NEVER STOP REWIRING: SAFE CONTINUAL REINFORCEMENT LEARNING ON NON-RESETTING GRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

A network agent that is to remain autonomous over a long horizon must keep learning as failure processes arrive, vanish and return, and it must do so without ever breaking the network, because on a graph that is never reset a wrong repair rewrites the topology that every later failure acts on. Continual RL supplies the first requirement and safe RL the second, but each is studied without the other: continual RL on benchmarks that reset between tasks, safe RL in episodes that undo a violation. We argue that lifelong network repair needs their combination, and study it as task-free safe continual RL on a non-resetting graph: a non-stationary constrained MDP with a service reward, a connectivity safety cost, no mode labels, and a *budget-B unrecoverable event*, a violation that no B edge additions can repair. Across Erdős–Rényi, Watts–Strogatz and Barabási–Albert streams with recurring and unseen failure modes, we obtain three findings. Reward forgetting, the failure continual RL is built to prevent, is near zero for every learner; what ends autonomy is an *absorbing violation state* into which the service reward leads the agent and out of which later learning does not climb. Continual learning alone makes this worse: PPO is absorbed in 17 of 30 lifetimes, policy consolidation in 26, because the cascade stores the unsafe behaviour; a Lagrangian constraint alone only delays it (12 of 30). Their combination, constrained policy consolidation, which prices the constraint on the visible policy of a multi-timescale cascade, keeps rewiring for 29 of 30 lifetimes with the lowest violation rate in every graph family and a violation rate that falls on every return to a mode, without task boundaries. For a network to become autonomous over its whole lifetime, its agent must therefore never stop learning and never stop being constrained: the reward it can relearn, the safe behaviour it must carry forward.

1 INTRODUCTION

Infrastructure networks fail in structured ways: random edge loss, attacks on the highest-degree hub, spatially concentrated outages, targeted removal of the bridges that hold components together (Albert et al., 2000; Artime et al., 2024). Learned controllers already act on graph structure online, reconnecting lines in a power grid (Yoon et al., 2021), steering a process by editing edges (Song et al., 2024) or selecting nodes (Meirom et al., 2021). They are trained and evaluated under a fixed failure process. Over a long deployment the failure process changes without announcement, earlier processes return, and some have not been seen before; in the terms of Abel et al. (2023), the agent has to keep learning for its whole lifetime.

Continued learning has a cost. On-policy updates overwrite earlier behaviour, and forgetting appears inside a single non-stationary stream, not only across benchmark suites (Kaplanis et al., 2018; 2019). On a graph, what is overwritten is a repair behaviour, and a wrong edge changes the state that every later failure process acts on: as in reset-free RL (Lu et al., 2021), the graph is not restored when a mode changes, a fragmented topology is inherited, damage cascades (Buldyrev et al., 2010), and recovery depends on where the intervention lands (Sanhedrai et al., 2022), so a one-edge-per-step budget cannot always undo it. Forgetting therefore produces a safety event, possibly one from which the system cannot return within its reconfiguration budget (Eysenbach et al., 2018; Grinsztajn et al., 2021; Plaut et al., 2026).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

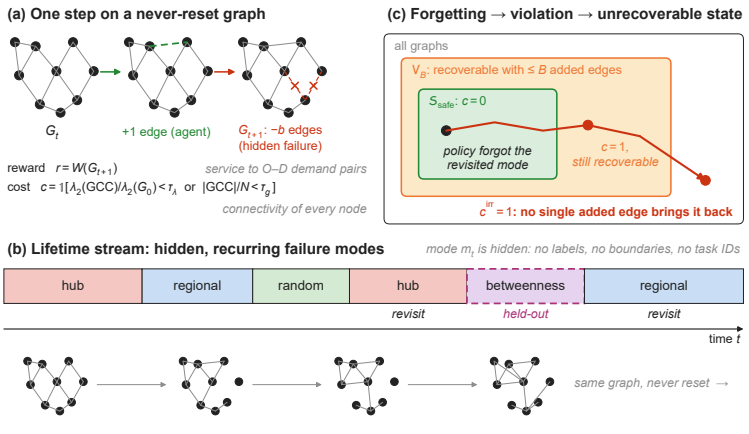


Figure 1: Lifelong reconfiguration on a non-resetting graph. (a) One step: the agent adds an edge, a hidden failure process removes edges; the service reward and the connectivity cost are separate signals. (b) Failure modes are hidden and recur; the topology is inherited across modes. (c) Forgetting a repair can push the graph out of the safe set $\mathcal{S}_{\mathrm{safe}}$ and, further, out of the set $\mathcal{V}_B$ from which B legal edge additions can restore safety.

Safe RL (Achiam et al., 2017; Ji et al., 2023) and safe continual RL (Coursey et al., 2025; 2026) address halves of this problem, the latter with known boundaries and both in episodic environments where an unsafe action does not rewrite the future state space; task-agnostic continual RL (Caccia et al., 2023) removes the boundary signal but considers neither safety nor irreversibility (Section 2).

This paper studies the combination of the two on a network that is never reset. We make four contributions.

1. **Problem.** We formalise lifelong network reconfiguration as task-free safe continual RL: a hidden-parameter constrained MDP whose hidden parameter is the failure process, with a service reward, an independent connectivity cost, no labels or boundaries, and a non-resetting state. Because the graph is never reset, we need a notion of irreversibility that a finite reconfiguration budget can certify: the *budget-B unrecoverable event* (no B legal additions return the graph to safety). It is the per-step certificate behind the lifetime-level failure we find below: inside an absorbing violation almost every step is budget-1 unrecoverable, so the failure is a property of the graph the learner has produced, not only of its cost trace (Section 3).
2. **Diagnosis.** Using a 2×2 factorial over {constraint, consolidation} on a shared graph policy, we show that the characteristic failure on a non-resetting graph is an *absorbing violation state* that learners, not the environment, produce: no one-step rule other than no-op ever enters it, unconstrained learners enter it in most lifetimes, consolidation alone makes entry more likely, and a constraint alone only delays it. We further show that *reward forgetting* and *safety forgetting* are distinct quantities: on lifetime streams the former is near zero for every learner while the latter separates them (Section 5).
3. **Method.** We propose constrained policy consolidation (CPC): a Lagrangian price on the visible policy handles the present constraint, while a cascade of slower shadow policies stores the *constrained* behaviour across timescales without task boundaries (Section 4). The two components are complementary rather than additive: the cascade consolidates safe repair behaviour, and at a mode switch the cascade slows the policy while the multiplier bounds it. CPC is the only learner that almost never enters the absorbing state, and the only one whose safety forgetting is negative with an interval excluding zero.
4. **Protocol.** A lifetime-stream protocol whose stream order is derived from a measured mode-pair interference matrix, whose hyperparameters are chosen once on a tuning stream, whose claims were pre-registered, and whose evaluation reports lifetime service, reward and safety forgetting, switch-aligned cost, violations and absorbing states with interval estimates.

2 Related Work

Safe RL and constrained MDPs. Constrained MDPs separate what the agent should achieve from what it must not do (Altman, 1999): CPO enforces the constraint inside a trust region (Achiam et al., 2017), Lagrangian PPO turns it into an adaptive price (Ray et al., 2019; Ji et al., 2023), CVPO recasts it as inference (Liu et al., 2022), model-based variants exist (Jayant & Bhatnagar, 2022). Because expected cumulative cost can hide single catastrophic steps, step-wise constraints (Xiong et al., 2023) and reachability-based feasible sets (Yu et al., 2022; Ganai et al., 2023; Wachi & Sui, 2020) have been proposed. All assume a stationary constraint and, mostly, a stationary environment.

Continual RL without boundaries. Continual World established forgetting and forward transfer as the axes of continual RL evaluation (Wołczyk et al., 2021; 2022); interference between consecutive tasks can be measured pairwise before building long sequences (Ahn et al., 2025). Regularisation (Kirkpatrick et al., 2017; Zenke et al., 2017; Schwarz et al., 2018) and replay (Rolnick et al., 2019) need boundaries or stored transitions; on an evolving graph a stored edge may already exist or no longer bridge anything. Policy consolidation (Kaplanis et al., 2019), rooted in multi-timescale synaptic models (Benna & Fusi, 2016; Kaplanis et al., 2018), needs neither. Task-agnostic CRL (Caccia et al., 2023) removes task identity but not the episodic reset. Reset-free learning (Lu et al., 2021) and the critique of lifetime tuning (Mesbahi et al., 2025) shape our protocol; plasticity loss over long streams (Dohare et al., 2024; Nikishin et al., 2022; Lyle et al., 2023; Tang et al., 2025) is treated as a limitation.

Safe continual RL. Safe lifelong policy search bounded regret under constraints on the policy parameters rather than on a state-dependent cost (Bou Ammar et al., 2015). Coursey et al. (2025; 2026) first studied the interaction: CPO forgets on revisits, PPO+EWC ignores constraints, a cost-shaped EWC sits in between, and task switches trigger cost spikes even when the constraint is fixed. We run their Safe EWC, given the true switch times, as a boundary-aware reference (Appendix G). Our setting differs in four ways: no boundaries; a non-resetting state whose future is rewritten by unsafe actions; an irreversibility notion tied to a reconfiguration budget; and a combinatorial action space, a masked categorical over $\approx 10^3$ absent edges whose legal set changes every step, where continual and safe RL benchmarks act through at most 18 discrete actions or a few continuous joints (Kirkpatrick et al., 2017; Wołczyk et al., 2021; Ray et al., 2019; Ji et al., 2023).

Irreversibility, and network repair. Leave No Trace treats states from which an agent cannot return as reset risks (Eysenbach et al., 2018); reversibility-aware RL learns which transitions cannot be undone (Grinsztajn et al., 2021); no-reset learning under irreversible dynamics is now studied formally (Plaut et al., 2026); like them we never claim global irreversibility. On the network side, error and attack tolerance (Albert et al., 2000), cascades (Buldyrev et al., 2010), spontaneous recovery (Majdandzic et al., 2014) and revival (Sanhedrai et al., 2022) describe failure; learned combinatorial decisions on graphs (Khalil et al., 2017; Fan et al., 2020) and learned topology control (Yoon et al., 2021; Song et al., 2024) describe control. Neither line evaluates whether a repair policy, having learned a new process, still remembers an old one.

3 Problem: Task-Free Safe CRL on a Non-Resetting Graph

3.1 Graph MDP with hidden failure modes

At step t the network is $G_t = (V, E_t)$ with fixed $N = |V|$ and a changing edge set. The agent adds one absent edge $a_t \in \bar{E}_t$, subject to a capacity $|E_t| \leq |E_0|$ (an addition at capacity is a no-op). A hidden failure mode $m_t \in \mathcal{M}$ then strikes: with probability p per step it removes a burst of b edges in a mode-specific priority order, under a global damage cap; modes remove the same number of edges and differ *only in where the damage lands*. The agent observes $(G_t, a_t, r_t, c_t, G_{t+1})$ and never m_t, a task index or a change-point flag. Modes share (S, A, r, c, γ) and differ in $p_m(G' \mid G, a)$: a hidden-parameter MDP (Fu et al., 2023; Xie et al., 2021) whose hidden parameter is the failure process. Visible modes are the classical error/attack pair, random failure and degree-ordered hub attack (Albert et al., 2000), and a regional outage (edges ordered by distance to a centre fixed at the last switch); a fourth, late-arriving mode that no agent sees before the second half of life removes edges in order of betweenness, i.e., bridges first (Holme et al., 2002). A lifetime is one trajectory of

T steps on one initial graph; modes dwell for random lengths and recur, the novel mode arrives late and recurs once, and the graph is never restored. Labels and boundaries exist only in the evaluation log. The question is: *how should an agent keep learning on this stream so that it acquires repairs for new modes, keeps repairs for returning modes, and stays inside the safe set while doing both?*

3.2 Reward and safety cost are separate signals

The reward measures *service*: a fixed set $\mathcal{D}$ of m origin–destination pairs is drawn once per graph, and

$$W(G) = \frac{1}{|\mathcal{D}|} \sum_{(s,t)\in\mathcal{D}} \frac{1}{d_G(s,t)}, \qquad r_t = W(G_{t+1}), \tag{1}$$

with $1/d = 0$ for unreachable pairs: the Latora–Marchiori efficiency (Latora & Marchiori, 2001) restricted to the pairs that carry demand. The reward is the service *level*, not its change, so a policy that lets service decay and repairs it later is not paid the same as one that never let it decay. Demand nodes are visible through a node feature. Safety, in contrast, protects *every* node: an indicator on two independent connectivity thresholds,

$$c_t = \mathbb{1}\Big[\lambda_2\big(\mathrm{GCC}(G_{t+1})\big)/\lambda_2(G_0) < \tau_\lambda \;\;\vee\;\; |\mathrm{GCC}(G_{t+1})|/N < \tau_g\Big], \tag{2}$$

where λ_2 is algebraic connectivity on the giant component GCC and G_0 the initial graph: τ_g measures how many nodes have been cut off, τ_λ how thin the surviving core has become. The safe set is $\mathcal{S}_{\mathrm{safe}} = \{G : c(G) = 0\}$.

These are two standard quantities for transport networks, shortest-path efficiency between demand points and the probability that a failure disconnects part of the network, and Tero et al. (2010) report that real and biological networks trade one against the other. We verify that they conflict in our environment: on the tuning stream a one-step reward-greedy policy reaches $W \approx 0.95$ while violating on 22% of steps, a one-step safety-greedy policy violates on 3% but serves little better than random (0.43 versus 0.36), and the two choose different edges on 96% of steps (Appendix E). A global-efficiency reward, by contrast, is aligned with the connectivity cost at the one-step level (the edge that shortens most paths also most raises λ_2), which is why we do not use it.

3.3 Non-stationary constrained objective

Because the stream has no episodes, the constraint is on the windowed average cost:

$$\max_{\pi} \; \mathbb{E}_\pi\Big[\textstyle\sum_t \gamma^t r_t\Big] \quad \text{s.t.} \quad \bar{c}_W(\pi) = \frac{1}{W}\sum_{s=t-W+1}^{t} c_s \le d. \tag{3}$$

This is the standard CMDP form (Altman, 1999; Achiam et al., 2017) with a non-stationary kernel; the agent must satisfy it across modes it cannot identify.

3.4 Budget-B unrecoverable events

Let $\mathcal{V}_B = \{G : \exists E^+ \subseteq \bar{E},\ |E^+| \le B,\ G \cup E^+ \in \mathcal{S}_{\mathrm{safe}}\}$ be the set of graphs that B legal edge additions can return to safety. A *budget-B unrecoverable event* is $c_t^{\mathrm{irr}} = \mathbb{1}[G_{t+1} \notin \mathcal{V}_B]$, computed exactly by enumeration for $N \le 80$, $B \in \{1, 2\}$, and only when $c_t = 1$. It is an evaluation quantity; the constraint in Eq. (3) uses c_t only. We avoid the word *irreversible*, which would require an infinite-horizon reachability argument (Yu et al., 2022; Plaut et al., 2026). Under ten-edge bursts most violated steps are budget-1 unrecoverable for every policy (68–83%, Table 5), so the event is not informative as a rate; its use is to separate a violation the agent will leave from one it will not (Section 5).

4 Method: Constrained Policy Consolidation

Every agent shares one actor architecture: a two-layer GraphSAGE encoder (Hamilton et al., 2017) over node features (degree, clustering, component-size fraction, core number, giant-component membership, demand) and a symmetric bilinear edge scorer producing a masked categorical over all absent edges; reward and cost critics are separate encoder–pooling networks. In the cascade every π_k is a *complete* actor with its own encoder and head, as in Kaplanis et al. (2019). Holding the architecture fixed lets the 2×2 comparison be attributed to the learning objective.

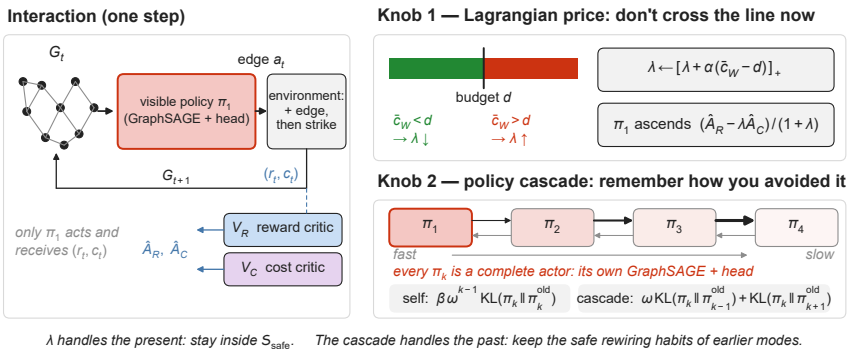


Figure 2: Constrained policy consolidation. Left: only the visible policy π_1 acts and receives (r_t, c_t); reward and cost critics are separate networks. Right: a Lagrangian price λ on π_1 handles the current constraint; a cascade of progressively slower shadow policies stores the constrained behaviour across timescales with no task boundary.

Two design choices. The four learners differ in two binary choices: whether the learner *prices the constraint*, and whether it *retains its past behaviour*. PPO (Schulman et al., 2017) does neither. PPO-Lagrangian (Ray et al., 2019) prices the constraint: a scalar λ rises when the window cost exceeds d and falls otherwise, and the policy maximises $\hat{A}_R - \lambda \hat{A}_C$; its memory is one scalar recording how much it has recently violated, not how to avoid violating under a mode it has left. Policy consolidation (Kaplanis et al., 2019) retains behaviour: a cascade $\pi_1, \ldots, \pi_K$ of policies with exponentially increasing inertia is coupled by KL terms, so the visible π_1 is pulled towards what it used to do over several timescales, without any boundary; it stores behaviour indiscriminately, and nothing tells it to stay safe *now*.

Constrained policy consolidation (CPC). CPC installs the Lagrangian price on the visible policy of the cascade. With $\rho_t = \pi_1(a_t \mid s_t)/\pi_1^{\mathrm{old}}(a_t \mid s_t)$ and normalised advantages, the loss is

$$\mathcal{L}_{\mathrm{PG}} = -\mathbb{E}_t\Big[\rho_t \, \tfrac{\hat{A}_{R,t} - \lambda \hat{A}_{C,t}}{1+\lambda}\Big], \qquad \lambda \leftarrow \max\big(0,\ \lambda + \alpha_\lambda(\bar{c}_W - d)\big), \tag{4}$$

$$\mathcal{L}_{\mathrm{self}} = \mathbb{E}_t\Big[\textstyle\sum_{k=1}^{K} \beta\, \omega^{k-1}\, D_{\mathrm{KL}}\big(\pi_k \,\|\, \pi_k^{\mathrm{old}}\big)\Big], \tag{5}$$

$$\mathcal{L}_{\mathrm{casc}} = \mathbb{E}_t\Big[\omega_{1,2} D_{\mathrm{KL}}(\pi_1 \| \pi_2^{\mathrm{old}}) + \textstyle\sum_{k=2}^{K} \omega D_{\mathrm{KL}}(\pi_k \| \pi_{k-1}^{\mathrm{old}}) + \sum_{k=2}^{K-1} D_{\mathrm{KL}}(\pi_k \| \pi_{k+1}^{\mathrm{old}})\Big], \tag{6}$$

with total loss $\mathcal{L}_{\mathrm{PG}} + \mathcal{L}_{\mathrm{self}} + \mathcal{L}_{\mathrm{casc}} + c_v \mathcal{L}_V - c_e \mathcal{H}$. Eqs. (5)–(6) are the policy-consolidation objective of Kaplanis et al. (2019) with their reverse-KL direction; the last policy π_K has no slower neighbour and hence no downward term. Eq. (4) is the PPO-Lagrangian surrogate. Setting $\lambda \equiv 0$ recovers PPO+PC; setting $K = 1$ recovers PPO-Lagrangian; both recover PPO.

Why the combination helps. First, the cascade consolidates a policy that has been constrained throughout, so the shadow policies store *safe* repair behaviour; safety knowledge acquires long-term storage instead of living only in the scalar λ. Second, at a mode switch the reward gradient pulls π_1 away, the cascade slows that pull and the rising λ bounds it. Third, on a revisit π_1 returns towards shadow policies that still hold the old behaviour, and λ need not climb again. The predicted cost is plasticity on a new mode. Algorithm 1 (Appendix A) gives the on-stream loop.

5 Experiments

The experiments answer three questions whose reading was fixed before the runs (Appendix B). **C1 (necessity):** does a constrained-only learner forget on revisits, and does a continual-only learner violate the constraint? **C2 (forgetting is risk):** are reward and safety forgetting distinct, and do violations concentrate after switches? **C3 (joint mitigation):** does CPC lower both kinds of forgetting and, at matched violation, achieve higher service across graph families?

5.1 PROTOCOL

Environments and streams. Graphs with $N = 50$ nodes from Erdős–Rényi ($\bar{k} = 4$), Watts–Strogatz ($k = 4, p = 0.3$) and Barabási–Albert ($m = 2$) families (Erdős & Rényi, 1960; Watts & Strogatz, 1998; Barabási & Albert, 1999); $|\mathcal{D}| = 8$ demand pairs. A strike occurs with probability $p = 0.08$ per step and removes $b = 10$ edges. The expected damage (0.8 edge per step) is below the repair rate, so $|E_t|$ drifts back to capacity after every strike and the threat comes from *where* edges are removed, not from an edge deficit (with $p\,b = 1$ every policy drifts to the damage floor; Appendix E). A lifetime has $T = 20000$ steps; segments dwell 1500–3000 steps (12–23 updates, enough for the unconstrained learner to acquire a mode) in the fixed *Hard* order hub, regional, random, hub, regional, betweenness, random, hub, betweenness, so the late-arriving mode enters after every other mode has been seen. Following the practice of measuring pairwise interference before building long sequences (Wołczyk et al., 2021; Ahn et al., 2025), the Hard order places adjacently the pairs that interfered most in $i \to j \to i$ pilot runs with PPO (Appendix D); an *Easy* order that keeps them apart and a per-seed *Random* order are run on ER (Appendix F).

Agents. The 2×2 factorial on a shared architecture: PPO; PPO-Lagrangian (constraint only); PPO+PC (consolidation only); CPC (both). Four non-learning anchors run on the same streams: *no-op*, *random* (a uniform legal edge), *reward-greedy* and *safety-greedy* (one-step lookahead on W and on the safety margin over 120 sampled candidates). They bound what is achievable without memory.

Hyperparameters and thresholds. Each method is tuned once on a *tuning stream* (ER, disjoint graph seeds); constrained methods are selected by the violation rate of the last block subject to λ not saturating, all others by lifetime service. Selected values are then frozen for all families and orders (Mesbahi et al., 2025). All learners update every 128 steps for 8 epochs with $\gamma = 0.95$. Consolidation uses $\beta = 0.5$, $\omega = 4$ from Kaplanis et al. (2019); $K = 4$ was fixed before the runs (their cascade has eight policies) and is varied in Appendix G; $\omega_{1,2} = 0.1$, because over a $\binom{N}{2}$-way action space the $\omega_{1,2} = 4$ stated in their main text pins the visible policy to its initially uniform shadow. The Lagrangian uses $\alpha_\lambda = 0.1$, $\lambda_{\max} = 5$; larger steps let λ saturate. The cost critic is trained on a shaped cost (indicator plus normalised distance below each threshold); the budget, the multiplier update and every reported number use the indicator only. Thresholds and strike intensity $(\tau_g, \tau_\lambda, p, b)$ were calibrated once with three non-learning policies and frozen at $(0.96, 0.30, 0.08, 10)$, the cell where a safety-greedy policy violates on $\leq 5\%$ of steps, a reward-greedy one on 15–40% and a random one on $\geq 30\%$; $d = 0.05$. All of this is in Appendix E.

Metrics. Lifetime service $\bar{W} = \frac{1}{T}\sum_t W(G_t)$. At every switch and every $\Delta = 500$ steps a frozen copy of the policy is probed on five fixed graphs under each mode for 200 steps; service and violation over the second half give a probe service $P_m(t)$ and probe violation $C_m(t)$ that isolate the policy from the path-dependent online state (as in Wołczyk et al., 2021; shorter probes have no dynamic range, Appendix E). For each revisit of a visible mode, *reward forgetting* F_R is the drop in P_m between leaving m and re-entering it, in service units (Wołczyk et al., 2021; Coursey et al., 2026), and *safety forgetting* F_C the corresponding rise in C_m. We further report the violation rate $\frac{1}{T}\sum_t c_t$, *unrecovered* violations (runs of $c_t = 1$ lasting ≥ 50 steps), whether and when a lifetime enters an *absorbing* violation—a run of $c_t = 1$ of at least 1000 steps (eight updates, longer than any strike-induced excursion under a non-learning policy) that persists to the end of life—switch-aligned cost after each change-point (Coursey et al., 2026), and acquisition of the novel mode as the area under $P_{\mathrm{btw}}(t)$. All aggregates use 10 seeds with IQM and 95% bootstrap intervals (Agarwal et al., 2021).

5.2 RESULTS

Absorbing violation states (C1). Figure 3 and Table 1 summarise the four learners on the Hard streams. Within the first hub segment PPO and PPO+PC raise service above the safety-greedy rule, and their violation rate climbs with it: the reward pays for connecting eight pairs, and the edges that raise it fastest, long chords between demand nodes, leave the rest of the graph to fragment. Once the giant component has fallen below τ_g the indicator cost is constant and its advantage zero, so an unconstrained objective receives no gradient towards leaving it: the lifetime enters an *absorbing violation state*. PPO is absorbed in 17 of 30 lifetimes across the three families, PPO+PC in 26 and

Table 1: Main results on Hard streams, $T = 20000$, $d = 0.05$: IQM over 10 seeds with 95% bootstrap intervals (IQM only for the one-step rules). Absorbed: lifetimes of 10 ending in an absorbing violation state. F_R, F_C: forgetting on revisits (negative = improvement on return). Novel: probe-service AUC on the late-arriving mode. Not shown: *no-op* (absorbed within 70 steps in 30/30) and *random* ($\bar{W} \approx 0.33$, violation 0.57–0.65, never absorbed).

	Agent	$\bar{W}$ ↑	Violation rate↓	Absorbed↓	F_R ↓	F_C ↓	Novel↑
ER	safety-greedy	0.353	0.145	0	–	–	–
	reward-greedy	0.949	0.552	0	–	–	–
	PPO	0.541 [0.40,0.64]	0.586 [0.34,0.87]	4	−0.018 [−0.08,0.03]	−0.039 [−0.13,0.05]	0.461
	PPO+PC	0.651 [0.58,0.71]	0.904 [0.75,0.97]	9	0.022 [−0.03,0.06]	0.054 [−0.03,0.10]	0.392
	PPO-Lag	0.448 [0.39,0.49]	0.354 [0.30,0.53]	1	0.007 [−0.02,0.03]	−0.056 [−0.13,0.04]	0.397
	CPC	0.425 [0.38,0.46]	**0.302** [0.24,0.39]	**0**	−0.036 [−0.09,0.02]	**−0.113** [−0.15,−0.06]	0.403
WS	safety-greedy	0.354	0.121	0	–	–	–
	reward-greedy	0.952	0.532	0	–	–	–
	PPO	0.612 [0.53,0.66]	0.760 [0.51,0.93]	7	−0.001 [−0.04,0.03]	0.009 [−0.06,0.05]	0.442
	PPO+PC	0.674 [0.63,0.71]	0.911 [0.63,0.96]	8	−0.007 [−0.05,0.03]	0.040 [−0.05,0.13]	0.506
	PPO-Lag	0.409 [0.32,0.47]	0.444 [0.27,0.62]	5	0.014 [−0.00,0.04]	−0.029 [−0.14,0.08]	0.335
	CPC	0.456 [0.42,0.48]	**0.297** [0.25,0.39]	**1**	−0.016 [−0.05,0.02]	**−0.060** [−0.16,0.04]	0.426
BA	safety-greedy	0.346	0.265	0	–	–	–
	reward-greedy	0.952	0.626	0	–	–	–
	PPO	0.552 [0.45,0.67]	0.798 [0.58,0.95]	6	0.036 [0.00,0.06]	−0.009 [−0.08,0.09]	0.395
	PPO+PC	0.603 [0.54,0.67]	0.893 [0.79,0.96]	9	0.002 [−0.05,0.05]	−0.040 [−0.10,0.02]	0.531
	PPO-Lag	0.355 [0.31,0.46]	0.556 [0.47,0.74]	6	0.010 [−0.00,0.04]	−0.052 [−0.12,0.09]	0.325
	CPC	0.408 [0.37,0.44]	**0.504** [0.44,0.61]	**0**	0.011 [−0.02,0.04]	**−0.075** [−0.15,−0.00]	0.404

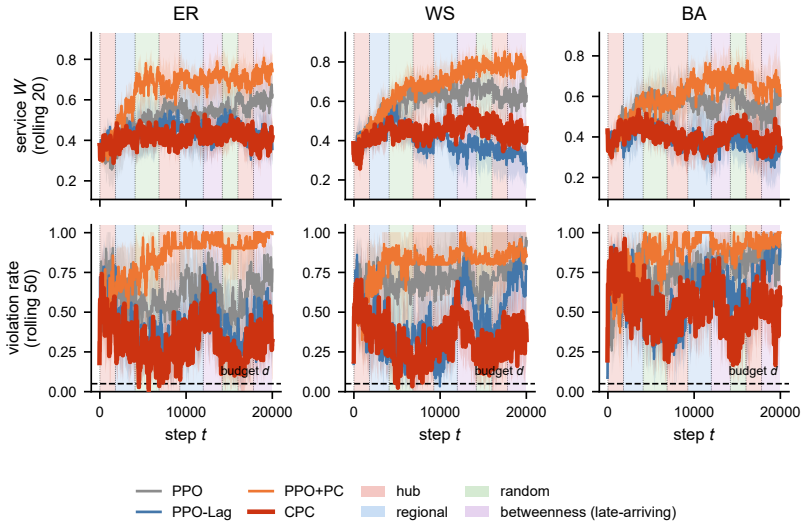


Figure 3: Lifetime service (top, 20-step rolling mean) and violation rate (bottom, 50-step rolling mean) on Hard streams over the three families; background bands are the hidden modes, dotted verticals the switches; mean and 95% CI over 10 seeds. Dashed line: the budget $d = 0.05$.

earlier (Fig. 5b): the cascade stores the behaviour of π_1, which in these lifetimes is unsafe. The state is produced by learning rather than by the environment: the one-step rules (reward-greedy at a violation rate of 0.55–0.63, random at 0.57–0.65) are never absorbed, because they have no parameters for the reward to move; only no-op is, within 70 steps. The budget-B oracle characterises the state: inside an absorbing run 95–100% of steps are budget-1 unrecoverable, against 68–83% of violated steps elsewhere (Table 5). No single edge leaves the state, yet a random policy leaves it within tens of steps, so it is the learned policy that keeps the agent there. PPO-Lag is absorbed in 12 of 30 lifetimes, mostly late in life: a scalar multiplier remembers how much it violated recently, not how to avoid violating under a mode it has left. CPC is absorbed once in 30 lifetimes and has the lowest violation rate in every family (0.30, 0.30, 0.50 against 0.35, 0.44, 0.56 for PPO-Lag); its service is below PPO-Lag on ER (0.43 against 0.45, intervals overlapping) and above it on WS and BA (0.46 against 0.41; 0.41 against 0.36).

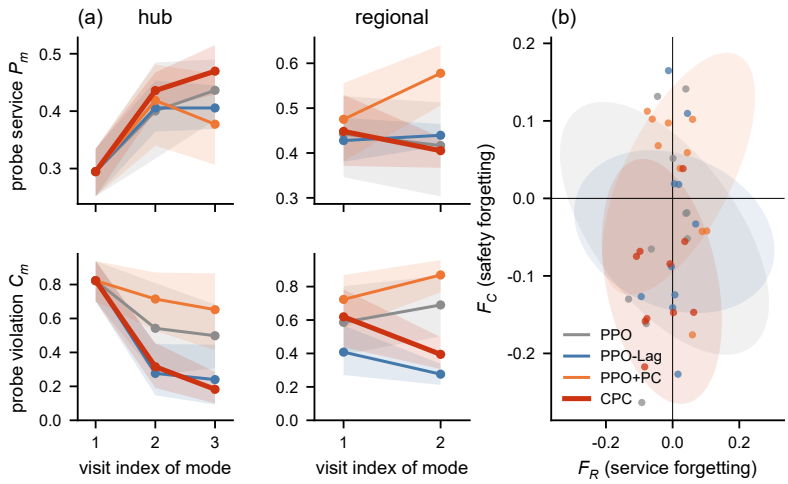


Figure 4: Reward versus safety forgetting (ER). (a) Probe service P_m (top) and probe violation C_m (bottom) at the first probe after each entry into hub and regional, versus visit index; mean and 95% CI over 10 seeds. (b) F_R against F_C per revisit, one point per seed with 95% ellipses; negative values are improvement on return.

Two kinds of forgetting (C2). Reward forgetting, the quantity C1 also predicted for the constrained learner, is indistinguishable from zero for every learner and family ($|F_R| \leq 0.04$, all intervals containing zero except PPO on BA at $+0.04$). Figure 4(a) shows why: hub probe service rises from 0.30 at the first visit to 0.40–0.47 at the second and does not fall back on the third (in the pilot $i \to j \to i$ streams of Appendix D the same learner did forget, $F_R = 0.11$–0.12; with every mode revisited, pairwise interference does not accumulate). Safety forgetting is a different quantity and does separate the learners. Hub probe violation stays high across visits for the unconstrained learners (PPO $0.82 \to 0.54 \to 0.50$, PPO+PC $0.82 \to 0.72 \to 0.65$) and falls with every visit for the constrained ones (PPO-Lag $0.82 \to 0.28 \to 0.24$, CPC $0.82 \to 0.32 \to 0.18$); on regional, CPC's probe violation falls most between visits ($0.62 \to 0.40$) while PPO+PC's rises ($0.72 \to 0.87$). Over all revisits CPC's F_C is negative with an interval excluding zero on ER (-0.11, $[-0.15, -0.06]$) and BA (-0.08, $[-0.15, -0.00]$) and negative on WS. In Fig. 4(b) all four clouds are centred on $F_R \approx 0$ and differ along F_C. Retaining the repair behaviour and retaining the safe behaviour are therefore different quantities, and CPC improves most on the second across visits.

Switch-aligned violations (C2). Figure 6 (Appendix C) aligns the violation rate to switches. Re-entering a seen mode produces no spike for any learner. Entering the late-arriving betweenness mode raises the violation rate of every learner not already at a ceiling by 0.25–0.35 within about 80 steps; by $t_s + 150$ CPC is back near its pre-switch level (0.40 against 0.30) while PPO-Lag and PPO remain elevated. Consolidation has a cost on the novel mode: the betweenness probe AUC of PPO+PC is highest on WS and BA (0.51, 0.53), and CPC's (0.40–0.43) is above PPO-Lag's (0.33–0.40) in every family and within 0.06 of PPO's: the cascade slows adaptation relative to consolidation alone, but not below the constrained-only learner.

Service at matched risk (C3). Figure 5(a) places the learners between the two myopic rules: reward-greedy serves 0.95 at a violation rate of about 0.55, safety-greedy 0.35 at 0.12–0.27; every learner is far below the reward-greedy frontier in service (a measure of lookahead, not of the trade-off). The ordering on violations is the same in every family (CPC, PPO-Lag, PPO, PPO+PC); the unconstrained learners buy 0.1–0.2 more service with a violation rate near 1 and an absorbed lifetime. Moving the budget d from 0.02 to 0.10 moves PPO-Lag along a line (ER: violation $0.40 \to 0.60$, service $0.42 \to 0.49$, 9 of 20 lifetimes absorbed) while CPC barely moves ($0.34 \to 0.32$, $0.41 \to 0.44$, 1 of 20): the behaviour is held by the cascade rather than by λ. C3 holds in its frontier form: CPC does not resolve the reward–safety trade-off (on ER it gives up 0.02 of service against PPO-Lag) but moves the reward–safety–forgetting frontier and removes the absorbing state that every other learner enters.

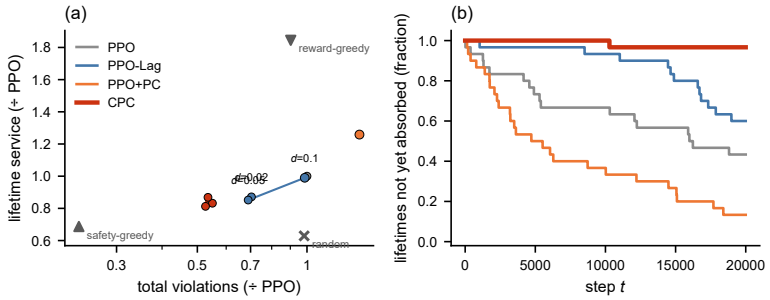


Figure 5: Reward–risk frontier and absorbing states, three families pooled. (a) Lifetime service against total violations, both normalised to PPO (log x); marker size $\propto F_R + F_C$; PPO-Lag and CPC traced over $d \in \{0.02, 0.05, 0.10\}$ on ER (CPC's three points overlap); grey markers are the one-step rules. (b) Fraction of the 30 lifetimes per learner ($d = 0.05$) not yet in an absorbing violation state.

Order, cascade, and a boundary-aware reference. The results do not depend on the Hard order (Appendix F): on the Easy and Random orders PPO is absorbed in 7 and 10 of 10 ER lifetimes, PPO+PC in 7 and 7, PPO-Lag in 2 and 2, CPC in 1 and 2; reward forgetting stays within ± 0.03 for every learner, and CPC keeps the lowest violation rate and a negative F_C on both. The cascade's benefit is not monotone in depth or timescale ratio (Appendix G): at $K \in \{2, 7\}$ or $\omega \in \{2, 8\}$ CPC is absorbed in 2–3 of 10 or 2 of 5 lifetimes and its violation rate rises to 0.39–0.49, so $K = 4, \omega = 4$ is the best cell we tested; the multiplier step and the cost shaping matter less. The Safe EWC of Coursey et al. (2026), given the true switch times that no other learner sees, is absorbed in 3–4 of 10 lifetimes at a violation rate of 0.40–0.47: boundary knowledge and cost shaping do not remove the absorbing state in this setting, whereas the constrained cascade does. Over the budget sweep on all three families (Table 4) CPC is absorbed in 5 of 70 lifetimes, PPO-Lag in 26.

6 Discussion, Limitations and Conclusion

The method combines two existing components; the main contribution is the problem setting and the diagnosis. On a non-resetting graph the learners do not forget repairs; they differ in whether the service reward leads them into a state that later learning does not leave. Episodic safe RL removes this state by resetting (Eysenbach et al., 2018); without a reset, errors persist, and consolidation, usually regarded as protective, preserves them as well. Limitations: every learner serves far below the one-step reward-greedy rule, and a learner with lookahead is untested; every mode recurs within a few segments here, and rarer recurrences may forget more; budget-1 unrecoverability separates absorbing from transient violations but not safe from unsafe policies, and larger B is exponential to certify; longer streams will meet plasticity loss (Dohare et al., 2024; Tang et al., 2025); synthetic 50-node graphs omit the physics and sparse topology of real grids.

In summary, on a non-resetting graph the unconstrained learners are led by the service reward into an absorbing violation state; consolidation alone stores this behaviour and a constraint alone delays it. Reward forgetting is close to zero for all learners while safety forgetting separates them, and CPC, which combines the two components, avoids the absorbing state in 29 of 30 lifetimes without task boundaries.

AI Use Statement

The research question, the choice of setting, the hypotheses C1–C3 and every design decision in this paper were made by the authors, who direct the work and are responsible for all of its claims. Under the authors' direction, an LLM-based assistant was used for tasks with required disclosure: refining the wording of the hypotheses and their pre-specified readings proposed by the authors, giving feedback on the experimental protocol (stream construction, calibration, tuning-stream selection, evaluation metrics) designed by the authors, implementing the environment, the learners and the evaluation code to the authors' specification, and assisting in interpreting the results. We have

not used generative AI tools to generate data (all data are produced by the simulator described in Section 5); developing theoretical models, formulating mathematical claims and writing proofs are not applicable to this work. Additionally, we used the same tool for tasks with recommended disclosure: drafting and editing the text, producing the figures and tables from the run logs, identifying and summarising related literature, and formatting references. We have reviewed all AI-assisted work: every design choice was required to trace to a published source, which the authors checked against its full text; the code was tested by the authors and every reported number was regenerated from the run logs by the released scripts; every citation was verified against the cited work. The authors take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

Ethics Statement

All experiments run on synthetic graphs produced by the simulator of Section 5; no human subjects, personal data or proprietary datasets are involved. The failure modes are standard models from the network-robustness literature, random and degree-targeted removal (Albert et al., 2000), betweenness-targeted removal (Holme et al., 2002) and localised outages (Artime et al., 2024), applied to 50-node synthetic graphs, and the paper studies how a network is repaired, not how it is attacked; we see no dual-use risk beyond that literature. The learners are evaluated in simulation only. Deploying a learning controller on real infrastructure would require guarantees that this work does not provide, and our results on absorbing violation states are a reason for caution rather than a licence to deploy; we state this as a limitation in Section 6.

Reproducibility Statement

The environment (stream, failure modes including the betweenness attack, capacity ceiling, reward and safety cost, budget-B oracle), the four learners, the Safe-EWC reference and the probe evaluator are described in Sections 3–5 and Appendices A, E and G; every hyperparameter, the tuning-stream selection and the threshold calibration are in Appendix E (Tables 2, 3); the stream orders and seeds are given in Section 5 and Appendix F; the pre-registered readings are in Appendix B. Code, run logs and the scripts that produce every table and figure from the logs will be released.

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, pp. 50377–50407, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/9d8cf1247786d6dfeefeeb53b8b5f6d7-Abstract-Conference.html.

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 22–31, 2017.

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, pp. 29304–29320, 2021.

Hongjoon Ahn, Jinu Hyeon, Youngmin Oh, Bosun Hwang, and Taesup Moon. Prevalence of negative transfer in continual reinforcement learning: Analyses and a simple baseline. In *International Conference on Learning Representations*, 2025.

Réka Albert, Hawoong Jeong, and Albert-László Barabási. Error and attack tolerance of complex networks. *Nature*, 406:378–382, 2000.

Eitan Altman. *Constrained Markov Decision Processes*, volume 7 of *Stochastic Modeling Series*. Chapman and Hall/CRC, Boca Raton, FL, 1999. ISBN 978-0-8493-0382-1.

Oriol Artime, Marco Grassia, Manlio De Domenico, James P. Gleeson, Hernán A. Makse, Giuseppe Mangioni, Matjaž Perc, and Filippo Radicchi. Robustness and resilience of complex networks. *Nature Reviews Physics*, 6:114–131, 2024.

Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286 (5439):509–512, 1999.

Marcus K. Benna and Stefano Fusi. Computational principles of synaptic memory consolidation. *Nature Neuroscience*, 19:1697–1706, 2016.

Haitham Bou Ammar, Rasul Tutunov, and Eric Eaton. Safe policy search for lifelong reinforcement learning with sublinear regret. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 2361–2369, 2015.

Sergey V. Buldyrev, Roni Parshani, Gerald Paul, H. Eugene Stanley, and Shlomo Havlin. Catastrophic cascade of failures in interdependent networks. *Nature*, 464:1025–1028, 2010.

Massimo Caccia, Jonas Mueller, Taesup Kim, Laurent Charlin, and Rasool Fakoor. Task-agnostic continual reinforcement learning: Gaining insights and overcoming challenges. In *Conference on Lifelong Learning Agents*, pp. 89–119, 2023.

Austin Coursey, Marcos Quinones-Grueiro, and Gautam Biswas. On the design of safe continual RL methods for control of nonlinear systems. In *European Control Conference*, pp. 892–897, 2025. doi: 10.23919/ECC65951.2025.11187149.

Austin Coursey, Abel Diaz-Gonzalez, Marcos Quinones-Grueiro, and Gautam Biswas. Safe continual reinforcement learning in non-stationary environments. *arXiv preprint arXiv:2604.19737*, 2026.

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*, 632:768–774, 2024.

Paul Erdős and Alfréd Rényi. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5:17–61, 1960.

Benjamin Eysenbach, Shixiang Gu, Julian Ibarz, and Sergey Levine. Leave no trace: Learning to reset for safe and autonomous reinforcement learning. In *International Conference on Learning Representations*, 2018.

Changjun Fan, Li Zeng, Yizhou Sun, and Yang-Yu Liu. Finding key players in complex networks through deep reinforcement learning. *Nature Machine Intelligence*, 2:317–324, 2020.

Haotian Fu, Jiayu Yao, Omer Gottesman, Finale Doshi-Velez, and George Konidaris. Performance bounds for model and policy transfer in hidden-parameter MDPs. In *International Conference on Learning Representations*, 2023.

Milan Ganai, Zheng Gong, Chenning Yu, Sylvia Herbert, and Sicun Gao. Iterative reachability estimation for safe reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 69764–69797, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/dca63f2650fe9e88956c1b68440b8ee9-Abstract-Conference.html.

Nathan Grinsztajn, Johan Ferret, Olivier Pietquin, Philippe Preux, and Matthieu Geist. There is no turning back: A self-supervised approach for reversibility-aware reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 1898–1911, 2021.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, volume 30, pp. 1024–1034, 2017.

Petter Holme, Beom Jun Kim, Chang No Yoon, and Seung Kee Han. Attack vulnerability of complex networks. *Physical Review E*, 65(5):056109, 2002. doi: 10.1103/PhysRevE.65.056109.

Ashish Kumar Jayant and Shalabh Bhatnagar. Model-based safe deep reinforcement learning via a constrained proximal policy optimization algorithm. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24432–24445, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/9a8eb202c060b7d81f5889631cbcd47e-Abstract-Conference.html.

Jiaming Ji, Borong Zhang, Jiayi Zhou, Xuehai Pan, Weidong Huang, Ruiyang Sun, Yiran Geng, Yifan Zhong, Juntao Dai, and Yaodong Yang. Safety-Gymnasium: A unified safe reinforcement learning benchmark. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 36, pp. 18964–18993, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/3c557a3d6a48cc99444f85e924c66753-Abstract-Datasets_and_Benchmarks.html.

Christos Kaplanis, Murray Shanahan, and Claudia Clopath. Continual reinforcement learning with complex synapses. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 2497–2506, 2018.

Christos Kaplanis, Murray Shanahan, and Claudia Clopath. Policy consolidation for continual reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 3242–3251, 2019.

Elias B. Khalil, Hanjun Dai, Yuyu Zhang, Bistra Dilkina, and Le Song. Learning combinatorial optimization algorithms over graphs. In *Advances in Neural Information Processing Systems*, pp. 6348–6358, 2017.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Vito Latora and Massimo Marchiori. Efficient behavior of small-world networks. *Physical Review Letters*, 87(19):198701, 2001. doi: 10.1103/PhysRevLett.87.198701.

Zuxin Liu, Zhepeng Cen, Vladislav Isenbaev, Wei Liu, Zhiwei Steven Wu, Bo Li, and Ding Zhao. Constrained variational policy optimization for safe reinforcement learning. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 13644–13668, 2022.

Kevin Lu, Aditya Grover, Pieter Abbeel, and Igor Mordatch. Reset-free lifelong learning with skill-space planning. In *International Conference on Learning Representations*, 2021.

Clare Lyle, Zeyu Zheng, Evgenii Nikishin, Bernardo Avila Pires, Razvan Pascanu, and Will Dabney. Understanding plasticity in neural networks. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 23190–23211, 2023.

Antonio Majdandzic, Boris Podobnik, Sergey V. Buldyrev, Dror Y. Kenett, Shlomo Havlin, and H. Eugene Stanley. Spontaneous recovery in dynamical networks. *Nature Physics*, 10(1):34–38, 2014. doi: 10.1038/nphys2819.

Eli A. Meirom, Haggai Maron, Shie Mannor, and Gal Chechik. Controlling graph dynamics with reinforcement learning and graph neural networks. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 7565–7577, 2021.

Golnaz Mesbahi, Parham Mohammad Panahi, Olya Mastikhina, Steven Tang, Martha White, and Adam White. Position: Lifetime tuning is incompatible with continual reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 81786–81796, 2025.

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron C. Courville. The primacy bias in deep reinforcement learning. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 16828–16847, 2022.

Benjamin Plaut, Juan Liévano-Karim, Hanlin Zhu, and Stuart Russell. Safe learning under irreversible dynamics via asking for help. *Journal of Machine Learning Research*, 27(188):1–43, 2026. URL https://www.jmlr.org/papers/v27/25-2248.html.

Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. Technical report, OpenAI, 2019. URL https://cdn.openai.com/safexp-short.pdf.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P. Lillicrap, and Gregory Wayne. Experience replay for continual learning. In *Advances in Neural Information Processing Systems*, pp. 348–358, 2019.

Hillel Sanhedrai, Jianxi Gao, Amir Bashan, Moshe Schwartz, Shlomo Havlin, and Baruch Barzel. Reviving a failed network through microscopic interventions. *Nature Physics*, 18(3):338–349, 2022. doi: 10.1038/s41567-021-01474-y.

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

Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & Compress: A scalable framework for continual learning. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 4528–4537, 2018.

Zitao Song, Wendi Ren, and Shuang Li. Amortized network intervention to steer the excitatory point processes. In *International Conference on Learning Representations*, 2024.

Hongyao Tang, Johan S. Obando-Ceron, Pablo Samuel Castro, Aaron C. Courville, and Glen Berseth. Mitigating plasticity loss in continual reinforcement learning by reducing churn. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 58883–58904, 2025.

Atsushi Tero, Seiji Takagi, Tetsu Saigusa, Kentaro Ito, Dan P. Bebber, Mark D. Fricker, Kenji Yumiki, Ryo Kobayashi, and Toshiyuki Nakagaki. Rules for biologically inspired adaptive network design. *Science*, 327(5964):439–442, 2010. doi: 10.1126/science.1177894.

Akifumi Wachi and Yanan Sui. Safe reinforcement learning in constrained Markov decision processes. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 9797–9806, 2020.

Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

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

Maciej Wołczyk, Michał Zając, Razvan Pascanu, Łukasz Kuciński, and Piotr Miłoś. Disentangling transfer in continual reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 6304–6317, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/2938ad0434a6506b125d8adaff084a4a-Abstract-Conference.html.

Annie Xie, James Harrison, and Chelsea Finn. Deep reinforcement learning amidst continual structured non-stationarity. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 11393–11403, 2021.

Nuoya Xiong, Yihan Du, and Longbo Huang. Provably safe reinforcement learning with stepwise violation constraints. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/aa3e67220ca4cd50010165c950fc8056-Abstract-Conference.html.

Deunsol Yoon, Sunghoon Hong, Byung-Jun Lee, and Kee-Eung Kim. Winning the L2RPN challenge: Power grid management via semi-Markov afterstate actor-critic. In *International Conference on Learning Representations*, 2021.

Dongjie Yu, Haitong Ma, Shengbo Eben Li, and Jianyu Chen. Reachability constrained reinforcement learning. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 25636–25655, 2022.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 3987–3995, 2017.

A ALGORITHM

Algorithm 1 CPC on one lifetime stream (no resets, no task identity)

Initialise $\pi_1, \ldots, \pi_K$ identically; $\lambda \leftarrow 0$; empty window buffer $\mathcal{D}$
for $t = 1, \ldots, T$ **do**
  $a_t \sim \pi_1(\cdot \mid G_t)$ over absent edges; execute; observe r_t, c_t, G_{t+1} ▷ mode m_t hidden
  Append to $\mathcal{D}$
  if $|\mathcal{D}| = W$ **then**
    GAE for reward and cost separately $\rightarrow \hat{A}_R, \hat{A}_C$; $\bar{c}_W \leftarrow$ mean cost in $\mathcal{D}$
    $\lambda \leftarrow \max(0, \lambda + \alpha_\lambda(\bar{c}_W - d))$
    Freeze $\pi_k^{\text{old}} \leftarrow \pi_k$ for all k
    for E epochs of minibatches **do**
      Minimise $\mathcal{L}_{\text{PG}} + \mathcal{L}_{\text{self}} + \mathcal{L}_{\text{casc}} + c_v\mathcal{L}_V - c_e\mathcal{H}$ Eqs. (4)–(6)
    end for
    Clear $\mathcal{D}$
  end if
  Every Δ steps: freeze a copy of π_1 and run offline probes on fixed graphs under each mode ▷ evaluation only
end for

B PRE-SPECIFIED READINGS AND OUTCOMES

The reading of each figure was fixed before the runs; we list each reading with the measured outcome. *C1*: PPO+PC's violations and absorbed lifetimes exceed those of PPO-Lag and CPC, with non-overlapping intervals; F_R lies within ± 0.03 of zero for every learner, so the learners separate on the cost side, not the reward side (Section 5). *C2*: F_R and F_C are not correlated (every method has low F_R; PPO+PC has positive, CPC negative F_C); post-switch violation exceeds the lifetime mean after the novel mode and not after revisits. *C3*: CPC is not worse than the better of PPO-Lag and PPO+PC on either forgetting quantity in any family and has fewer violations in every family; at matched violations it is not better in service on ER, which is the frontier form of the reading: CPC moves the reward–safety–forgetting frontier and does not resolve the reward–safety trade-off.

C SWITCH-ALIGNED VIOLATION RATE

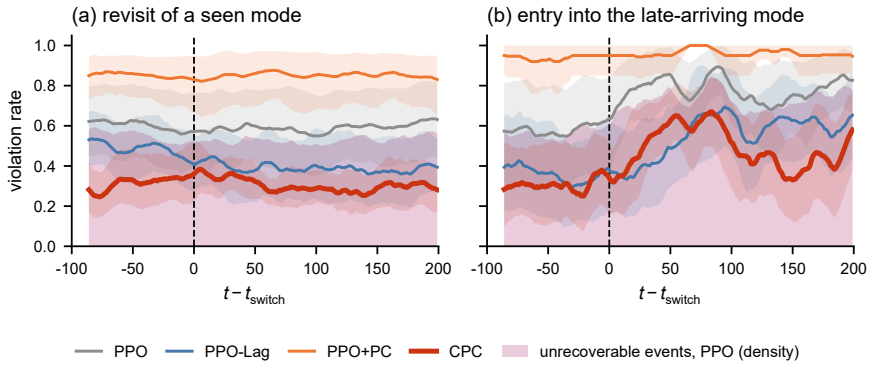


Figure 6: Violation rate aligned to mode switches (ER, Hard), $t - t_s \in [-100, 200]$: (a) re-entry into a seen mode, (b) entry into the late-arriving betweenness mode; shaded overlay is the density of unrecovered violations under PPO. Mean and 95% CI over switches and 10 seeds.

D MODE-PAIR INTERFERENCE MATRIX AND STREAM CONSTRUCTION

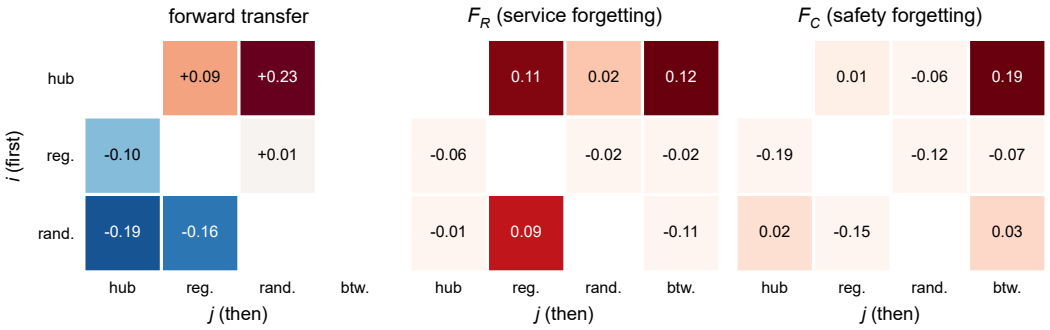


Figure 7: Mode-pair interference from $i \to j \to i$ runs with PPO: forward transfer, F_R and F_C of i caused by j. Five seeds, $L = 3000$ steps per segment, ER graphs; entries are means, F_R in service units. Hub is the only mode PPO acquires within a segment; what it acquires transfers forward (row 1, FT $= 0.09, 0.23$) but is overwritten by regional and betweenness ($F_R = 0.11, 0.12$) but not by random (0.02); betweenness also induces safety forgetting on hub ($F_C = 0.19$). The held-out mode is never trained first, so it appears only as a column. The Hard order therefore interleaves hub with the other modes (Appendix E).

E SENSITIVITY AND CALIBRATION

Calibration of thresholds and strike intensity. Table 2 reports, for cells of the calibration grid on the tuning stream (two graph seeds, 1000 steps, hub→regional→random, greedy policies scoring 120 sampled candidate edges per step), the violation rate of a random, a one-step reward-greedy and a one-step safety-greedy policy, together with their service levels. The frozen cell is one in which safety is satisfiable ($\leq 5\%$ for safety-greedy) but costly (15–40% for reward-greedy) and the threat is real ($\geq 30\%$ for random).

Table 2: Calibration grid, $\tau_\lambda = 0.30$ (violation rate; service W in parentheses for the two greedy policies). Frozen cell in bold.

p	b	τ_g	random	reward-greedy	safety-greedy
0.05	10	0.90	0.03	0.02 (0.95)	0.00 (0.46)
0.05	10	0.96	0.32	0.21 (0.95)	0.02 (0.46)
0.06	13	0.90	0.15	0.13 (0.95)	0.02 (0.36)
0.06	13	0.96	0.39	0.35 (0.95)	0.05 (0.39)
0.08	10	0.90	0.19	0.07 (0.95)	0.02 (0.43)
0.08	**10**	**0.96**	**0.41**	**0.22 (0.95)**	**0.03 (0.43)**
0.08	13	0.96	0.60	0.52 (0.94)	0.14 (0.37)
0.08	16	0.96	0.77	0.63 (0.93)	0.25 (0.35)

Why $p\,b < 1$ and a shaped training cost. The agent adds one edge per step, so $|E_t|$ is a random walk with drift $1 - p\,b$ reflected at the capacity $|E_0|$ and at the damage floor. At $p\,b = 1$ (our first frozen cell, $p = 0.10$) the drift is zero and over 15000 steps the walk reaches the floor ($\text{std} \approx \sqrt{Tp}\,b \approx 390 \gg |E_0|$) and stays there; a graph with half its edges has $|\text{GCC}|/N \ll \tau_g$, so every policy, constrained or not, ended in a permanent violation in 6 of 12 pilot lifetimes. This is a property of the environment, not of the learner. Once there, the indicator cost is constant, its advantage is identically zero, and the Lagrangian has no gradient to climb back with. We therefore (i) chose $p\,b = 0.8$, so that violations come from the location of a strike and $|E_t|$ recovers within tens of steps, and (ii) train the cost critic on $\tilde{c}_t = c_t + \max(0, \frac{\tau_g - |\text{GCC}|/N}{\tau_g}) + \max(0, \frac{\tau_\lambda - \lambda_2/\lambda_2(G_0)}{\tau_\lambda})$, which is non-constant inside a violation. The budget d, the multiplier update and all reported costs

use c_t. Under a gentler per-edge failure process (Poisson removal with mean 2 per step under a 25% floor) no policy that repaired at all ever violated, which is why bursts are used.

Probe length. A frozen policy trained by PPO on a single mode for 6000 steps, cross-evaluated on every mode with 300-step probes (probe service / violation): trained on hub gives $0.458/0.33$ on hub, $0.520/0.56$ on regional, $0.578/0.62$ on random and $0.550/0.82$ on betweenness; trained on regional gives $0.294/0.71$ on hub; trained on random $0.420/0.52$ on hub; untrained $0.289/0.63$ on hub. The modes therefore do require different policies, asymmetrically: hub-trained policies transfer, regional- and random-trained ones lose hub service and double hub violations. With 60-step probes all of these collapse into 0.34–0.42, because the first half of a probe is dominated by the undamaged initial graph; this is why we probe for 200 steps and score the second half.

Segment length and the interference matrix. On the pilot $i \to j \to i$ streams of Figure 7 ($L = 3000$ per segment, i.e., 23 updates with a 128-step window; 5 seeds), unconstrained PPO raises its hub probe service from 0.30 to 0.48 within one segment, so a segment of 1500–3000 steps is long enough for something to be learned and hence forgotten. What is forgotten depends on what follows: after a regional or betweenness segment the hub probe falls back to 0.37 (about 60% of the gain lost, 90% bootstrap intervals excluding zero), after a random segment only to 0.46. Random failure teaches no location and does not overwrite the hub repair. Safety forgetting appears on the same pair: the hub probe violation rate rises from 0.66 to 0.85 after the betweenness segment. Regional and random are acquired much more slowly by the unconstrained learner ($+0.05$ and $+0.00$ in a segment, with in-stream violation rates above 0.9), so their rows of the matrix carry little signal; this is why the raw drop, not a ratio to the gain, is the primary forgetting measure.

Learner tuning on the tuning stream. Table 3 reports the last block of a 12000-step stationary hub stream and a 300-step probe on hub graphs. The upper panel (difference reward $W(G_{t+1}) - W(G_t)$, window 256, $\gamma = 0.99$, 3 seeds) shows two failures. (i) With $\alpha_\lambda = 1$, λ saturates at $\lambda_{\max}$ within ~ 40 updates and the surrogate becomes $-\hat{A}_C$: the policy stops learning service. (ii) With the $\omega_{1,2} = 4$ of the main text of Kaplanis et al. (2019), the consolidation cascade pins the visible policy to its uniform shadow; PPO+PC does not learn at all, and rescaling every KL term by $\log |A|$ does not rescue it. The lower panel (level reward, window 128, 5 seeds) is the regime used in the paper: with $\gamma = 0.99$ or a 64-step window the unconstrained learners collapse, with $\gamma = 0.95$ they learn to serve the eight demand pairs at the price of a violation rate that climbs to 1, while the constrained learners hold the violation rate at 0.1–0.4 and λ below saturation. The difference reward gives no signal for *staying* at a high level, which is why it was replaced.

Cost budget. Table 4 gives the Hard-stream results behind the traces in Fig. 5(a) and their WS/BA counterparts. On ER the constrained-only learner follows its budget while the consolidated constrained learner does not move. On WS and BA (5 seeds per off-budget cell) the picture is the same with one exception: over the nine cells PPO-Lag is absorbed in 26 of 70 lifetimes and CPC in 5, and CPC's one weak cell is WS at $d = 0.02$ (2 of 5, $F_C = +0.08$).

Capacity ceiling and rejected actions. Because $|E_t|$ drifts back to the capacity $|E_0|$ within tens of steps after a strike, every policy has 22–24% of its actions rejected as no-ops at capacity, independent of the policy (no-op: 0). The ceiling therefore does not differentiate the agents.

Budget-1 unrecoverability inside and outside absorbing runs. Table 5 evaluates c_t^{irr} ($B = 1$, exact enumeration) on every violated step of the main runs. For every policy, learned or not, roughly three violated steps in four cannot be repaired by one edge: a ten-edge burst that cuts three nodes off rarely leaves a single bridging edge that restores $|\mathrm{GCC}| \geq \tau_g N$. Inside an absorbing run the fraction is 0.95–1.00. The rate of the event therefore does not rank policies, but its conditional value does distinguish the violations a policy leaves from the one it does not.

F STREAM ORDERS

All three orders keep the Hard multiset (hub $\times 3$, regional $\times 2$, random $\times 2$, betweenness $\times 2$), the same per-seed segment lengths and the same graphs. *Easy* = hub, random, hub, random, hub,

Table 3: Tuning-stream results (stationary hub, $T = 12000$). In-stream values are the last 2000-step block; probe values are second-half service / violation of a 300-step frozen probe on three hub graphs. Upper panel: difference reward, window 256, $\gamma = 0.99$, 3 seeds. Lower panel: level reward, window 128, 5 seeds.

Variant	in-stream $\bar{W}$	in-stream viol.	λ_{end}	probe P/C
Difference reward, window 256, $\gamma = 0.99$				
PPO	0.690	1.00	–	0.479 / 0.47
PPO-Lag, $\alpha_\lambda = 1, \lambda_{\max} = 10$	0.342	0.25	9.7	0.335 / 0.39
PPO-Lag, $\alpha_\lambda = 0.25, \lambda_{\max} = 5$	0.415	0.25	3.2	0.415 / 0.23
PPO-Lag, $\alpha_\lambda = 0.5, \lambda_{\max} = 5, d = 0.10$	0.529	0.55	4.3	0.319 / 0.45
PPO+PC, $\omega_{1,2} = 4$	0.380	0.38	–	0.381 / 0.60
PPO+PC, $\omega_{1,2} = 1$	0.422	0.58	–	0.620 / 0.37
PPO+PC, $\omega_{1,2} = 0.1$	0.560	0.94	–	0.492 / 0.44
CPC, $\omega_{1,2} = 1$	0.381	0.20	2.3	0.372 / 0.33
CPC, $\omega_{1,2} = 0.3$	0.394	0.19	3.0	0.410 / 0.35
CPC, $\omega_{1,2} = 0.1$	0.462	0.19	2.4	0.479 / 0.23
Level reward, window 128 (64 where stated)				
PPO, window 64, $\gamma = 0.99$	0.245	0.94	–	0.399 / 0.48
PPO, $\gamma = 0.95$	0.497	0.83	–	0.477 / 0.67
PPO, $\gamma = 0.90$	0.558	0.75	–	0.331 / 0.69
PPO-Lag, $\gamma = 0.95, \alpha_\lambda = 0.1$ (used)	0.456	0.38	2.7	0.381 / 0.21
PPO-Lag, $\gamma = 0.95, \alpha_\lambda = 0.05$	0.533	0.67	3.0	0.447 / 0.54
PPO-Lag, window 64, $\gamma = 0.95, \alpha_\lambda = 0.1$	0.400	0.69	5.0	0.344 / 0.35
PPO+PC, $\gamma = 0.95$	0.631	1.00	–	0.502 / 0.93
PPO+PC, window 64, KL$/\log\lvert A\rvert$, $\omega_{1,2} = 4$	0.196	0.67	–	0.306 / 0.89
CPC, $\gamma = 0.95, \alpha_\lambda = 0.1$ (used)	0.465	0.10	1.3	0.439 / 0.10
CPC, $\gamma = 0.95, \alpha_\lambda = 0.05$	0.455	0.09	1.3	0.449 / 0.30
CPC, $\gamma = 0.90, \alpha_\lambda = 0.1$	0.490	0.15	2.2	0.458 / 0.36
CPC, window 64, KL$/\log\lvert A\rvert$, $\omega_{1,2} = 4$	0.178	0.67	4.0	0.143 / 0.67

Table 4: Cost-budget sweep on Hard streams, $T = 20000$ (IQM; 10 seeds per ER cell and for $d = 0.05$, 5 seeds for the WS/BA cells at $d \in \{0.02, 0.10\}$). Absorbed as count/n.

Family	Agent	d	$\bar{W}$	Violation rate	F_C	Absorbed
ER	PPO-Lag	0.02	0.421	0.396	−0.033	4/10
ER	PPO-Lag	0.05	0.448	0.354	−0.056	1/10
ER	PPO-Lag	0.10	0.486	0.596	+0.044	5/10
ER	CPC	0.02	0.406	0.342	−0.078	1/10
ER	CPC	0.05	0.425	0.302	−0.113	0/10
ER	CPC	0.10	0.442	0.324	−0.067	0/10
WS	PPO-Lag	0.02	0.466	0.363	−0.116	0/5
WS	PPO-Lag	0.05	0.409	0.444	−0.029	5/10
WS	PPO-Lag	0.10	0.399	0.459	−0.028	1/5
WS	CPC	0.02	0.436	0.420	0.084	2/5
WS	CPC	0.05	0.456	0.297	−0.060	1/10
WS	CPC	0.10	0.426	0.251	−0.073	0/5
BA	PPO-Lag	0.02	0.423	0.566	−0.065	2/5
BA	PPO-Lag	0.05	0.355	0.556	−0.052	6/10
BA	PPO-Lag	0.10	0.534	0.597	0.017	2/5
BA	CPC	0.02	0.463	0.373	−0.017	0/5
BA	CPC	0.05	0.408	0.504	−0.075	0/10
BA	CPC	0.10	0.436	0.449	−0.012	1/5

regional, betweenness, regional, betweenness: hub is adjacent only to random (the pair with the smallest interference in Fig. 7, $F_R = +0.02$) and never to betweenness (the pair with the largest safety interference, $F_C = +0.19$), and the held-out mode still arrives in the last third of life. *Random* is a per-seed permutation with no repeated neighbours and a visible first segment; the held-out mode may then arrive early, so the novel-mode AUC is not comparable and is omitted. Table 6 shows

Table 5: Fraction of violated steps that are budget-1 unrecoverable, Hard streams, $d = 0.05$, means over 10 seeds; "in" = steps inside an absorbing run, "out" = violated steps outside one (no-op is absorbed within 70 steps in every lifetime; one-step rules other than no-op are never absorbed).

	ER		WS		BA	
Agent	in	out	in	out	in	out
no-op	1.00	–	1.00	–	1.00	–
random	–	0.75	–	0.73	–	0.78
reward-greedy	–	0.75	–	0.73	–	0.77
safety-greedy	–	0.76	–	0.73	–	0.88
PPO	1.00	0.81	0.98	0.83	1.00	0.78
PPO+PC	0.99	0.75	1.00	0.79	1.00	0.77
PPO-Lag	0.95	0.78	1.00	0.77	0.98	0.80
CPC	–	0.68	1.00	0.68	–	0.79

that the absorbing state is a property of the learner, not of the order: over the three orders PPO is absorbed in 21 of 30 ER lifetimes, PPO+PC in 23, PPO-Lag in 5 and CPC in 3. Easy is not easier for the unconstrained learners—keeping hub away from betweenness does not keep the reward from fragmenting the graph—and Random, where the held-out mode can arrive first, is hardest for everyone (PPO $10/10$). Reward forgetting is within ± 0.03 on every order for every learner. CPC has the lowest violation rate on every order (0.30, 0.35, 0.38 against PPO-Lag's 0.35, 0.38, 0.45) and a negative F_C on every order (-0.11, -0.08, -0.07; the Easy interval excludes zero), and is the only learner with both properties; on Random its absorption count ties PPO-Lag's ($2/10$) and its service is 0.03 lower.

Table 6: Stream orders, ER, $T = 20000$, $d = 0.05$, 10 seeds per cell: IQM with 95% bootstrap intervals; Absorbed as count of lifetimes.

Order	Agent	$\bar{W}\uparrow$	Violation rate↓	Absorbed↓	$F_R\downarrow$	$F_C\downarrow$	Novel↑
Hard	PPO	0.541 [0.40,0.64]	0.586 [0.34,0.86]	4/10	−0.018 [−0.08,0.03]	−0.039 [−0.13,0.05]	0.461
Hard	PPO+PC	0.651 [0.58,0.71]	0.904 [0.75,0.97]	9/10	0.022 [−0.03,0.06]	0.054 [−0.03,0.09]	0.392
Hard	PPO-Lag	0.448 [0.39,0.49]	0.354 [0.30,0.53]	1/10	0.007 [−0.02,0.03]	−0.056 [−0.13,0.04]	0.397
Hard	CPC	0.425 [0.38,0.46]	0.302 [0.24,0.39]	0/10	−0.036 [−0.09,0.02]	−0.113 [−0.15,−0.06]	0.403
Easy	PPO	0.485 [0.31,0.64]	0.850 [0.58,0.97]	7/10	−0.006 [−0.02,0.01]	−0.047 [−0.11,0.03]	0.340
Easy	PPO+PC	0.575 [0.50,0.69]	0.735 [0.44,0.97]	7/10	−0.002 [−0.02,0.02]	−0.023 [−0.05,0.02]	0.424
Easy	PPO-Lag	0.404 [0.37,0.43]	0.382 [0.29,0.45]	2/10	−0.003 [−0.02,0.01]	−0.013 [−0.06,0.03]	0.317
Easy	CPC	0.412 [0.38,0.46]	0.351 [0.28,0.42]	1/10	0.014 [−0.00,0.05]	−0.078 [−0.13,−0.02]	0.362
Random	PPO	0.554 [0.48,0.66]	0.948 [0.84,0.98]	10/10	−0.025 [−0.06,0.02]	−0.028 [−0.09,0.03]	
Random	PPO+PC	0.612 [0.51,0.68]	0.760 [0.57,0.91]	7/10	−0.021 [−0.06,0.02]	−0.030 [−0.08,0.05]	
Random	PPO-Lag	0.426 [0.37,0.51]	0.446 [0.35,0.64]	2/10	0.000 [−0.02,0.03]	−0.024 [−0.07,0.06]	
Random	CPC	0.391 [0.36,0.43]	0.381 [0.32,0.47]	2/10	0.011 [−0.02,0.03]	−0.074 [−0.16,0.06]	

G Ablations and a boundary-aware reference

Table 7 varies the parts of CPC on the ER Hard streams of Table 1. *Cascade length.* With $K = 1$ (PPO-Lag), 2, 4 and 7 the absorbed counts are $1, 2, 0, 3$ of 10, the violation rates $0.35, 0.39, 0.30, 0.49$ and F_C $-0.06, -0.02, -0.11, -0.09$. The benefit is not monotone: two policies give the cascade little to consolidate into, and at $\omega = 4$ the last shadow of a seven-policy cascade has a self trust-region weight $\beta\omega^6 \approx 2000$, never leaves the initial policy, and pulls the cascade toward its uniform start. *Timescale ratio.* $\omega = 2$ and $\omega = 8$ are each absorbed in 2 of 5 lifetimes (F_C $+0.08$ and -0.04). Our pre-fixed $K = 4$ with the $\omega = 4$ of Kaplanis et al. (2019) (whose cascade has eight policies) is therefore the best cell we tested, not an arbitrary one; the removal of the absorbing state is sensitive to them, which we list as a limitation. *Multiplier and cost critic.* A five-fold larger multiplier step ($\alpha_\lambda = 0.5$: $1/5$ absorbed, violation 0.32, service 0.48) and an unshaped cost critic ($1/5$, 0.42) change less: the cascade rather than the multiplier rule carries the effect, and the shaped training cost of Appendix E helps recovery but is not what prevents absorption. *Boundary-aware reference.* Safe EWC with the $\beta = 5$ of Coursey et al. (2026) is absorbed in 4 of 10 lifetimes (violation 0.40, service 0.42, F_C -0.04), with $\beta = 1$ in 3 of 10 (0.47, 0.42,

0.00). Given the boundaries, EWC protects whatever the cost-shaped reward has taught, and on a non-resetting graph that includes the road into the absorbing state; its violation rate is above PPO-Lag's and CPC's on the same streams and its service equals CPC's. $\lambda_{\mathrm{EWC}} = 10$ is of the order of the values used by Coursey et al. (2026) (11–23) and was not tuned on our stream.

Table 7: Ablations of CPC on ER Hard streams ($T = 20000$, $d = 0.05$; 10 seeds for K, 5 for the rest) and the boundary-aware Safe-EWC reference of Coursey et al. (2026) (cost-shaped reward $r - \beta c$, non-online EWC with $\lambda_{\mathrm{EWC}} = 10$, Fisher estimated on the last window before each *true* switch, which no other learner sees; 10 seeds). IQM with 95% intervals; Absorbed as count/n.

Variant	$\bar{W} \uparrow$	Violation rate↓	Absorbed↓	$F_R \downarrow$	$F_C \downarrow$	Novel↑
PPO-Lag ($K = 1$)	0.448 [0.39,0.49]	0.354 [0.30,0.53]	1/10	0.007 [−0.02,0.03]	−0.056 [−0.13,0.04]	0.397
CPC, $K = 2$	0.406 [0.35,0.48]	0.394 [0.30,0.56]	2/10	0.025 [−0.01,0.06]	−0.015 [−0.08,0.09]	0.322
CPC, $K = 4$ (used)	0.425 [0.38,0.46]	0.302 [0.24,0.39]	0/10	−0.036 [−0.09,0.02]	−0.113 [−0.15,−0.06]	0.403
CPC, $K = 7$	0.433 [0.37,0.51]	0.486 [0.38,0.73]	3/10	0.013 [−0.00,0.03]	−0.091 [−0.16,−0.03]	0.346
CPC, $\omega = 2$	0.466 [0.38,0.56]	0.405 [0.25,0.65]	2/5	−0.026 [−0.04,−0.01]	0.084 [−0.20,0.19]	0.417
CPC, $\omega = 8$	0.409 [0.38,0.48]	0.447 [0.42,0.65]	2/5	0.003 [−0.06,0.05]	−0.038 [−0.16,0.08]	0.354
CPC, $\alpha_\lambda = 0.5$	0.483 [0.44,0.53]	0.320 [0.17,0.46]	1/5	0.010 [−0.05,0.04]	−0.057 [−0.14,0.00]	0.433
CPC, unshaped cost critic	0.438 [0.40,0.60]	0.420 [0.28,0.81]	1/5	0.020 [−0.01,0.05]	−0.073 [−0.16,0.01]	0.358
Safe-EWC (oracle boundaries), $\beta = 5$	0.418 [0.37,0.51]	0.395 [0.31,0.54]	4/10	0.003 [−0.03,0.04]	−0.039 [−0.10,0.03]	0.339
Safe-EWC (oracle boundaries), $\beta = 1$	0.417 [0.35,0.60]	0.469 [0.29,0.80]	3/10	−0.005 [−0.02,0.01]	0.000 [−0.04,0.05]	0.351

Not run. Neighbours of the frozen threshold cell (τ_g, τ_λ), $B = 2$, $p = 0.15$ and a PID multiplier were pre-registered but not run before submission. In Table 3 the two cells in which CPC does not learn are the two in which PPO+PC does not learn either.

H WHY REPLAY IS NOT A DROP-IN BASELINE

CLEAR (Rolnick et al., 2019) assumes a stored transition remains a legal sample of the current MDP. On a non-resetting graph the stored edge may already exist, or its endpoints may no longer lie in different components; replayed tuples would then prescribe masked or useless actions. Any replay baseline must re-encode current graphs and replay mode-conditional advantages, which is a different method.