TASK DIVERSITY PRODUCES SYSTEMATIC TRANSFER BUT INHIBITS CONTINUAL RL

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual reinforcement learning (RL) aims to produce agents that never stop adapting to new tasks. A key question is how this interacts with the diversity of tasks an agent experiences. Prior work has shown that training on many diverse tasks leads to agents with strong zero-shot and in-context adaptation. However, this work evaluated agents *after* they'd stopped learning, i.e. with frozen weights. How task diversity affects an agent's ability to *continue* learning over a sequence of distribution shifts remains unclear. We introduce Banyan, a GPU-accelerated continual RL domain where one can parametrically control three independent axes that define a task: the map layouts an agent must navigate, the objects it must interact with, and the hierarchical structures of sub-goal dependencies. We find that increasing diversity along each axis induces *systematic transfer*—that is, agents begin training on a new task distribution near the performance attained on the previous one, even when the shift changes the structure of the optimal policy. While increasing diversity improves systematic transfer, we find that *too much* diversity inhibits a learner's ability to continue adapting to new task distributions. As diversity increases, learners plateau in the success rate they achieve on new tasks, yet continue improving on old tasks—even without further exposure to them. We find this phenomenon manifests across continual learning algorithms, memory architectures, architecture sizes, and in Kinetix—a physics-based control domain. We release Banyan as a domain for running controlled experiments that study continual RL in the many-tasks regime.

1 INTRODUCTION

What do we need to develop reinforcement learning (RL) agents that can continuously generalize to new tasks? For much of the history of RL, researchers have focused on developing novel agent components for promoting generalization(Mazzaglia et al., 2024; Oh et al., 2017; Klissarov et al., 2025; Barreto et al., 2017; Carvalho et al., 2023a;b). Recent work, however, has shown that task diversity itself—paired with basic but powerful learning algorithms and architectures—can promote remarkable generalization. Agents trained on many diverse tasks achieve zero-shot sim-to-real transfer (Tobin et al., 2017), generalize zero-shot to held-out games built from the same task primitives (Team et al., 2021; Nikulin et al., 2024), and generalize to human partners without any human training data (Jha et al., 2025). These works, however, have primarily studied generalization while an agent's weights are held fixed.

One prominent idea is that task diversity can induce a generalizable learning strategy. This is perhaps best displayed by work showing that training on diverse tasks gives rise to in-context learning, a form of meta-learning in which an agent uses the examples it sees in context to generalize to new tasks within that distribution (Bauer et al., 2023; Kirsch et al., 2022). However, this work, too, has focused primarily on evaluating agents with fixed weights. What happens when the weights are allowed to change and the task distribution shifts many times?

In this work, we study how task diversity affects continual RL, where an agent must continuously generalize to new tasks (Abel et al., 2023). Prior work has commonly studied adaptation to a new task after an agent has converged on the current one (Kirkpatrick et al., 2017; Rusu et al., 2016; Ahn et al., 2025). However, the world does not wait for an agent to converge. On some tasks an

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.

agent will have converged, on others not. The question is therefore not only whether a final policy generalizes, but whether the learning process itself remains adaptive under task distribution shifts.

Existing environments have not allowed researchers to run controlled experiments on how task diversity affects generalization as an agent continually transfers to new task distributions. Continual RL benchmarks study sequential learning but typically contain at most hundreds of tasks (Wolczyk et al., 2021; Tomilin et al., 2023). GPU-accelerated environments with procedurally generated tasks contain millions or billions of tasks (Nikulin et al., 2024; Matthews et al., 2024), but are not designed to isolate how the amount and type of diversity affect continual learning across controlled distribution shifts.

To enable researchers to run controlled experiments studying how task diversity affects continual RL, we introduce Banyan, a GPU-accelerated domain for continual RL over compositional task spaces. A task is specified by a task tree and a layout (see Figure 1). Task trees specify which objects must be combined or transformed, and in what order, to produce a goal object. Layouts govern how the agent navigates the map. This factorization yields three axes of controllable diversity: layouts, task-tree topologies, and task-tree instances. Banyan supports procedural generation along each axis independently, yielding billions of possible compositional tasks while preserving control over the source and amount of diversity.

Using Banyan, we study whether task diversity produces learning that transfers across distribution shifts, and whether this transfer compounds into continual learning. The answer is mixed. Our contributions are as follows:

- **Banyan, a controlled many-task domain for continual RL.** We introduce a GPU-accelerated procedurally generated environment in which layouts, task-tree instances, and task-tree topologies can be varied independently. This makes it possible to (1) isolate how the amount and type of diversity affect learning under distribution shift, and (2) study continual RL over billions of tasks.
- **Experimental evidence that task diversity improves forward and backward transfer across task distributions.** When agents train on one task distribution and then continue on a different distribution, increasing diversity along each axis reduces the performance drop at the shift boundary.
- **Experimental evidence that diversity does not by itself enable longer-run continual learning, but can inhibit it.** When agents train on a small amount of diversity, we see that this improves learning on a sequence of new task distributions; however, large amounts of diversity hinder learning on new distributions. We find that this phenomenon occurs in other domains, across different architectures, in large architectures, and with different continual learning algorithms.

2 Related work

A key question in reinforcement learning is how experience on one set of tasks affects performance and learning on another. **Forward transfer** measures whether prior experience improves initial performance or learning speed on new tasks; **backward transfer** measures how later learning changes performance on previously encountered tasks, with negative backward transfer corresponding to **forgetting**. Much of the transfer literature builds mechanisms that make experience reusable: successor features factor value functions into reusable dynamics-dependent features and task-specific reward weights (Barreto et al., 2017), hierarchical methods compose lower-level skills into higher-level behavior (Sutton et al., 1999; Dietterich, 2000; Bacon et al., 2017), and multitask and meta-learning methods learn policies, representations, or initializations that can be adapted to new tasks (Teh et al., 2017; Finn et al., 2017). These approaches ask how agents should be structured so that experience transfers. We ask the complementary question: to what extent can transfer be induced by the structure of the task distribution itself?

Task diversity as a source of transfer. A growing body of work shows that diversity in the training distribution can itself drive generalization. Domain randomization varies visual or physical properties to enable sim-to-real transfer (Tobin et al., 2017; Peng et al., 2018; OpenAI et al., 2019); procedural generation constructs distinct training and test distributions over levels or tasks (Cobbe et al., 2019; 2020); and large-scale environments such as XLand and AdA show that broad task distributions can support zero-shot generalization and in-context adaptation (Team et al., 2021; Bauer

et al., 2023). Recent work further shows that task diversity can improve generalization under fixed data budgets (Taiga et al., 2023) and support coordination with novel partners (Jha et al., 2025). Together, these results establish task diversity as a powerful source of post-training generalization and adaptation. They do not, however, show whether diversity makes the learning process itself more transferable as agents continue updating their weights under distribution shift.

Systematic transfer. In compositional domains, the strongest form of transfer is not merely robustness to new instances, but systematic reuse of familiar components in new combinations. This capability is often studied as **systematic generalization** in cognitive science and machine learning (Fodor & Pylyshyn, 1988; Lake & Baroni, 2018; Bahdanau et al., 2019). In RL, related questions arise in modular policy learning, compositional task specifications, and compositional control benchmarks (Andreas et al., 2017; Nangue Tasse et al., 2020; Mendez & Eaton, 2021; Mendez, 2022). None of this work studied whether the data distribution itself could give rise to systematic generalization. Most similar to our work are Hill et al. (2019) and Lake & Baroni (2023), who both study how an agent's data distribution can give rise to systematic generalization. However, neither studied this within a continual learning setting, and Lake & Baroni (2023) focused on natural language processing tasks, not RL tasks.

Many-task environments for continual RL. Studying how task diversity affects continual learning requires both scale and control. Continual RL benchmarks make sequential learning explicit and measure transfer and forgetting across changing tasks (Wolczyk et al., 2021; Tomilin et al., 2023; 2026), but they typically contain at most hundreds of tasks. GPU-accelerated procedurally generated environments make many-task training practical (Nikulin et al., 2024; Matthews et al., 2024), but they are not designed and have not been used to independently vary the amount and type of diversity across controlled distribution shifts. Banyan fills this gap by providing a GPU-accelerated continual RL domain in which layouts, task-tree instances, and task-tree topologies can be varied independently.

3 BANYAN: A DOMAIN FOR CONTROLLED CONTINUAL RL EXPERIMENTS

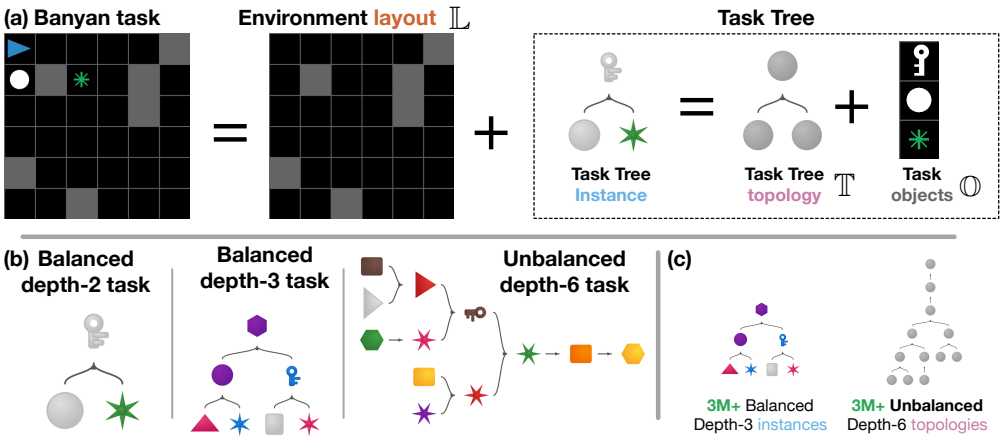

Figure 1: Overview of Banyan. (a) A task decomposes into an environment layout and a task tree; each task tree further decomposes into a topology and an instance over task objects. (b) Tree depth and topology control the horizon and dependency structure of a task. (c) Procedural generation yields millions of task instances and topologies, enabling controlled many-task continual RL.

Banyan is a MiniGrid-style (Chevalier-Boisvert et al., 2023) domain for studying continual RL over large compositional task spaces. A Banyan task $\tau = \langle \mathbb{L}, \mathbb{T}, \mathbb{O} \rangle$ has three independently varied parts (Figure 1): a layout $\mathbb{L}$, a task-tree topology $\mathbb{T}$, and an object assignment $\mathbb{O}$. The **layout** fixes the wall configuration and the initial placement of objects, and so determines the navigation problem the agent must solve. The remaining two pieces specify a **task tree**: a compositional recipe for the goal object. Its leaves are the objects placed in the layout; each internal node is an object produced by transforming or combining its children; the root is the goal. The depth of the tree is the longest leaf-to-root path, so deeper tasks require longer sequences of object interactions. Within the task

tree, the **topology** $\mathbb{T}$ is the abstract shape — depth, branching, and which nodes are transformations versus combinations — and the **object assignment** $\mathbb{O}$ pins concrete identities to the nodes of that shape. Changing $\mathbb{T}$ alters which sub-goals must be composed and in what order, and so changes the optimal subgoal policy. Changing $\mathbb{O}$ preserves the solution structure but changes which objects ground it.

Interacting with Banyan. When an episode begins, a layout and task tree are sampled. First, the layout walls are placed; afterwards, the leaf nodes associated with the task tree are placed. **Observations**. The agent observes the entire grid and a description of the goal object (the root node of the task tree). It does not observe the task tree structure and must learn which sequences of object interactions complete a task. **Actions**. The agent can move around the map using the "up", "down", "left", "right", and "stay" actions. The agent can use the "pickup" and "drop" actions to move items around the map and place particular items adjacent to one another to combine them and create a new item. The agent can also use the "toggle" action while standing on a tile with an item to perform a single-item transformation that creates a new item when needed. **Dynamics**. How two objects combine or the effect of "transform" on a single object stays fixed across tasks. Tasks therefore differ in where the agent must move, which objects are needed, and how those objects compose into the goal. **Rewards**. The agent receives a terminal success reward of $+1$ when the root object is produced and a penalty of -1 if a dead-end condition is triggered. Dead-ends are triggered when the agent places two items adjacent to one another that combine under the world's merge rules but whose combination is not part of the current task tree, consuming an object the task requires. **Episode termination**. An episode terminates if (i) the root goal is achieved, (ii) the episode hits the max-step horizon (timeout), or (iii) a dead-end condition is triggered. We describe the domain in more detail in §A.1.

How task trees relate to their optimal solutions. Holding the topology fixed while sampling new instances creates many tasks with the same solution structure but different object identities. Holding the object rules fixed while sampling new topologies creates new subgoal dependencies from familiar components. Increasing depth changes the horizon of the problem, creating long dependency chains that are harder to solve than shallow trees.

How changing a layout, instance, or topology affects transfer. Layout shifts vary navigation while preserving the object-composition problem. Instance shifts vary the objects grounding a task while preserving the abstract task structure. Topology shifts vary the task structure itself, making them the strongest test of whether learning transfers compositionally.

4 Experiments

Does task diversity induce a generalizable learning strategy that keeps promoting transfer across multiple distribution shifts? We address this with two sets of experiments. §4.1 studies a single shift: how does diversity on the first distribution shape forward transfer to the second, and backward transfer to the first once training completes? §4.2 extends the protocol to a sequence of nine shifts, and asks whether the single-shift effect of diversity persists as shifts compound.

A **task distribution** d is a distribution over Banyan tasks $\tau = (\mathbb{L}, \mathbb{T}, \mathbb{O})$. Every distribution we use spans depths 1 through 6, and we require the depths to nest: every task at depth l also appears as a sub-task inside some task at every deeper level. (As a reminder, a tree's depth is the length of the longest leaf-to-root path; see Figure 1 for examples). The constraint serves two purposes. Deeper tasks are hard to learn from scratch, so the shallower depths act as a built-in curriculum; and the nesting makes diversity easy to count: the depth-6 tasks determine everything shallower, so we count diversity by the depth-6 tasks alone. A distribution with n topologies has $6n$ tasks in total but effective diversity n.

Metrics. To study how task diversity affects transfer, we use metrics for forward and backward transfer between task distributions. For **forward transfer**, we define the **transfer gap** between adjacent distributions: the difference between the agent's terminal success rate on one distribution, $S_{\text{end}}(d_i)$, and its initial success rate on the next, $S_{\text{start}}(d_{i+1})$:

$$\Delta_i = S_{\text{start}}(d_i) - S_{\text{end}}(d_{i-1}). \tag{1}$$

Negative values mean success drops at the shift; a value near zero means the agent begins d_i at the level it reached on d_{i-1}. For **backward transfer**, we ask how later training changes performance

on an earlier distribution. Let $S_{\text{end}}(d_i; k)$ denote success on d_i evaluated after training through phase k, with $S_{\text{end}}(d_i; i) = S_{\text{end}}(d_i)$. Backward transfer is the change in success on d_i between the end of its own phase and the end of phase k:

$$B(k, i) = S_{\text{end}}(d_i; k) - S_{\text{end}}(d_i; i). \tag{2}$$

Positive values mean later training improved performance on d_i; negative values mean forgetting. All plots show mean and standard error across seeds.

Baselines. This work seeks to understand how data diversity can enable transfer and continual learning in RL algorithms. We thus focus on algorithms that minimally use auxiliary tasks and have been shown to benefit from data diversity. For policy-based methods, we focus on Proximal Policy Optimization (PPO; Schulman et al., 2017). For value-based methods, we focus on Parallelised Q-Network (PQN; Gallici et al., 2025). See §B for a summary of the methods we study.

4.1 Diversity produces systematic transfer

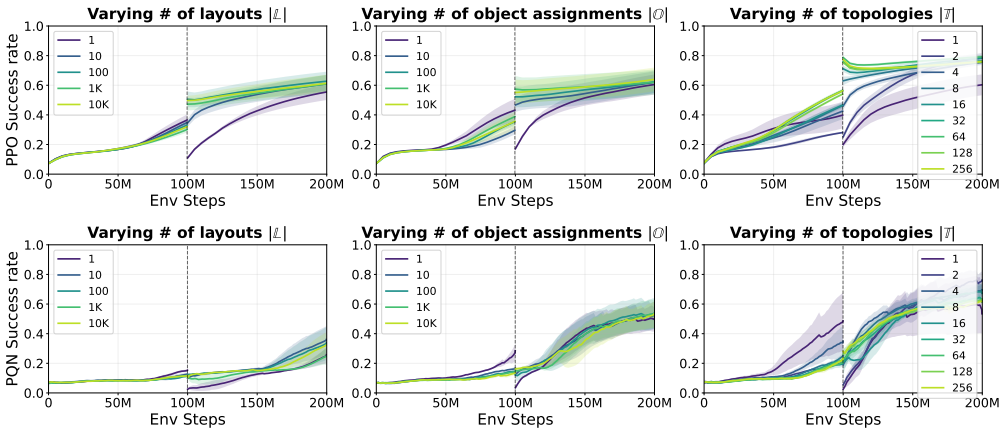

Figure 2: **Increasing task diversity in d_1 leads to the transfer gap Δ_2 to approach 0**. Increasing the number of layouts $\mathbb{L}$, object assignments $\mathbb{O}$, or topologies $\mathbb{T}$, leads agents to begin d_2 close to their performance on d_1. This is true for both PPO (top) and PQN (bottom). Dashed lines mark the $d_1 \to d_2$ boundary.

Experimental setup. We test how holding the world fixed and varying diversity along a single axis affects transfer. In d_1 we vary the number of layouts $\mathbb{L}$, task-tree object assignments $\mathbb{O}$, or task-tree topologies $\mathbb{T}$; d_2 contains 10K new held-out items along the same axis for layouts and object assignments, or 256 for topologies. Each PPO run has 10 seeds and each PQN run has 3 seeds. **Question 1: How does increasing task diversity in d_1 affect forward transfer to d_2?** Across all dimensions we vary, increasing the number of examples in d_1 causes Δ_2 to approach zero (Figure 2; Figure 10). Layout count $|\mathbb{L}|$ and object assignment count $|\mathbb{O}|$ qualitatively have the same effect on reducing the transfer gap, whereas topology count $|\mathbb{T}|$ seems to have a different effect: learning seems to benefit strongly from increasing topology count. We attribute this asymmetry to what each axis demands of the policy. Ignoring efficiency, a policy that is optimal for one layout $\mathbb{L}$ or task-tree object assignment $\mathbb{O}$ of a given topology $\mathbb{T}$ can reapply the subtask progression order it has learned to other layouts or task-tree object assignments of that topology. In contrast, every topology requires its own progression order, so spanning the family requires learning a richer set of knowledge. The results hold for both PPO and PQN, which suggests the effect is not an artifact of how policy and value functions absorb structure.

Question 2: How does increasing task diversity affect backward transfer to d_1 after the agent has been exposed to d_2? Neither layout count $|\mathbb{L}|$ nor object assignment count $|\mathbb{O}|$ visibly affects backward transfer across PPO and PQN. However, varying topology count $|\mathbb{T}|$ leads to markedly better backward transfer across both PPO and PQN (Figure 3; Figure 11). As with the forward-transfer result, we find that varying layout count and varying object assignment count have a similar effect that is qualitatively different from varying topology count.

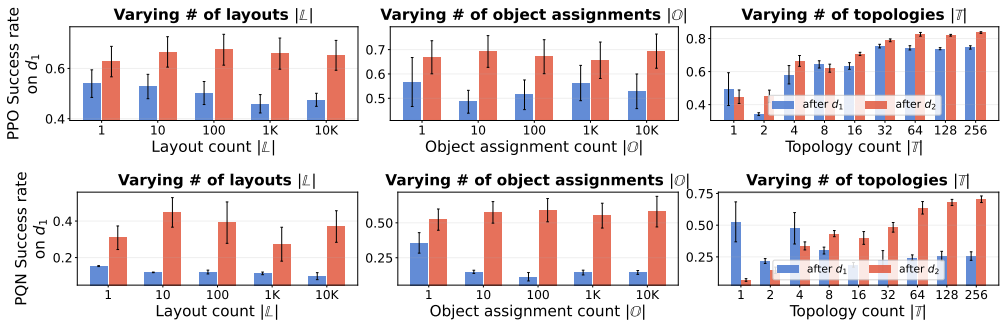

Figure 3: **Increasing task diversity in d_1 improves backward transfer to d_1 after d_2, but only when varying topology count $|\mathbb{T}|$.**

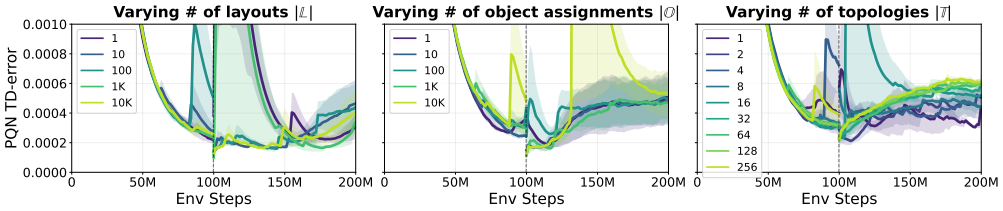

Figure 4: **Increasing task diversity in d_1 leads to value-predictions in d_2 with a lower initial TD-error**. Dashed lines mark the $d_1 \to d_2$ boundary.

Question 3: Why does increasing task diversity induce systematic transfer? One simple hypothesis is that diversity produces better representations, and better representations produce better predictions. PQN lets us check this directly: its policy is derived from its value prediction, so prediction quality is observable as TD-error. We examine the TD-error at the $d_1 \to d_2$ boundary — the first d_2 data the agent's d_1-trained predictions are evaluated on. Higher diversity along every axis yields both a more performant initial policy (Figure 2) *and* a smaller initial TD-error at the onset of d_2 (Figure 4). As is a common empirical phenomenon with value-based methods, as PQN improves, the magnitudes of both predicted return and TD-error grow together.

Question 4: Do the effects of task diversity on transfer hold in a different control task? We adapt PointMass (Connell, 1992; Sutton & Barto, 2018), a domain in which an agent controls a 2D point mass to move objects around by bumping into them, to support tree-structured tasks. **Actions**. The agent selects a continuous (x, y) direction to move one unit per step and moves objects by hitting them. Two objects can be combined by placing both inside a special "combine" object, which then merges them according to the task tree (See §A.2 for more details). We run PPO only, as value-based methods require additional care to extend to continuous action spaces, which we leave for future work. We sweep the object assignment axis $\mathbb{O}$ from 1 to 1,000 balanced trees at depths 1 and 2. Like before, we find that diversity improves systematic transfer—Δ_2 rises from ≈ -0.95 at $n = 1$ to ≈ -0.5 at $n = 1000$—and improves backward transfer—$B(2, 1)$ rises from ≈ -0.8 to ≈ 0 across the same range (Figure 5; Figure 12).

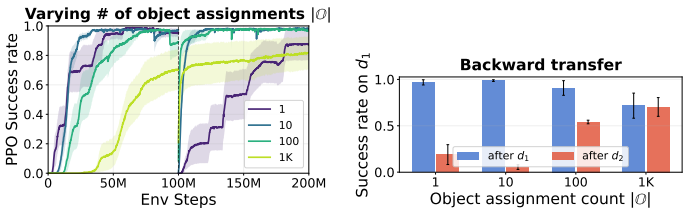

Figure 5: **Task diversity also improves forward and backward transfer in a continuous control domain.**

4.2 Systematic transfer does not enable continual reinforcement learning

We just showed that increasing task diversity can improve both forward and backward transfer within a single distribution shift. Here we ask whether this effect compounds across longer sequences by

studying transfer across ten task distributions. We focus on PPO, which we found more performant than PQN in the continual setting. All experiments use 3 seeds unless otherwise stated.

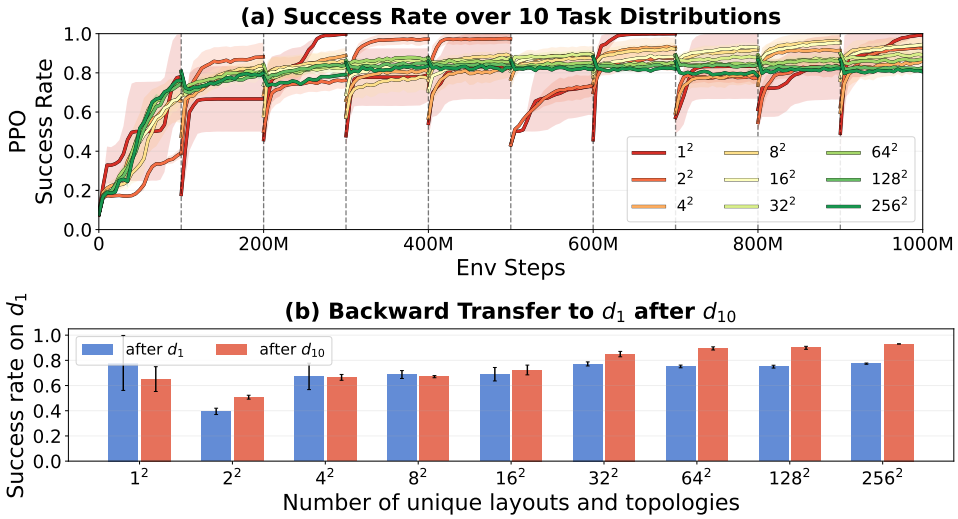

Figure 6: **Task diversity inhibits forward transfer in continual RL but improves backward transfer**. (a) Task diversity leads to continual improvement at intermediate diversity ($n^2 = 16^2$), but inhibits continual learning at high diversity ($n^2 = 256^2$). Success rates across 9 distribution shifts, averaged over depths 1–6. (b) Increasing task diversity improves backward transfer to the first task distribution after 9 distribution shifts. Dashed lines mark the $d_i \to d_{i+1}$ boundary.

Question 5: How does task diversity affect transfer across a sequence of 10 task distributions? We co-vary layout and topology diversity from $n = 1$ to $n = 256$, yielding up to $n^2 = 65{,}536$ depth-6 layout–topology combinations per distribution. As in the single-shift experiment, increasing diversity drives $\Delta_i \to 0$ at every transition: at high n, each new distribution begins at the success rate the agent reached on the previous one (Figure 6). At intermediate diversity ($n = 16$) the agent continues to learn after each transfer: $S_{\text{end}}(d_i)$ climbs from ≈ 0.69 at d_1 to ≈ 0.94 at d_{10}. At the highest diversity, this intra-distribution improvement shrinks: at $n = 256$, $S_{\text{end}}(d_i)$ moves only from ≈ 0.79 at d_3 to ≈ 0.81 at d_{10}. The endpoint $S_{\text{end}}(d_{10})$ in fact declines monotonically as diversity grows past $n = 16$, falling from 0.94 at $n = 16$ through $0.89, 0.87, 0.83$ to 0.81 at $n = 256$. Backward transfer moves in the opposite direction: as diversity grows, the agent's performance on d_1 after the full sequence exceeds its level at the end of phase 1; at $n = 256$, $S_{\text{end}}(d_1; 1) \approx 0.77$ rises to $S_{\text{end}}(d_1; 10) \approx 0.93$ (Figure 6; Figure 13). This suggests that when an agent is forced to shift task distributions, the network keeps learning the general task structure but is inhibited in its ability to learn the structure specific to each new task distribution. These results hold as we increase the capacity of the network (Figure 17).

Question 6: How do different continual learning methods better accommodate task diversity during continual RL? We study three continual learning methods: (1) **Continual backprop** (Dohare et al., 2024), which targets the loss of plasticity that arises under repeated distribution shifts; (2) **Replay** (Shin et al., 2017), which continues training on earlier task distributions so the agent never trains on the current distribution alone; (3) **Multi-head** policies (Kessler et al., 2022; Caccia et al., 2023), which give each distribution its own output head while a shared body absorbs the structure the distributions have in common. No method restores continued improvement at high diversity (Figure 7, (a–c)). Unsurprisingly, since the multi-head policy must learn a new output head for each distribution, its success drops to about 0.1 at each shift before recovering to the level of the other methods. For backward transfer, every method with a shared policy head retains the benefit of diversity. Multi-head policies, however, lose it entirely (Figure 7, (d–f); Figure 14). This suggests that sharing a policy head is key to learning shared structure across task distributions.

Question 7: Do transformers better accommodate task diversity during continual RL? The plateau at high diversity could reflect the recurrent network rather than the learning problem, so we ask whether a transformer absorbs high-diversity task structure that an RNN cannot. We adapt an

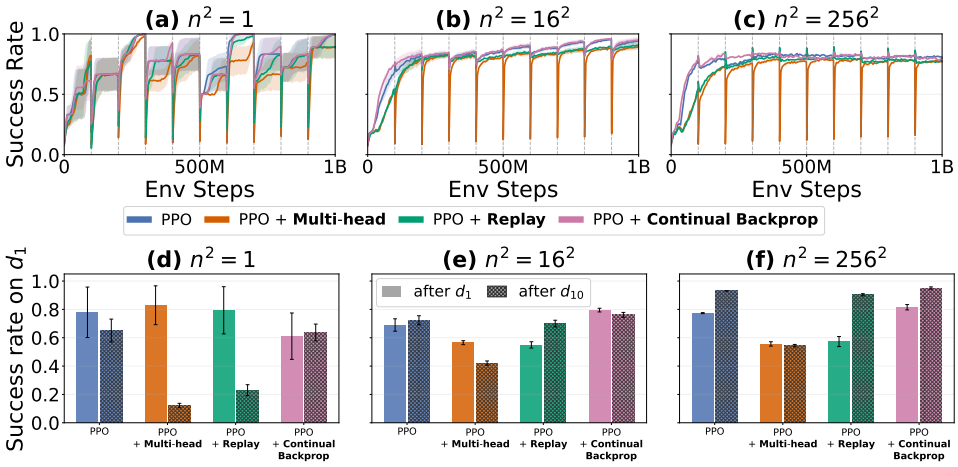

Figure 7: **No continual-learning approach mitigates the negative effect of task diversity on continual RL, but methods with a shared policy head retain the backward-transfer benefit.**

open-source implementation of PPO + Transformer-XL (Parisotto et al., 2020) and hold everything else fixed. Transformers reproduce the same qualitative forward and backward transfer (Figure 8; Figure 15): at high diversity, continued improvement stalls and backward transfer improves with diversity, just as for the RNN.

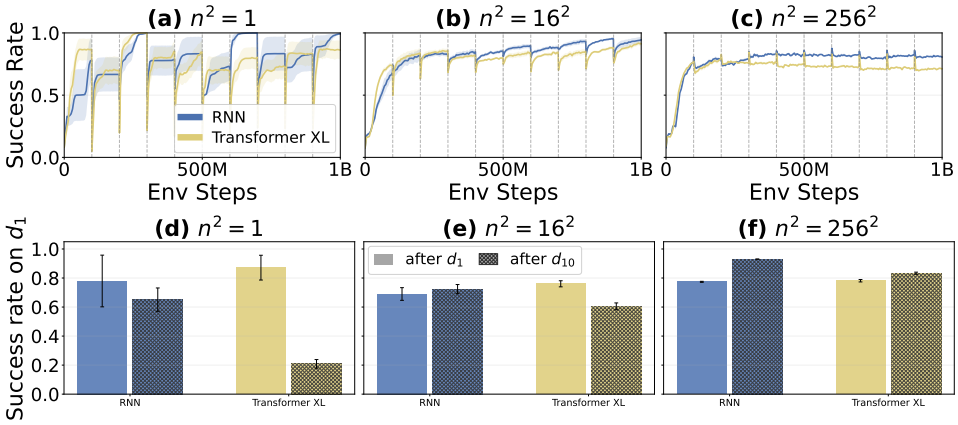

Figure 8: **Using a Transformer memory architecture does not mitigate the negative effect of task diversity on continual RL.** 5 seeds for the Transformer-XL.

Question 8: Do the effects of task diversity show up in a more realistic, less structured domain? Banyan tasks are cleanly specified by task trees, so a natural question is whether the same phenomenon appears when they are not. We study Kinetix (Matthews et al., 2025), an open-ended 2D physics-based control domain in which every task shares one goal: make the green and blue objects touch without the green object touching the red one. Kinetix procedurally generates millions of tasks, so we vary diversity by changing the number of random tasks in each task distribution from $n^2 = 1$ to $n^2 = 256^2$. **Observations**. The agent receives a symbolic description of every shape, joint, and thruster in the scene. **Actions**. The agent sets each motor to reverse, off, or forward and each thruster to off or on. **Dynamics**. Rigid-body 2D physics with gravity, collisions, and the motors and thrusters the agent controls. **Reward**. The agent receives $+1$ when the green and blue objects touch and -1 when the green and red objects touch; either event ends the episode. Kinetix is harder than Banyan, so success rates are lower throughout. The pattern, however, is the same. At intermediate diversity the agent keeps improving across task distributions, whereas at high diversity its improvement plateaus (Figure 9, (b)). And just as in Banyan, task diversity improves backward transfer: $B(10, 1)$ is strongly negative at $n^2 = 1$ and $n^2 = 16^2$ and turns positive at $n^2 = 256^2$ (Figure 9, (c); Figure 16). No forgetting is itself a gain, and success on d_1 after the full sequence is higher at $n^2 = 256^2$ than at $n^2 = 16^2$.

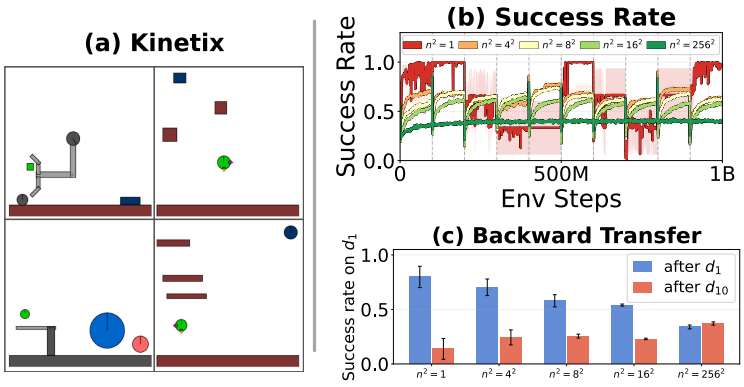

Figure 9: **Kinetix reproduces the same qualitative trade-off between diversity and continual RL observed in Banyan.** (a) Example Kinetix tasks. (b) Success rates across 9 distribution shifts as task diversity varies. (c) Backward transfer to the first task distribution after nine shifts.

5 Discussion

Our results present a paradox. Across a single distribution shift, increasing diversity along any axis drove the transfer gap toward zero—at the highest diversity, the agent entered each new distribution at the performance it had reached on the previous one. Across nine shifts the relationship shifted. The same diversity that smoothed every single boundary instead caused agents to plateau, and the highest diversity produced the least learning over the full sequence. This pattern persists with larger networks, across continual learning methods—i.e. continual backprop, specialized policy heads, replay—with Transformer memory, and in Kinetix—suggesting that it is not solely a problem of capacity, plasticity, shared-head interference, architecture, or of Banyan. Across these experiments, backward transfer reveals the same signature: performance on the first distribution keeps climbing while performance on each new distribution flatlines. This suggests that updates refine structure shared across distributions at the expense of fitting structure specific to each new distribution. What stalls is not learning, but specialization. We hypothesize that this may be a continual learning version of the "primacy bias" (Nikishin et al., 2022), where the policy an agent learns is biased by the features it first learned to represent. Here, the policy an agent learns as it encounters new task distributions seems "primed" by the task distributions it encountered early on. We leave studying this further to future work.

Limitations. Our experiments span learning algorithms, memory architectures, and continual-learning methods, but we varied one component at a time. Combinations of these components could mitigate the negative effects of task diversity on continual RL, even though none individually resolved them. We also tested whether our findings extend to PointMass, a continuous-control domain, and Kinetix, an open-ended physics-based control domain, but all three domains remain far from real-world settings. Finally, although the effects persist with larger networks, we did not study architectures with trillions of parameters. It remains possible that *more is different* (Anderson, 1972), i.e. that sufficiently large architectures could enter a fundamentally different learning regime.

Conclusion. More work is needed to determine whether task diversity inhibits continual RL in the real-world settings we care about. The current scaling paradigm in RL relies on relatively simple learning algorithms that benefit from increasingly diverse training data. Large RL models trained this way can generalize to new tasks, but they are typically deployed with frozen network parameters. Our results suggest that the same diversity that supports generalization can also inhibit continual learning when parameters continue to change across a sequence of task distributions. A key question is therefore how to use diversity to promote continual reinforcement learning rather than inhibit it. We do not yet know whether our findings extend to real-world settings, but we hope Banyan supports the development of methods that address this challenge if they do. Learning and scale, on their own, may not be enough.

USE OF LARGE LANGUAGE MODELS

Writing. Large language models (LLMs) polished the text of the main paper, formatted bibliography entries, and audited the draft for errors and gaps in logic. In the appendix they also rearranged content, for example separating algorithm definitions from implementation details.

Retrieval and discovery. LLMs searched for related work, including prior continual learning methods that revisit earlier task distributions rather than storing experience. The authors verified every citation they surfaced before inclusion.

Research execution. LLMs adapted open-source implementations of Transformer-XL PPO and PQN to our codebase. They also launched experiments and hyperparameter searches under author supervision, and wrote and edited the scripts that generate the figures from logged runs.

Drafting. LLMs drafted the appendix descriptions of algorithms and experiment settings by reading our experiment files and Weights & Biases runs directly to extract values. We wrote scripts to double-check every extracted value against the runs that produced the figures.

REFERENCES

David Abel, Andre Barreto, Benjamin Van Roy, Doina Precup, Hado van Hasselt, and Satinder Singh. A definition of continual reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=ZZS9WEWYbD.

Hongjoon Ahn, Jinu Kim, Minjong Yeo, Dohun Kim, Junhyuk Lee, Haneol Kim, Sungbin Lee, and Taesup Moon. Reset & distill: Overcoming negative transfer in continual reinforcement learning. In *Proceedings of the 13th International Conference on Learning Representations (ICLR)*, 2025.

P. W. Anderson. More is different. *Science*, 177(4047):393–396, 1972. doi: 10.1126/science.177.4047.393.

Jacob Andreas, Dan Klein, and Sergey Levine. Modular multitask reinforcement learning with policy sketches. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 166–175. JMLR.org, 2017.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pp. 1726–1734. AAAI Press, 2017.

Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron Courville. Systematic generalization: What is required and can it be learned? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HkezXnA9YX.

André Barreto, Will Dabney, Rémi Munos, Jonathan J. Hunt, Tom Schaul, Hado van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 4058–4068, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Jakob Bauer, Kate Baumli, Feryal Behbahani, Avishkar Bhoopchand, Nathalie Bradley-Schmieg, Michael Chang, Natalie Clay, Adrian Collister, Vibhavari Dasagi, Lucy Gonzalez, Karol Gregor, Edward Hughes, Sheleem Kashem, Maria Loks-Thompson, Hannah Openshaw, Jack Parker-Holder, Shreya Pathak, Nicolas Perez-Nieves, Nemanja Rakicevic, Tim Rocktäschel, Yannick Schroecker, Satinder Singh, Jakub Sygnowski, Karl Tuyls, Sarah York, Alexander Zacherl, and Lei Zhang. Human-timescale adaptation in an open-ended task space. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

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*, 2023.

Wilka Carvalho, Angelos Filos, Richard L. Lewis, Honglak Lee, and Satinder Singh. Composing task knowledge with modular successor feature approximators. In *International Conference on Learning Representations*, 2023a.

Wilka Carvalho, Andre Saraiva, Angelos Filos, Andrew Lampinen, Loic Matthey, Richard L. Lewis, Honglak Lee, Satinder Singh, Danilo J. Rezende, and Daniel Zoran. Combining behaviors with the successor features keyboard. In *Advances in Neural Information Processing Systems*, 2023b.

Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo De Lazcano Perez-Vicente, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and J K Terry. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=PFfmfspm28.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1724–1734, 2014.

Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1282–1289. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/cobbe19a.html.

Karl Cobbe, Christopher Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Jonathan H. Connell. SSS: A hybrid architecture applied to robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2719–2724, 1992.

Thomas G. Dietterich. Hierarchical reinforcement learning with the maxq value function decomposition. *J. Artif. Int. Res.*, 13(1):227–303, November 2000. ISSN 1076-9757.

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. doi: 10.1038/s41586-024-07711-7.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 1126–1135. JMLR.org, 2017.

Jerry A. Fodor and Zenon W. Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1):3–71, 1988. ISSN 0010-0277. doi: https://doi.org/10.1016/0010-0277(88)90031-5. URL https://www.sciencedirect.com/science/article/pii/0010027788900315.

Matteo Gallici, Mattie Fellows, Benjamin Ellis, Bartomeu Pou, Ivan Masmitja, Jakob Nicolaus Foerster, and Mario Martin. Simplifying deep temporal difference learning. *The International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2407.04811.

Felix Hill, Andrew Lampinen, Rosalia Schneider, Stephen Clark, Matthew Botvinick, James L McClelland, and Adam Santoro. Environmental drivers of systematicity and generalization in a situated agent. *arXiv preprint arXiv:1910.00571*, 2019.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997.

Kunal Jha, Wilka Carvalho, Yancheng Liang, Simon Shaolei Du, Max Kleiman-Weiner, and Natasha Jaques. Cross-environment cooperation enables zero-shot multi-agent coordination. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=zBBYsVGKuB.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, 1998.

Samuel Kessler, Jack Parker-Holder, Philip Ball, Stefan Zohren, and Stephen J. Roberts. Same state, different task: Continual reinforcement learning without interference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

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.

Louis Kirsch, James Harrison, Jascha Sohl-Dickstein, and Luke Metz. General-purpose in-context learning by meta-learning transformers. *arXiv preprint arXiv:2212.04458*, 2022.

Martin Klissarov, Mikael Henaff, Roberta Raileanu, Shagun Sodhani, Pascal Vincent, Amy Zhang, Pierre-Luc Bacon, Doina Precup, Marlos C. Machado, and Pierluca D'Oro. MaestroMotif: Skill design from artificial intelligence feedback. In *International Conference on Learning Representations*, 2025.

Brenden M Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. *International Conference on Machine Learning (ICML)*, 2018.

Brenden M Lake and Marco Baroni. Human-like systematic generalization through a meta-learning neural network. *Nature*, 623(7985):115–121, 2023.

Michael Matthews, Michael Beukman, Benjamin Ellis, Mikayel Samvelyan, Matthew Jackson, Samuel Coward, and Jakob Foerster. Craftax: a lightning-fast benchmark for open-ended reinforcement learning. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Michael Matthews, Michael Beukman, Chris Lu, and Jakob Foerster. Kinetix: Investigating the training of general agents through open-ended physics-based control tasks. In *International Conference on Learning Representations*, 2025.

Pietro Mazzaglia, Tim Verbelen, Bart Dhoedt, Aaron Courville, and Sai Rajeswar. GenRL: Multimodal-foundation world models for generalization in embodied agents. In *Advances in Neural Information Processing Systems*, 2024.

Jorge A. Mendez. Lifelong machine learning of functionally compositional structures, 2022. URL https://arxiv.org/abs/2207.12256.

Jorge A. Mendez and Eric Eaton. Lifelong learning of compositional structures. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://arxiv.org/abs/2007.07732.

Geraud Nangue Tasse, Steven James, and Benjamin Rosman. A boolean task algebra for reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9497–9507. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6ba3af5d7b2790e73f0de32e5c8c1798-Paper.pdf.

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron Courville. The primacy bias in deep reinforcement learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16828–16847. PMLR, 2022. URL https://proceedings.mlr.press/v162/nikishin22a.html.

Alexander Nikulin, Vladislav Kurenkov, Ilya Zisman, Artem Sergeevich Agarkov, Viacheslav Sinii, and Sergey Kolesnikov. XLand-minigrid: Scalable meta-reinforcement learning environments in JAX. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=zg8dpAGl1I.

Junhyuk Oh, Satinder Singh, Honglak Lee, and Pushmeet Kohli. Zero-shot task generalization with multi-task deep reinforcement learning. In *International Conference on Machine Learning*, 2017.

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, 2019. URL https://arxiv.org/abs/1910.07113.

Emilio Parisotto, Francis Song, Jack Rae, Razvan Pascanu, Caglar Gulcehre, Siddhant Jayakumar, Max Jaderberg, Raphael Lopez Kaufman, Aidan Clark, Seb Noury, Matthew Botvinick, Nicolas Heess, and Raia Hadsell. Stabilizing transformers for reinforcement learning. In *International Conference on Machine Learning*, 2020.

Jing Peng and Ronald J Williams. Incremental multi-step Q-learning. *Machine Learning*, 22:283–290, 1996.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3803–3810, 2018. doi: 10.1109/ICRA.2018.8460528.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016.

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

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 2nd edition, 2018.

Richard S. Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: a framework for temporal abstraction in reinforcement learning. *Artif. Intell.*, 112(1–2):181–211, August 1999. ISSN 0004-3702. doi: 10.1016/S0004-3702(99)00052-1. URL https://doi.org/10.1016/S0004-3702(99)00052-1.

Adrien Ali Taiga, Rishabh Agarwal, Jesse Farebrother, Aaron Courville, and Marc G Bellemare. Investigating multi-task pretraining and generalization in reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=sSt9fROSZRO.

Open Ended Learning Team, Adam Stooke, Anuj Mahajan, Catarina Barros, Charlie Deck, Jakob Bauer, Jakub Sygnowski, Maja Trebacz, Max Jaderberg, Michael Mathieu, Nat McAleese, Nathalie Bradley-Schmieg, Nathaniel Wong, Nicolas Porcel, Roberta Raileanu, Steph Hughes-Fitt, Valentin Dalibard, and Wojciech Marian Czarnecki. Open-ended learning leads to generally capable agents, 2021. URL https://arxiv.org/abs/2107.12808.

Yee Whye Teh, Victor Bapst, Wojciech Marian Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: robust multitask reinforcement learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 4499–4509, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 23–30, 2017. doi: 10.1109/IROS.2017.8202133.

Tristan Tomilin, Meng Fang, Yudi Zhang, and Mykola Pechenizkiy. COOM: A game benchmark for continual reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=qmCxdPkNsa.

Tristan Tomilin, Luka van den Boogaard, Samuel Garcin, Constantin Ruhdorfer, Bram Grooten, Fabrice Kusters, Yali Du, Andreas Bulling, Mykola Pechenizkiy, and Meng Fang. MEAL: A benchmark for continual multi-agent reinforcement learning, 2026. URL https://openreview.net/forum?id=I3W8PynQU0.

Maciej Wolczyk, Michał Zając, Razvan Pascanu, Łukasz Kuciński, and Piotr Miłoś. Continual world: A robotic benchmark for continual reinforcement learning. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=5qsptDcsdEj.