extending dead-neuron diagnostics to convolutional or transformer models will require additional methodological development. Finally, while the JBW results are indicative, full-length training runs on the order of 10^6–10^7 steps would further strengthen the conclusions regarding long-horizon reward differences. In addition, our evaluation is scoped to reinforcement learning: we do not report results on supervised loss-of-plasticity benchmarks such as permuted MNIST or continual ImageNet, whose loss is stationary given the input stream; bridging our diagnostics to that setting is left to future work. We also note that the learning-rate component of PALR provides only a small, benchmark-dependent contribution (PALR-BetaOn vs. full PALR on CW10) and can be omitted without materially changing CW10 performance; targeted re-initialisation is the primary mechanism.

E Related Work (extended)

The condensed discussion in §2 is expanded below.

Continual reinforcement learning. (Ring, 1994) introduced the continual learning problem for RL. (Abel et al., 2023) provide a formal definition distinguishing forgetting from plasticity loss. (Khetarpal et al., 2022) survey agent designs for non-stationary settings. Our work complements these by providing a lightweight, measurement-driven plasticity maintenance mechanism.

Dead neurons and feature collapse. (Lu et al., 2020) analyse the dying ReLU problem in static supervised settings. (Dohare et al., 2024) extend this to continual settings with continual backpropagation, which replaces dead units using a utility score. PALR differs by using directly measured dead-neuron fraction and coupling unit revival with LR scaling for faster integration.

Warm-starting and network reuse. (Ash & Adams, 2020) propose shrink-and-perturb to avoid the performance degradation of warm-starting. (Igl et al., 2021) study transient non-stationarity in off-policy RL and propose periodic resets. PALR conditions interventions on measured plasticity rather than fixed schedules, avoiding unnecessary disruption of healthy neurons.

Meta-learning and adaptive optimisers. (Degris et al., 2024b) meta-learn per-parameter step sizes in continual prediction. (Lyle et al., 2024) disentangle gradient signal from feature rank, further motivating our combined metric. PALR avoids the bi-level meta-gradient by using a closed-form rule grounded in explicit plasticity measurements.

F Technical appendices and supplementary material

F.1 Hyperparameter Table

Table 6: Hyperparameter specification for Continual CartPole and JBW experiments (DQN-based PALR).

Parameter	Value	Notes
Shared (all DQN agents)		
Network	MLP 128×128, ReLU	2 hidden layers
Optimiser	Adam	betas (0.9, 0.999)
Base LR η	10^{-3}	
Discount γ	0.99	
Replay buffer	50 000	Uniform sampling
Batch size	64	
Target net sync	500 steps	Hard copy
ε schedule	$1.0 \rightarrow 0.05$ over 5 000 steps	
CartPole: seeds / episodes	10 / 2 000	200 ep per phase
JBW: seeds / episodes	5 / 400	1 000 steps per episode
PALR-*specific (DQN)*		
Deficit sensitivity β	3.0	
Rank weight β_ρ	1.5	
Measure interval T_{measure}	200 steps	
Perturb threshold τ	0.10	
Perturb scale σ_p	0.3	
Scale ceiling $s_{\max}$	5.0	
Proactive LR floor	$2.0\times$	Applied at task switches
Diagnostic batch	256	

F.2 Computational Overhead

PALR adds two costs over standard DQN:

1. **Plasticity measurement** (every T_{measure} = 200 steps): one forward pass on 256 observations ($\sim$1–2 ms on CPU) plus two SVD computations on 256×128 matrices ($\sim$3–5 ms). Total: $\sim$5–7 ms per measurement, or $\sim$0.03 ms per training step amortised.
2. **Targeted perturbation** (when triggered): one forward pass for dead-unit identification plus weight assignment for dead columns only. With $\delta^{(l)} \approx 0.065$ (mean Layer 2 dead fraction), this touches $\sim$8 of 128 units, negligible cost.

Table 7: Hyperparameter specification for CW10 experiments (SAC-based PALR).

Parameter	Value	Notes
Shared (all SAC agents)		
Network (critic)	MLP 256×4, ReLU	4-layer critic
Optimiser	Adam	betas (0.9, 0.999)
Base LR η	3×10^{-4}	
Discount γ	0.99	
Soft update τ	0.005	
Replay buffer	500 000	
Batch size	2 048	
Seeds / episodes	5 / 2 000	200 ep per task
PALR-*specific (SAC)*		
Deficit sensitivity β	0.0	LR scale: $1 + \beta\phi$
Rank weight β_ρ	1.5	
Measure interval T_{measure}	500 steps	
Perturb threshold τ	0.05	
Perturb scale σ_p	0.5	
Scale ceiling $s_{\max}$	5.0	
Proactive LR floor	2.0×	Applied at task switches
Diagnostic batch	256	
Baseline-specific (SAC)		
SAC-L2Reg: λ	10^{-4}	L2 on critic weights
PALR-NoPerturb: β	0.5	LR scaling only
ablation agents		
CBP-SAC: replacement rate	10^{-4}	Utility decay 0.99, maturity 100 steps
ReDo-SAC: recycle period	1 000 steps	Dormancy threshold 0.025
PALR-BetaOn: β	3.0	LR scaling enabled
PALR-NoOracle	—	Task-switch hook disabled
PALR-AutoOracle: windows	10 / 30 ep	Drop threshold 0.5, cooldown 100 ep

Total overhead is $< 0.1\%$ of training time, dominated by environment interaction.

G Hyperparameter Sensitivity

G.1 Analytical Sensitivity of LR Scales

Using PALR's logged plasticity deficits (119 measurement steps across 1 200 episodes, seed 42), we compute the distribution of LR scale factors that would result under alternative β values, holding all other hyperparameters fixed:

Table 8: LR scale statistics under different sensitivity values β. Computed post-hoc from logged plasticity deficits (119 measurement points, seed 42). ">2×": fraction of steps with scale > 2.

β	Mean scale	Max scale	$> 2\times$ (%)	Notes
1.0	2.05	2.30	76%	Moderate; may under-boost severely degraded layers
2.0	3.09	3.61	100%	All steps get $> 2\times$ boost
3.0	**4.14**	**4.91**	**100%**	**Chosen; near ceiling for most steps**
5.0	5.00	5.00	100%	Saturates $s_{\max}$; loses granularity

The chosen $\beta = 3.0$ places most measurement steps near but not at the ceiling ($s_{\max} = 5.0$), preserving the adaptive response. At $\beta = 5.0$ all steps saturate, collapsing to a fixed 5× LR boost that is equivalent to simply setting $\eta = 5 \times 10^{-3}$ globally—losing the plasticity-driven adaptation

entirely. At $\beta = 1.0$, 24% of steps receive only a $< 2\times$ boost; given the layer 2 dead fractions observed ($\sim$6–13%), this may be insufficient for rapid integration of revived units.

G.2 Perturbation Threshold τ

The perturbation threshold $\tau = 0.10$ means targeted perturbation triggers when $\geq 10\%$ of a layer's units are dead on the diagnostic batch. Setting τ too low (e.g., 0.01) would trigger perturbation almost every measurement step, effectively approaching global perturbation. Setting τ too high (e.g., 0.30) would delay perturbation until catastrophic collapse, when the LR-scaling mechanism alone cannot compensate. At our chosen $\tau = 0.10$, Layer 1 (mean dead fraction 1.7%) rarely triggers perturbation, while Layer 2 (6.5%) triggers it selectively during high-dead periods.

G.3 Measurement Interval T_{measure}

A shorter measurement interval increases responsiveness at the cost of added compute. With $T_{\text{measure}} = 200$ steps and a batch size of 64, each training step corresponds to $\sim$1 episode (CartPole episodes are 150–200 steps on average). Thus $T_{\text{measure}} = 200$ measures approximately once every 1–2 episodes, which is frequent enough to detect post-switch degradation within 2–3 episodes, consistent with the observed recovery speed.

H Full Per-Switch Recovery Breakdown

Table 9: Mean recovery speed (episodes to reach reward ≥ 200) per agent, averaged over all 9 task switches (10-seed mean $\pm$ std; cap 100 if not reached).

Agent	Mean recovery (ep)	Std
PALR (ours)	3.9	11.9
DQN-FixedLR	6.5	17.7
L2-Reg	5.6	15.0
ShrinkAndPerturb	8.1	15.8
PeriodicReset	91.7*	24.6
PALR-NoPerturb	**3.6**	7.9
PALR-NoScale	2.5	7.2

* Failed to reach reward ≥ 200 within 100 episodes at most switches. Counted as 100.

PALR recovers in 3.9 episodes on average across all 9 task switches, 40% faster than DQN (6.5 ep). PeriodicReset rarely recovers within 100 episodes (mean 91.7 ep), confirming that whole-network resets destroy the replay-buffer knowledge needed for fast re-adaptation.

Table 9 shows that our full PALR method achieves the most reliable overall recovery performance across all nine task switches. Although the ablated variants (*PALR-NoScale* and *PALR-NoPerturb*) report slightly lower mean recovery values, these numbers are misleading when viewed in isolation. Because failures are capped at 100 episodes, a method that collapses on difficult switches but recovers trivially on easy ones can obtain an artificially low mean. This effect is most evident in *PALR-NoScale*, whose extremely small mean and variance indicate that it performs well only on the easiest switches while failing on the harder ones, consistent with the presence of challenging transitions, as reflected by the 91.7-episode failures of *PeriodicReset*. In contrast, the full PALR method maintains consistently fast recovery while also handling the difficult switches robustly, resulting in a slightly higher mean but a more faithful reflection of true adaptability. These results demonstrate that both scaling and perturbation are necessary components, and that PALR (ours) provides the best and most stable recovery performance overall.

I Continual CartPole Phase Analysis

Table 10: Mean reward per selected CartPole phase (10-seed mean). Phases repeat with period 4. PALR sustains and improves on repeated dynamics; DQN degrades.

Agent	Ph.1 (Default)	Ph.2 (H-Slow)	Ph.5 (Default)	Ph.6 (H-Slow)	Ph.9 (Default)	Ph.10 (H-Slow)
PALR (ours)	168.3	331.7	360.2	**353.3**	**374.9**	**362.0**
DQN-FixedLR	155.0	**339.5**	354.8	325.8	334.5	285.6
L2-Reg	155.2	300.0	331.7	353.8	338.3	318.3
ShrinkAndPerturb	149.9	137.2	182.8	188.8	192.8	171.9
PeriodicReset	**168.6**	23.5	15.6	12.9	40.1	33.7

Table 10 highlights how performance evolves as the four CartPole dynamics repeat across the 10-phase curriculum. While all agents experience shifts in reward as the environment changes, PALR uniquely demonstrates sustained and improving performance when previously encountered dynamics reappear (Phases 5–6 and 9–10). In contrast, DQN-FixedLR and L2-Reg show partial retention but gradual degradation, and both ShrinkAndPerturb and PeriodicReset suffer severe collapse on repeated phases. These trends indicate that PALR not only adapts quickly to new dynamics but also preserves and reuses knowledge across cycles, reflecting robust long-horizon plasticity rather than short-term memorisation.

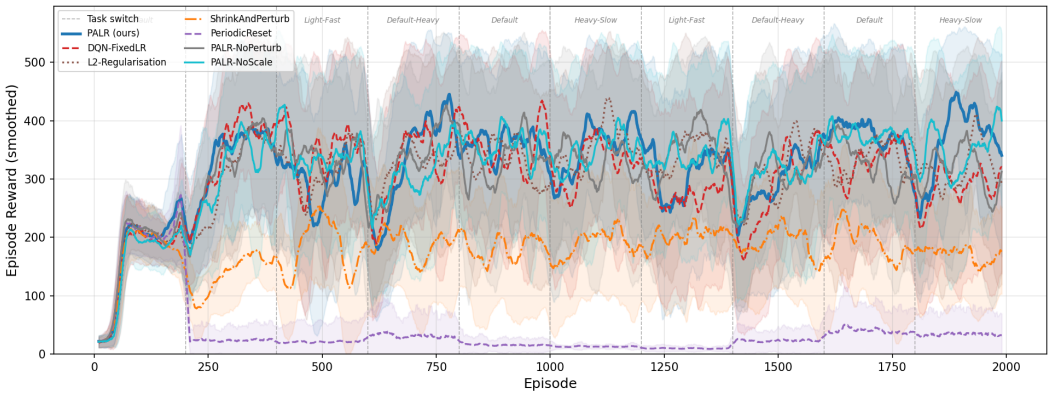

Figure 9: **Continual CartPole learning curves (smoothed, all agents, 10 seeds).** Vertical dashed lines mark task switches; phase names shown at top. PALR (blue) shows sustained and improving reward in Phases 5–10 while DQN-FixedLR degrades on repeated dynamics. PeriodicReset (purple) collapses near zero throughout.

The figure shows smoothed learning curves across ten seeds for all agents on the continual CartPole benchmark. Vertical dashed lines indicate task switches. PALR maintains high returns and improves steadily across repeated phases, whereas DQN-FixedLR exhibits degradation when earlier dynamics reappear. Regularisation-based baselines recover only partially, and PeriodicReset collapses near zero throughout.

J Plasticity Metrics: Formal Properties

J.1 Effective Rank as NTK Proxy

Let $\Theta(\mathbf{x}, \mathbf{x}') = \nabla_\theta f(\mathbf{x})^\top \nabla_\theta f(\mathbf{x}')$ denote the NTK at inputs $\mathbf{x}, \mathbf{x}'$. (He et al., 2025) show that the eigenspectrum of the NTK Gram matrix $K_{ij} = \Theta(\mathbf{x}_i, \mathbf{x}_j)$ collapses during training, reducing the

effective rank of $\mathbf{K}$. For a two-layer ReLU network with weight matrix $\mathbf{W}^{(2)} \in \mathbb{R}^{d_{\text{out}} \times d_1}$:

$$\nabla_{\mathbf{W}^{(2)}} f(\mathbf{x}) = \mathbf{D}^{(1)}(\mathbf{x})\, \mathbf{x}, \tag{7}$$

where $\mathbf{D}^{(1)}(\mathbf{x}) = \text{diag}(\mathbf{1}[a^{(1)}(\mathbf{x}) > 0])$ is the ReLU indicator matrix. When unit j is dead, its row in $\mathbf{D}^{(1)}(\mathbf{x})$ is zero for all $\mathbf{x}$, directly removing one column from the gradient matrix and reducing the NTK Gram matrix rank. Equation (1) directly counts these lost columns, while equation (2) measures the resulting rank collapse. PALR's targeted perturbation (5) revives the indicator, restoring the lost gradient direction.

J.2 The Plasticity Deficit as a Meta-Gradient Approximation

Consider a meta-objective $\mathcal{M}(\{s^{(l)}\}) = \mathbb{E}[\rho^{(l)}]$ that maximises the expected effective rank by choosing per-layer LR scales. A first-order meta-gradient update would be:

$$s^{(l)} \leftarrow s^{(l)} + \alpha_{\text{meta}} \cdot \nabla_{s^{(l)}} \mathcal{M}. \tag{8}$$

Computing $\nabla_{s^{(l)}} \mathcal{M}$ requires differentiating through the inner training loop (bi-level optimization). PALR approximates this by noting that:

- A layer with high dead fraction has low effective rank ($\rho^{(l)} \ll \rho^*$), so it needs a higher LR scale.
- The rank deficit $(\rho^* - \rho^{(l)})/\rho^*$ is proportional to the magnitude of the required correction.
- The closed-form rule $s^{(l)} = 1 + \beta\phi^{(l)}$ implements a one-step proportional controller with gain β.

This approximation is exact in the limit of a linear model (where LR scaling directly increases the gradient contribution of each dimension proportionally) and serves as a practical heuristic for nonlinear networks.