CONTINUAL REINFORCEMENT LEARNING WITH NEUROEVOLUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite many studies about causes and remedies of plasticity loss in Reinforcement Learning (RL) under continual task changes, no RL method has yet consistently achieved a good balance between adaptation and forgetting. Here we turn to an alternative optimization paradigm, neuroevolution (NE): algorithms, such as evolution strategies (ES) and genetic algorithms (GAs), that search directly in weight space through mutation and selection over a population of neural networks. We compare ES and a GA against state-of-the-art continual RL variants and population-based RL, with no method informed of a task switch, across different kinds of environments and environmental changes and policies ranging from a few hundred parameters to million-parameter networks. We find that ES most consistently achieves a good stability-plasticity trade-off, while the GA is the most plastic method. To explain this, we study the return landscape around each method's solutions. We observe that ES finds the widest neighborhoods, regions of weight space in which perturbed policies still solve the task, and that the size of the overlap between the neighborhoods of consecutive tasks correlates with the stability-plasticity trade-off of a method. The GA has narrower neighborhoods and moves further per task, so it is more plastic but forgets more. Rewarding behavioral diversity in a GA through novelty search makes the population even more plastic at the cost of forgetting. Finally, symptoms of plasticity loss commonly reported in RL do not transfer to NE suggesting that they are specific to gradient-based optimization rather than non-stationarity. Overall, these results establish NE as a competitive alternative to RL under continual task changes, and suggest that training under perturbations in weight space may be a useful mechanism for continual learning more broadly.

1 INTRODUCTION

Reinforcement learning (RL) has achieved remarkable successes, ranging from superhuman gameplay (Silver et al., 2016) to the control of high-dimensional robots (OpenAI et al., 2019) and the fine-tuning of large language models (Ouyang et al., 2022). Yet modern RL remains *static*: policies are trained offline with task-specific hyperparameters, so they fail under even minor task variations at deployment (Kudithipudi et al., 2022; Chollet, 2019), or forget past skills when adapted to new tasks (Abel et al., 2023; Luo et al., 2025).

Continual RL advances a contrasting vision: agents that adapt online in non-stationary environments, without external tuning or human supervision (Pan et al., 2025; Abel et al., 2023). It requires balancing two competing objectives: retaining prior skills (stability) and acquiring new ones (plasticity) (Khetarpal et al., 2022; Pan et al., 2025). State-of-the-art policy gradient methods such as Proximal Policy Optimization (PPO) (Schulman et al., 2017) lose plasticity when the task changes (Abel et al., 2023; Aleixo et al., 2023; Klein et al., 2024). A growing body of work identifies various causes of plasticity loss and proposes restoring it by modifying the optimizer (Muppidi et al., 2024), the loss (Tang et al., 2025) or the training procedure (Sokar et al., 2023). However, recent analyses show that symptoms are hard to tell apart from causes (Lyle et al., 2024), and that the type of task and of variation strongly affects how methods perform (Hiratani, 2024; Wołczyk et al., 2021). The search for methods with a good stability-plasticity trade-off in RL is thus ongoing.

We here turn to an alternative optimization paradigm: neuroevolution (NE), the evolutionary optimization of neural networks (Stanley et al., 2019; Risi et al., 2025). NE belongs to the family

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.

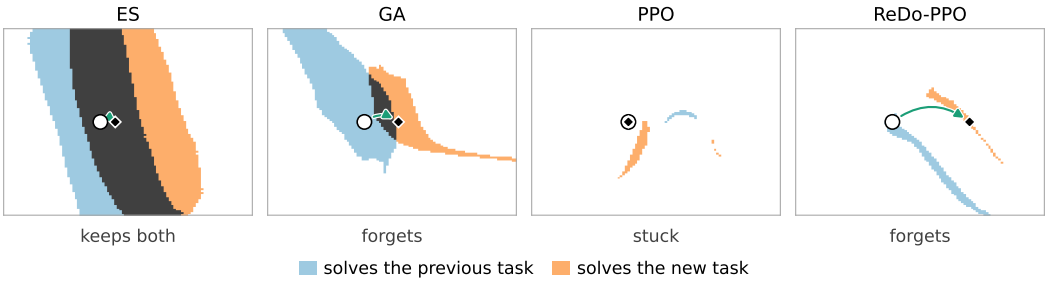

Figure 1: Balancing stability and plasticity with neuroevolution and reinforcement learning. Each panel shows one method's return landscape, the return of the policies whose weights lie around its solution. The circle marks the solution before the task switch and the diamond the solution after it. Its neighborhood, the region of weights whose policies still solve the task, is blue for the task before the switch and orange for the task after it; dark grey is their overlap, where a policy solves both tasks. ES sits inside the overlap, so it learns the new task and keeps the old one. The GA's overlap is smaller and at the switch the solution escapes it. PPO has lost plasticity: it lies in a tiny neighborhood of the previous task and does not move at the switch. ReDo-PPO, which recycles dormant neurons to counter plasticity loss (Sokar et al., 2023), learns the new task, but its neighborhoods remain narrow and do not overlap.

of evolutionary algorithms, black-box optimizers that iteratively refine a population of candidate solutions through mutation and selection (Mitchell, 1996; Risi et al., 2025). We compare two NE algorithms, evolution strategies (ES) (Salimans et al., 2017) and genetic algorithms (GAs) (Such et al., 2018), against PPO, three state-of-the-art variants for continual RL (Tang et al., 2025; Muppidi et al., 2024; Sokar et al., 2023) and, to isolate the advantage of a population, population-based RL (Jaderberg et al., 2017). Across a range of environments and types of change, with policies from a few hundred to more than a million parameters, we find that the GA is the most plastic method and that ES most consistently achieves a good trade-off between stability and plasticity. PPO, in contrast, loses plasticity, and continual RL variants restore some of it at the expense of stability.

To understand these differences, we look at the return landscape around the solutions each method finds, the map from policy weights to return (Rahn et al., 2023; Lehman et al., 2018). We perturb the weights of a solution and call the region in which the perturbed policies still solve the task its *neighborhood*. Lehman et al. (2018) showed that ES finds wider neighborhoods than gradient-based optimization, because it maximizes the expected return of perturbed copies of its solution. Here we further propose that neighborhood size matters for continual learning: when the neighborhoods of two consecutive tasks overlap, a solution to the new task lies close to the current one, so at a task switch a method can reach it while still solving the old task, and be both stable and plastic (Figure 1). We find that ES has the widest neighborhoods and the largest overlap between consecutive tasks, and that the size of this overlap predicts each method's trade-off.

Why does the GA learn new tasks better than ES but forget more? We hypothesize that both follow from how the GA selects solutions: where ES averages its perturbed copies into a single centroid, the GA keeps its best individuals and mutates them. We show that its population is therefore far more spread out in weight space than that of ES, and that its neighborhoods are narrower, since its selection does not reward robustness to perturbations. This spread is what lets GAs escape local optima (Such et al., 2018): at a task switch, a member far from the current solution may already solve the new task, and selection lets it take over the population. Finally, we show that rewarding behavioural diversity through novelty search (Bahlous-Boldi et al., 2025) makes the GA more plastic, as it becomes more likely that some individual already solves the new task.

Finally, NE does not show symptoms that accompany plasticity loss in RL: dormant neurons (Sokar et al., 2023) do not accumulate, and weight drift (Muppidi et al., 2024) does not cost it plasticity. This suggests that these symptoms stem from gradient-based optimisation rather than from non-stationarity.

In summary, we make the following observations:

- ES achieves a good stability-plasticity trade-off most consistently, and the GA is the most plastic method.

- ES finds the widest neighborhoods, and the share of a method's neighborhood that solves both consecutive tasks correlates with its trade-off.
- Behavioural diversity makes the best individual more plastic by raising the chance that some member of the population solves the new task.
- Dormant neurons do not accumulate in NE, and weight drift does not cost it plasticity.

We release a codebase that enables reproducing all our experimets: `https://anonymous.4open.science/r/cont_ne_anon-33C7`.

2 BACKGROUND

2.1 CONTINUAL REINFORCEMENT LEARNING

Following previous work (Ahn et al., 2025; Muppidi et al., 2024), we define continual RL as a sequence of T RL tasks indexed by $\tau \in \{1, \dots, T\}$. At task τ the agent interacts with a Markov Decision Process (MDP) $\mathcal{M}_\tau = (\mathcal{S}_\tau, \mathcal{A}_\tau, p_\tau, r_\tau)$ of states, actions, transition probability $p_\tau(s_{t+1} \mid s_t, a_t)$ and reward $r_\tau(s_t, a_t)$. An episode starts from $s_0 \sim \rho_\tau$ and terminates after H_τ steps or upon a terminal condition, and the agent maximizes the episodic return of each task, $R_\tau = \sum_{t=0}^{H_\tau} r_\tau(s_t, a_t)$. We adopt a boundary-free setting: no information about task change is provided.

2.2 NEUROEVOLUTION

Neuroevolution optimises the weights of a neural network with an evolutionary algorithm (Risi et al., 2025; Stanley et al., 2019), using the episodic return R_τ as the fitness of each candidate. We consider a generic NE loop over the parameters $\theta \in \mathbb{R}^d$ of a policy π_θ (Lange, 2023). At generation g the algorithm holds a search state Ω_g: `ask()` produces P candidates from it, each candidate is evaluated episodically to obtain its fitness, and `tell()` updates the state using these fitnesses. We study one population-based and one distribution-based method (Gallagher & Frean, 2005); they differ in what the state carries from one generation to the next and in how selection maps fitnesses onto it. A *genetic algorithm* (GA) is population-based: Ω_g is an archive of N genomes, `ask()` perturbs parents drawn uniformly from it with isotropic Gaussian noise of scale σ, and `tell()` keeps the fittest N of archive and offspring (Such et al., 2018). *Evolution strategies* (ES) are distribution-based: Ω_g is the mean θ of an isotropic Gaussian $\mathcal{N}(\theta, \sigma^2 I)$ of fixed width. `ask()` samples P candidates from it, and `tell()` moves θ towards the fitter ones, along a Monte Carlo estimate of the gradient of the expected return under this distribution (Wierstra et al., 2008; Salimans et al., 2017). Pseudocode and further details for both methods are given in Appendix A.2 (Algorithms 1 and 2).

3 RELATED WORK

Loss of plasticity in deep RL. Policy-gradient methods produce today's strongest results in domains ranging from robotic locomotion (Schulman et al., 2017) to LLM fine-tuning (Shenfeld et al., 2026). When the task changes over time, however, they lose plasticity: a network trained on a sequence of tasks learns each new task less well than a freshly initialized one (Klein et al., 2024; Abel et al., 2023; Ahn et al., 2025; Juliani & Ash, 2024). Plasticity plays an important role even within a single task, since RL is inherently non-stationary: the agent's observations and targets depend on its own policy (Igl et al., 2021). Several causes have been proposed for this loss (Lyle et al., 2024). Neural networks are prone to forgetting earlier knowledge, as updates may overwrite useful weights (McCloskey & Cohen, 1989). Gradient-based optimization accumulates *dormant* neurons, units whose activation is near zero on every input and which therefore receive no gradient (Sokar et al., 2023). It also lets the weights drift away from their initialization, where the network is most plastic (Muppidi et al., 2024). An update computed on some inputs also changes the network's outputs on all others, a phenomenon called *churn*, which grows as the representation collapses and gradients on different inputs align (Tang et al., 2025). Whether these are symptoms or causes of plasticity loss is contested: they co-occur and no single one accounts for the loss (Lyle et al., 2024). We benchmark three continual RL variants that address plasticity loss: ReDo, which periodically re-initializes dormant neurons (Sokar et al., 2023); TRAC, a parameter-free optimizer that adaptively pulls the weights back towards their initialization (Muppidi et al., 2024); and C-CHAIN, which targets the collapse of the representation by reducing churn (Tang et al., 2025).

Population-based RL. Population-based training (PBT; Jaderberg et al., 2017) trains a population of gradient-based learners in parallel and periodically replaces the weakest members with copies of the strongest. We include PBT-PPO to isolate the advantages of a population. To our knowledge, PBT has not been studied in continual settings.

Evolution in changing environments. Evolutionary biology and artificial evolution have long studied how environmental change shapes evolution (Grove, 2014; Nisioti & Moulin-Frier, 2022; Lehman & Miikkulainen, 2015). In these fields change is often seen as a driver rather than a setback. Extinction events can free a population from local optima (Lehman & Miikkulainen, 2015), and a history of change leaves a population more diverse and better prepared for the next change (Grove, 2014; Nisioti & Moulin-Frier, 2022). These studies used either very small networks or much simpler representations, and measured only how well a population adapts but not whether it forgets. Quality-Diversity optimization searches for a diverse set of well-performing solutions rather than a single one, measuring diversity in the space of a *behavioural descriptor*, a low-dimensional summary of what a policy does (Pugh et al., 2016). This diversity makes a robot robust to damage during deployment (Cully et al., 2015) and lets NE outcompete RL by escaping local optima (Chalumeau et al., 2023). However, it has not been studied in continual settings, where the archive may need to adapt online. Similarly to Quality-Diversity, Nisioti et al. (2025) link population diversity to plasticity. They compared NE to RL on transferring skills in a curriculum of tasks and found that RL frequently collapsed at task changes while NE methods reached higher levels of the curriculum. However, they did not consider the trade-off with stability, and they studied toy tasks without comparing to continual RL variants. Unlike these studies, we measure stability, plasticity and population diversity on a range of RL benchmarks with policies of a few hundred to more than a million parameters, compare NE to continual RL methods, and propose a mechanism behind NE's advantage.

Neighborhoods in continual loss landscapes Studying evolution in switching environments, Wang & Dai (2019) asked whether a hill climber that alternates between two landscapes can find a local optimum shared by both. They showed that this is possible through *basin linking*: if the optimum reached on one task lies within a neighborhood of the other, the next switch carries the climber there, and repeated switches chain neighborhoods together until it reaches an optimum shared by both tasks. We recover this behaviour with neural policies on RL tasks and show that it depends on the width of the neighborhoods and on the selection scheme. In gradient-based continual learning, Mirzadeh et al. (2020) linked wider neighborhoods to less forgetting in supervised learning and widened them through the training regime, such as the learning rate and the batch size; here we instead study neighborhood width in RL tasks and focus on how selection under noise increases neighborhood size and affects the stability-plasticity trade-off.

4 Experimental set-up

Evaluation metrics All metrics are computed from the return R_τ (Section 2.1), averaged over several episodes over multiple environment seeds and deterministic actions. For the population-based methods (ES, the GA and PBT-PPO) we evaluate the *centroid*, the agent whose parameters are the mean of the population's parameters.

For plasticity, we report the **learning accuracy (LA)**, the return on each task at the end of training on it, and the **cumulative return (Cum.)** on the current task over training, which also captures how quickly a method recovers after a switch. For stability, we report **forgetting (F)**, the return lost on a task between the end of training on it and the end of the run. We score the stability-plasticity trade-off as LA $-$ F, as is common in continual learning (Lopez-Paz & Ranzato, 2017; Díaz-Rodríguez et al., 2018). Finally, **zero-shot transfer (ZT)** is the return on the next task at a switch, before any training on it.

Unless stated otherwise, results report the mean over ten trials with a 95% percentile-bootstrap confidence interval, and we test differences within a setting with a one-sided Mann-Whitney U test with Holm's correction ($p < 0.05$). To put settings on one scale we rescale LA and F per setting so that 0 is the untrained network and 1 the best method's mean LA.

Neighborhood width The *neighborhood* of a solution is the region of weight space around it in which the policy still solves its task. We say that a policy solves a task when its rescaled return is

at least 0.5. To measure the size of the neighborhood we follow Lehman et al. (2018). We take the centroid at the end of a task and add to each of its weight tensors a random Gaussian direction whose length is a fraction ϵ of that tensor's norm, so that the measure does not depend on the scale of the weights (Li et al., 2018). The *neighborhood width* is the radius ϵ at which half of the perturbed policies no longer solve the task.

A wide neighborhood helps at a switch only if it also contains solutions of the next task. The *shared neighborhood* measures this: we perturb the centroid in the same way at a single radius, $\epsilon = 0.1$, and report the fraction of perturbed policies that solve both the current and the next task.

In Appendix H.1 we show that these metrics are robust to different values of these threshold choices.

Population diversity For methods that keep a population of policies we measure how different its members are, both in behavior and weight space. Behavioural diversity is how often two members choose different actions in the same states (for continuous actions, how far apart their actions are). Genomic diversity is the average distance between the weights of two members, normalised by the number of weights so that networks of different sizes can be compared (see Appendix F for the exact definitions).

Environments and task sequences We consider classic control (CartPole, Acrobot, MountainCar; Lange, 2026), sparse-reward navigation (MiniGrid empty rooms; Chevalier-Boisvert et al., 2023; Nikulin et al., 2024), locomotion (Brax HalfCheetah; Freeman et al., 2021) and pixel-based physics puzzles requiring million-parameter convolutional networks (Kinetix; Matthews et al., 2025). To study hard exploration we also consider DeepSea (Osband et al., 2020). To introduce non-stationarity we vary tasks in one of three ways. *Observation offsets* add a fixed random vector to the observation (Muppidi et al., 2024). *Changes in the transition dynamics* rescale a physical parameter of the agent or environment, change the room size in MiniGrid, or load a new Kinetix level, where both the robot's morphology and the optimal behavior vary widely across tasks. *Action reversal* reverses the order of the discrete actions, or flips the sign of continuous ones; on DeepSea a task instead redraws which action moves right in each cell. Every run lasts twenty tasks, in one of two settings, a 20-task setting and a *two-task* setting, where two tasks alternate ten times. The two settings differ in how much forgetting can be avoided: in a sequence, a single policy can at best stay close to the solutions of neighbouring tasks, so tasks further back are hard not to forget, whereas with two tasks a single policy can show no forgetting (see Appendix A.1 for every environment and task sequence).

Algorithms For **NE** we benchmark ES (Salimans et al., 2017) and a GA (Such et al., 2018) (Section 2.2). Unlike the centroid of ES, which is the point the population is sampled around, the centroid of the GA is the mean of its archive and not one of its members, so nothing ensures that it tracks the best solution. On MountainCar and Kinetix it does not: the archive holds several unrelated solutions, as a GA's population can spread over several optima of equal fitness (Mahfoud, 1995). On these two environments we therefore add a heuristic rule utilising solely current task fitnesses that narrows the GA's search while its centroid scores below its archive (see Appendix A.2.1 for the rule). To study the role of population diversity we add Dominated Novelty Search (Bahlous-Boldi et al., 2025), which we call GA + Novelty. It keeps the GA but selects on a *competition fitness*, an individual's mean distance in descriptor space to its nearest fitter individuals, so a weak individual survives if no fitter one behaves like it. Its descriptors are learned online with AURORA (Cully, 2019; Grillotti & Cully, 2022), as the latent code of an auto-encoder trained on the observations a policy visits (see Appendix A.2.1 for details). As **RL baselines** we benchmark PPO (Schulman et al., 2017) and three continual variants: TRAC-PPO (Muppidi et al., 2024), ReDo-PPO (Sokar et al., 2023) and C-CHAIN (Tang et al., 2025). To isolate what a population contributes, we also benchmark population-based PPO (PBT-PPO; Jaderberg et al., 2017).

Appendix A.2 gives the pseudocode and hyperparameters of every method. All methods are tuned to solve the stationary version of every environment (see Appendix B for their performance without task switches). To check that our conclusions do not depend on one set of hyperparameters, we vary the number of gradient steps of PPO and the search width of the GA and ES (see Appendix D), and the frequency of task switching (see Appendix E).

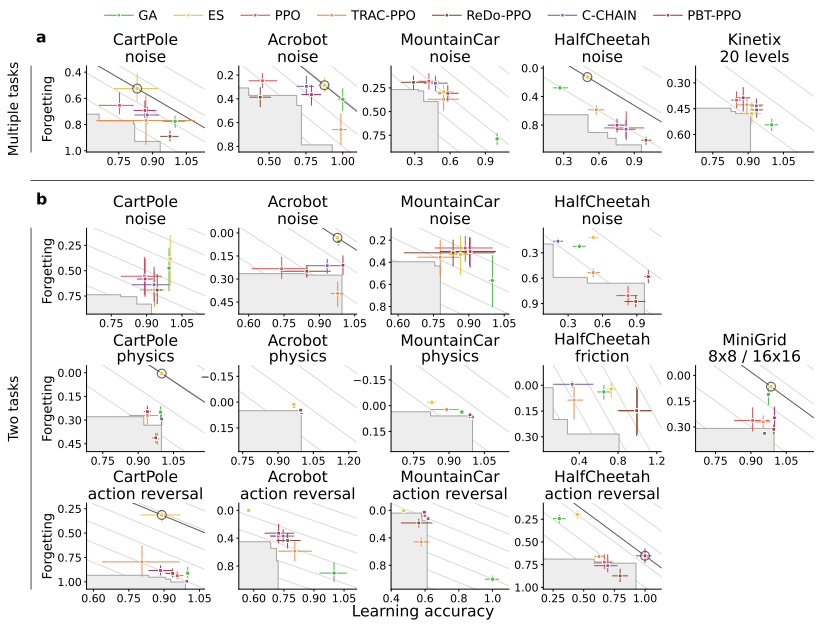

Figure 2: Stability-plasticity trade-off. Learning accuracy (LA, higher is better) against forgetting (F, lower is better) a) with a sequence of twenty tasks and b) with two alternating tasks. Grey lines are equal LA − F, and the grey region lies beyond the 95% interval of some RL method on both axes. A ring marks the best method on the stability-plasticity trade-off (LA-F). GA is the most plastic method and ES the one with the best trade-off.

5 Results

ES achieves the best stability-plasticity trade-off. Figure 2 shows plasticity (LA) against stability (F) for every method and setting (see Appendix C.1 for the return over training and every metric in every setting). In each setting, we circle the method with the highest LA − F if it significantly outperforms every method of the other family. A method that learns only one of two alternating tasks is not eligible, since it forgets nothing.[1] ES is circled in seven of the eighteen settings and PBT-PPO in one, HalfCheetah under action reversal. ES falls short in two cases. Under action reversal on Acrobot and MountainCar, it learns only one ordering of the actions, because at the switch the fitness variance of its population is close to zero (see Appendix F for the fitness variance over training). On HalfCheetah, the RL methods reach higher returns than ES and the GA by the end of each task, as they do without task switches (see Appendix B), so the gap is not a failure of neuroevolution in the continual setting. They also forget more, so ES still has the best trade-off over a sequence of tasks, but not with two alternating tasks.

The GA is the most plastic method. The GA has the highest learning accuracy in eleven of the eighteen settings (Figure 2). Under action reversal on Acrobot and MountainCar it is the only method that fully learns both action orderings: ES freezes on the first ordering it solves in both, and PPO, C-CHAIN and PBT-PPO do so on MountainCar. The GA forgets more than ES in every setting, and under action reversal it forgets almost all it learns. Whether GA or RL forgets more depends on the task: GA forgets more in five settings (MountainCar under noise, Acrobot and MountainCar under action reversal, and Kinetix), where it also learns the most, but less than every RL method in six, among them HalfCheetah under action reversal and under noise over a sequence of tasks, and MiniGrid.

ES has the highest zero-shot transfer. ES has the highest zero-shot transfer in nine of the fourteen settings, significantly above every RL method in six (Figure 3). The GA transfers better than every RL method in four settings (significantly on MiniGrid and CartPole under action reversal), but under action reversal on Acrobot and MountainCar it falls to the return of an untrained policy, having

[1] Under action reversal: ES on Acrobot and MountainCar, and PPO, C-CHAIN and PBT-PPO on MountainCar.

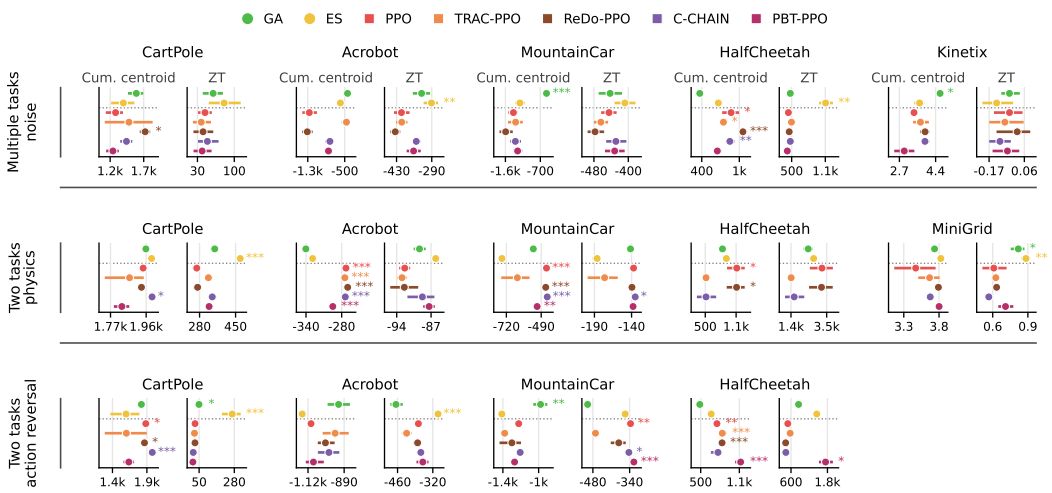

Figure 3: Return and zero-shot transfer. Cumulative return of the centroid (Cum.) and its zero-shot transfer to the next task (ZT), with a sequence of tasks (top) and with two alternating tasks. Mean and 95% bootstrap interval over trials.

specialised to the ordering it just learned. The RL methods transfer less, and on HalfCheetah under noise and CartPole under reversal they relearn each task after every switch. No method transfers to an unseen Kinetix level (see Appendix C.1 for every setting).

Continual RL variants improve plasticity at the expense of stability. TRAC-PPO, ReDo-PPO and C-CHAIN keep the network plastic: C-CHAIN learns each task better than PPO in fourteen of the eighteen settings, ReDo-PPO in nine and TRAC-PPO in eight (Figure 2). Each of the three forgets more than PPO in more than half of the settings (Figure 1; see Appendix H.5 for the return landscapes of every method). The best RL method on the trade-off is PBT-PPO: it has the best LA − F among RL methods, in eleven settings. Its advantage comes neither from diversity, as its population is the least behaviourally diverse of the three population methods on six of the eleven tasks, including those where it keeps the most among RL methods (see Appendix F and Figure 5), nor from averaging weights or a more stochastic policy (see Appendix H.4 for these tests). We hypothesize that it comes from selection at initialisation: PBT-PPO starts from eight independently initialised members, its first exploit steps keep those whose return is highest this favours solutions in wider neighborhoods, about twice as wide as PPO's (Figure 4). Its lead shrinks over training, however, as its members become copies of one another (see Appendix H.1 for its width over training), so we expect its advantage to fade in longer continual settings. Training a population thus helps RL keep what it learns, but still leaves it behind ES.

ES sits in the widest neighborhood, and the shared neighborhood predicts the trade-off. ES has the widest neighborhood of all methods, wider than PPO's in every setting, followed by the GA and PBT-PPO (Figure 4, left; see Appendix H.2 for the test per environment).Appendix H.3). The shared neighborhood (Section 4) predicts LA − F, across methods and within each family .

Novelty search makes populations more plastic. The GA also keeps a diverse population, which can help it escape a local optimum after a switch or raise the chance that some individual already lies in the neighborhood of the new task. We test this link between diversity and plasticity by rewarding diversity explicitly through novelty search (Bahlous-Boldi et al., 2025). We choose three settings where a switch can trap the whole population in a local optimum (described in Appendix A.1): MountainCar, whose sparse reward gives no signal until the car reaches the flag, under stronger observation noise and under action reversal, and DeepSea, where each task changes the action map of the single rewarded path. GA + Novelty keeps a more behaviourally diverse population in all three, and its best individual gains cumulative return in each (Figure 5); on DeepSea it re-finds the new action map after 57% of the switches, against 11% for the GA. The centroid does not track this gain: it averages over members that solve different tasks, and the average of their weights solves none of them. Diversity thus buys plasticity for a method that can pick its best individual, moving the GA towards the plastic end of the trade-off (see Appendix C.3 for more task settings).

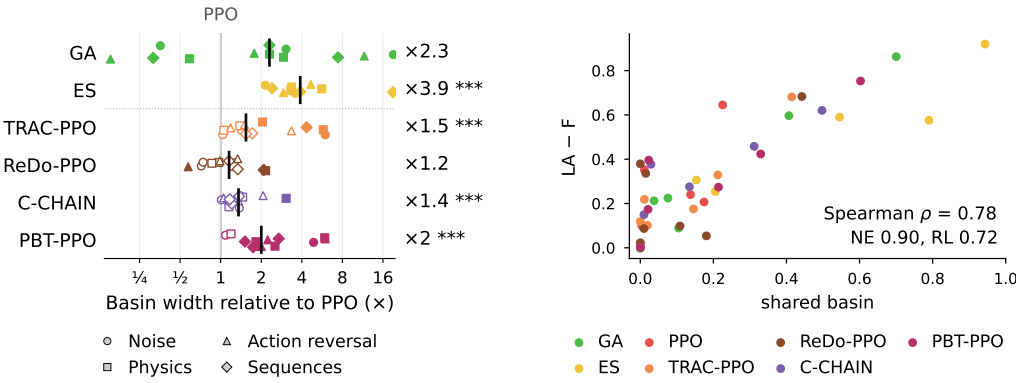

Figure 4: Neighborhood width and the shared neighborhood. Left: neighborhood width relative to PPO, one point per method and classic control or MiniGrid setting; bars are medians and stars a Wilcoxon test over settings. Right: shared neighborhood (random moves of radius 0.1) against LA − F, with the Spearman correlation over all points and within each family. ES exhibits the largest neighborhood and neighborhood overlap correlates with the stability-plasticity trade-off.

NE does not show the symptoms of plasticity loss of RL. PPO's dormant fraction keeps rising over task switches, and ReDo-PPO, which recycles dormant units, keeps it the lowest (Figure 6, top). The dormant fraction of NE depends on the task but does not increase at task switches (see Appendix C.2 for every setting. Weight drift (Muppidi et al., 2024) does not cost NE its plasticity either (Figure 6, bottom). ES and the continual RL variants keep their weights below PPO's. The GA's weights grow significantly in some tasks, yet it remains the most plastic method.

6 Discussion

We compared two neuroevolution methods, ES and a GA, with PPO and its continual variants across environments and kinds of task change, with matched compute and no signal of a switch. ES achieves the best stability-plasticity trade-off. The GA is the most plastic method in almost every environment, and rewarding novelty increases its plasticity further. PPO loses plasticity in most environments, and its continual variants restore it at the cost of stability. Interestingly, PBT-PPO outperforms variants explicitly designed for continual learning, suggesting that selection within a population, even one that only copies weights, brings RL part of the benefit of NE. Neuroevolution does not show the symptoms of plasticity loss reported in continual RL: its networks do not accumulate dormant units, and the GA's weights drift further than PPO's without costing it plasticity, which suggests that these symptoms stem from gradient-based optimisation rather than from the non-stationarity. Our results suggest that exploring in parameter space biases learning towards wide neighborhoods, which are more likely to overlap with the neighborhood of the next task: across methods, this shared neighborhood predicts the stability-plasticity trade-off. This builds on evidence that wider minima forget less (Mirzadeh et al., 2020), that ES prefers robust solutions (Lehman et al., 2018) and that alternating tasks can link neighborhoods (Wang & Dai, 2019). To the best of our knowledge, we are the first to propose parameter-space exploration as a mechanism for this trade-off.

Our work does not propose that a large shared neighborhood is necessary for a good trade-off. We measure the neighborhood with random moves of a fixed radius, which is how NE explores: every member of an ES or GA population perturbs the centroid. PPO instead follows the gradient of the return (Lehman et al., 2018), so its neighborhood could be wider along its own path. It is not. Along the updates PPO actually makes, its neighborhood is as narrow as in random directions, and only ReDo-PPO keeps its task further (see Figure 26 in Appendix H.4, which repeats the measure along every method's own updates). This agrees with evidence that gradient-based training settles in narrow minima that forget more (Mirzadeh et al., 2020), and that the return of RL policies can collapse under small parameter changes (Rahn et al., 2023). Given enough updates, a method may still reach a solution of the next task outside the measured neighborhood. We therefore propose that holding both tasks is easier for NE than for RL. The neighborhood of NE more often contains a

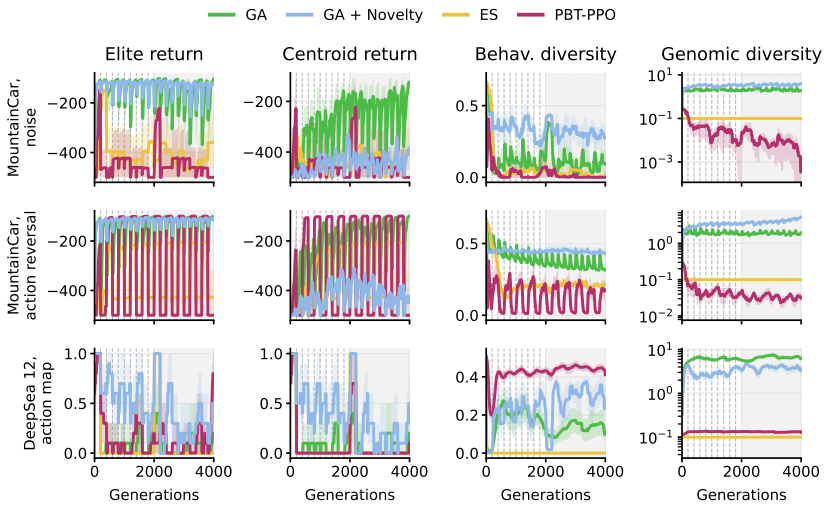

Figure 5: Effect of novelty search. Return of the best individual (elite) and of the centroid, behavioural diversity and genomic diversity; dashed lines mark task switches.

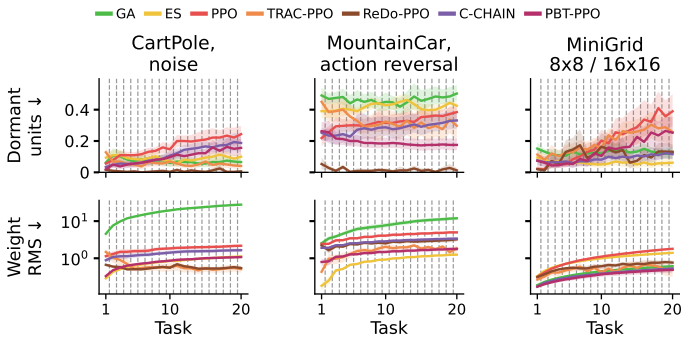

Figure 6: Dormant units and weight drift. Fraction of dormant hidden units of the centroid (top) and root mean square of its weights (bottom). Dashed lines mark task switches. The dormant fraction of NE depends on the task and stays constant; that of RL rises.

solution of both tasks, while the longer path of RL may lose the old task on the way. Whether RL can hold both tasks given enough updates remains open.

Despite these benefits NE may still not outperform RL in every setting. First, it has a higher sample complexity and complex tasks often require large populations (Such et al., 2018). Second, where very finely tuned solutions exist, as in HalfCheetah, NE does not match PPO's stationary performance: gradient steps can climb narrow peaks of the return that ES smooths over (Lehman et al., 2018). Yet recent studies have shown promising scaling for ES: fine-tunes language models with billions of parameters (Qiu et al., 2026; Sarkar et al., 2026), which has been attributed to the low intrinsic dimensionality of these tasks (Liang et al., 2026).

Future work could let ES adapt its search width (Hansen, 2016; Wierstra et al., 2008) and so move along the stability-plasticity trade-off during training. The width should then react to the return, not to a signal of the switch. Parameter-space exploration could also become a method for continual RL. Injecting noise into the policy's weights (Plappert et al., 2018) would let PPO follow the gradient of a smoothed return, as ES does.

AI USE STATEMENT

We used a large language model assistant to edit the text of this paper and to write and debug analysis and plotting code. All code was verified by the authors against the underlying runs, and all claims, numbers and figures were checked by the authors. We take responsibility for the final content of this work.

Reproducibility statement

Appendix A gives every environment, network, method and hyperparameter, and the definition of every metric is in Section 4. An anonymized version of our code, including the scripts that produce every run, table and figure, is available at `https://anonymous.4open.science/r/cont_ne_anon-33C7`.

References

David Abel, André Barreto, Benjamin Van Roy, Doina Precup, Hado van Hasselt, and Satinder Singh. A definition of continual reinforcement learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

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. URL `https://openreview.net/forum?id=KAIqwkB3dT`.

Everton L. Aleixo, Juan G. Colonna, Marco Cristo, and Everlandio Fernandes. Catastrophic Forgetting in Deep Learning: A Comprehensive Taxonomy, December 2023.

Ryan Bahlous-Boldi, Maxence Faldor, Luca Grillotti, Hannah Janmohamed, Lisa Coiffard, Lee Spector, and Antoine Cully. Dominated Novelty Search: Rethinking Local Competition in Quality-Diversity. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '25, pp. 104–112, New York, NY, USA, July 2025. Association for Computing Machinery. ISBN 979-8-4007-1465-8. doi: 10.1145/3712256.3726310.

Felix Chalumeau, Raphael Boige, Bryan Lim, Valentin Macé, Maxime Allard, Arthur Flajolet, Antoine Cully, and Thomas Pierrot. Neuroevolution is a competitive alternative to reinforcement learning for skill discovery. In *International Conference on Learning Representations*, 2023. URL `https://arxiv.org/abs/2210.03516`.

Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo Perez-Vicente, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. In *Advances in Neural Information Processing Systems 36 (Datasets and Benchmarks Track)*, 2023.

François Chollet. On the measure of intelligence. *CoRR*, abs/1911.01547, 2019. URL `http://arxiv.org/abs/1911.01547`.

Antoine Cully. Autonomous skill discovery with quality-diversity and unsupervised descriptors. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 81–89, 2019.

Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. Robots that can adapt like animals. *Nature*, 521(7553):503–507, May 2015. ISSN 1476-4687. doi: 10.1038/nature14422.

Natalia Díaz-Rodríguez, Vincenzo Lomonaco, David Filliat, and Davide Maltoni. Don't forget, there is more than forgetting: New metrics for Continual Learning. arXiv:1810.13166, 2018.

C. Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. Brax – a differentiable physics engine for large scale rigid body simulation. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021. URL `http://github.com/google/brax`.

Marcus Gallagher and Marcus Frean. Population-Based Continuous Optimization, Probabilistic Modelling and Mean Shift. *Evolutionary Computation*, 13(1):29–42, January 2005. ISSN 1063-6560. doi: 10.1162/1063656053583478.

Luca Grillotti and Antoine Cully. Unsupervised behavior discovery with quality-diversity optimization. *IEEE Transactions on Evolutionary Computation*, 26(6):1539–1552, 2022. doi: 10.1109/TEVC.2022.3159855.

Matt Grove. Evolution and dispersal under climatic instability: a simple evolutionary algorithm. *Adaptive Behavior*, 22(4):235–254, August 2014. ISSN 1059-7123, 1741-2633. doi: 10.1177/1059712314533573. URL http://journals.sagepub.com/doi/10.1177/1059712314533573.

Nikolaus Hansen. The cma evolution strategy: A tutorial. *arXiv preprint arXiv:1604.00772*, 2016.

Naoki Hiratani. Disentangling and mitigating the impact of task similarity for continual learning. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

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. URL https://openreview.net/forum?id=Qun8fv4qSby.

Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M. Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, Chrisantha Fernando, and Koray Kavukcuoglu. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*, 2017.

Arthur Juliani and Jordan T. Ash. A Study of Plasticity Loss in On-Policy Deep Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 37, 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, November 2024. URL http://arxiv.org/abs/2411.04832. arXiv:2411.04832 [cs].

Dhireesha Kudithipudi, Mario Aguilar-Simon, Jonathan Babb, Maxim Bazhenov, Douglas Blackiston, Josh Bongard, Andrew P Brna, Suraj Chakravarthi Raja, Nick Cheney, Jeff Clune, et al. Biological underpinnings for lifelong learning machines. *Nature Machine Intelligence*, 4(3):196–210, 2022.

Robert Tjarko Lange. evosax: JAX-based evolution strategies. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation (GECCO '23 Companion)*, pp. 659–662, 2023. doi: 10.1145/3583133.3590733.

Robert Tjarko Lange. gymnax: A JAX-based reinforcement learning environment library. http://github.com/RobertTLange/gymnax, 2026. Version 1.0.0.

Joel Lehman and Risto Miikkulainen. Extinction Events Can Accelerate Evolution. *PLOS ONE*, 10(8):e0132886, August 2015. ISSN 1932-6203. doi: 10.1371/journal.pone.0132886. URL https://dx.plos.org/10.1371/journal.pone.0132886.

Joel Lehman, Jay Chen, Jeff Clune, and Kenneth O. Stanley. ES is more than just a traditional finite-difference approximator. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '18, pp. 450–457, New York, NY, USA, July 2018. Association for Computing Machinery. ISBN 978-1-4503-5618-3. doi: 10.1145/3205455.3205474.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Qiyao Liang, Jinyeop Song, Yizhou Liu, Jeff Gore, Ila Fiete, Risto Miikkulainen, and Xin Qiu. The Blessing of Dimensionality in LLM Fine-tuning: A Variance-Curvature Perspective, January 2026.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient Episodic Memory for Continual Learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An Empirical Study of Catastrophic Forgetting in Large Language Models During Continual Fine-Tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 33:3776–3786, 2025. doi: 10.1109/TASLPRO.2025.3606231.

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. In *Proceedings of the 3rd Conference on Lifelong Learning Agents*, volume 274 of *Proceedings of Machine Learning Research*, pp. 750–783, 2024.

Samir W Mahfoud. *Niching methods for genetic algorithms*. PhD thesis, University of Illinois at Urbana-Champaign, 1995.

Michael Matthews, Michael Beukman, Chris Lu, and Jakob Nicolaus Foerster. Kinetix: Investigating the Training of General Agents through Open-Ended Physics-Based Control Tasks. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=zCxGCdzreM.

Michael McCloskey and Neal J. Cohen. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. In Gordon H. Bower (ed.), *Psychology of Learning and Motivation*, volume 24, pp. 109–165. Academic Press, January 1989. doi: 10.1016/S0079-7421(08)60536-8. URL https://www.sciencedirect.com/science/article/pii/S0079742108605368.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Razvan Pascanu, and Hassan Ghasemzadeh. Understanding the role of training regimes in continual learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, pp. 7308–7320, Red Hook, NY, USA, December 2020. Curran Associates Inc. ISBN 978-1-7138-2954-6.

Melanie Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, 1996.

Aneesh Muppidi, Zhiyu Zhang, and Heng Yang. Fast TRAC: A Parameter-Free Optimizer for Lifelong Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://openreview.net/forum?id=QEaHE4TUgc.

Alexander Nikulin, Vladislav Kurenkov, Ilya Zisman, Artem Agarkov, Viacheslav Sinii, and Sergey Kolesnikov. XLand-MiniGrid: Scalable meta-reinforcement learning environments in JAX. In *Advances in Neural Information Processing Systems 37 (Datasets and Benchmarks Track)*, 2024.

Eleni Nisioti and Clément Moulin-Frier. Plasticity and evolvability under environmental variability: the joint role of fitness-based selection and niche-limited competition. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '22, pp. 113–121, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392372. doi: 10.1145/3512290.3528826. URL https://doi.org/10.1145/3512290.3528826.

Eleni Nisioti, Erwan Plantec, Milton Montero, Joachim Pedersen, and Sebastian Risi. When Does Neuroevolution Outcompete Reinforcement Learning in Transfer Learning Tasks? In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '25, pp. 48–57, New York, NY, USA, July 2025. Association for Computing Machinery. ISBN 979-8-4007-1465-8. doi: 10.1145/3712256.3726475.

OpenAI, Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, Jonas Schneider, Nikolas Tezak, Jerry Tworek, Peter Welinder, Lilian Weng, Qiming Yuan, Wojciech Zaremba, and Lei Zhang. Solving rubik's cube with a robot hand. *CoRR*, abs/1910.07113, 2019. URL http://arxiv.org/abs/1910.07113.

Ian Osband, Yotam Doron, Matteo Hessel, John Aslanides, Eren Sezener, Andre Saraiva, Katrina McKinney, Tor Lattimore, Csaba Szepesvári, Satinder Singh, Benjamin Van Roy, Richard Sutton, David Silver, and Hado van Hasselt. Behaviour suite for reinforcement learning. In *International Conference on Learning Representations*, 2020.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022.

Chaofan Pan, Xin Yang, Yanhua Li, Wei Wei, Tianrui Li, Bo An, and Jiye Liang. A Survey of Continual Reinforcement Learning, June 2025. URL http://arxiv.org/abs/2506.21872. arXiv:2506.21872 [cs].

Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y. Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter space noise for exploration. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=ByBAl2eAZ.

Justin K. Pugh, Lisa B. Soros, and Kenneth O. Stanley. Quality Diversity: A New Frontier for Evolutionary Computation. *Frontiers in Robotics and AI*, 3:40, 2016. doi: 10.3389/frobt.2016.00040.

Xin Qiu, Yulu Gan, Conor F. Hayes, Qiyao Liang, Yinggan Xu, Roberto Dailey, Elliot Meyerson, Babak Hodjat, and Risto Miikkulainen. Evolution Strategies at Scale: LLM Fine-Tuning Beyond Reinforcement Learning. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2509.24372.

Nate Rahn, Pierluca D'Oro, Harley Wiltzer, Pierre-Luc Bacon, and Marc G. Bellemare. Policy optimization in a noisy neighborhood: On return landscapes in continuous control. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, pp. 30618–30640, Red Hook, NY, USA, December 2023. Curran Associates Inc.

Sebastian Risi, Yujin Tang, David Ha, and Risto Miikkulainen. *Neuroevolution: Harnessing Creativity in AI Agent Design*. MIT Press, Cambridge, MA, 2025. URL https://neuroevolutionbook.com/.

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution Strategies as a Scalable Alternative to Reinforcement Learning, September 2017. URL http://arxiv.org/abs/1703.03864. arXiv:1703.03864 [stat].

Bidipta Sarkar, Mattie Fellows, Juan Agustin Duque, Alistair Letcher, Antonio León Villares, Anya Sims, Clarisse Wibault, Dmitry Samsonov, Dylan Cope, Jarek Liesen, Kang Li, Lukas Seier, Theo Wolf, Uljad Berdica, Valentin Mohl, Alexander David Goldie, Aaron Courville, Karin Sevegnani, Shimon Whiteson, and Jakob Nicolaus Foerster. Evolution Strategies at the Hyperscale. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2511.16652.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. RL's Razor: Why Online Reinforcement Learning Forgets Less. In *International Conference on Learning Representations*, 2026.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Vedavyas Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy P Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016. doi: 10.1038/nature16961.

Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. The Dormant Neuron Phenomenon in Deep Reinforcement Learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32145–32168, 2023.

Kenneth O. Stanley, Jeff Clune, Joel Lehman, and Risto Miikkulainen. Designing neural networks through neuroevolution. *Nature Machine Intelligence*, 1(1):24–35, January 2019. ISSN 2522-5839. doi: 10.1038/s42256-018-0006-z. URL https://www.nature.com/articles/s42256-018-0006-z. Publisher: Nature Publishing Group.

Felipe Petroski Such, Vashisht Madhavan, Edoardo Conti, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning, April 2018. URL http://arxiv.org/abs/1712.06567. arXiv:1712.06567 [cs].

Hongyao Tang, Johan Obando-Ceron, Pablo Samuel Castro, Aaron Courville, and Glen Berseth. Mitigating plasticity loss in continual reinforcement learning by reducing churn. In *International Conference on Machine Learning*, 2025. URL https://openreview.net/pdf?id=EkoFXfSauv.

Shenshen Wang and Lei Dai. Evolving generalists in switching rugged landscapes. *PLOS Computational Biology*, 15(10):e1007320, October 2019. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1007320.

Daan Wierstra, Tom Schaul, Jan Peters, and Juergen Schmidhuber. Natural Evolution Strategies. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pp. 3381–3387, Hong Kong, China, June 2008. IEEE. ISBN 978-1-4244-1822-0. doi: 10.1109/CEC.2008.4631255.

Maciej Wołczyk, Michał Zajac, Razvan Pascanu, Łukasz Kuciński, and Piotr Miłoś. Continual world: A robotic benchmark for continual reinforcement learning. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, pp. 28496–28510, Red Hook, NY, USA, December 2021. Curran Associates Inc. ISBN 978-1-7138-4539-3.