

Task diversity produces systematic transfer but inhibits continual reinforcement learning

Purab Seth^{1,2,*}, Neil Shah^{1,2,*}, Kunal Jha³,
Samuel J. Gershman^{1,4}, Max Kleiman-Weiner³, Wilka Carvalho^{1,†}

¹Kempner Institute for the Study of Natural and Artificial Intelligence, Harvard University

²Harvard College

³University of Washington

⁴Department of Psychology and Center for Brain Science, Harvard University

Abstract

Continual reinforcement learning aims to produce agents that learn not only to improve at their current tasks but also to adapt as task distributions change. Training an agent on many diverse tasks can induce zero-shot generalization, but previous work generally evaluates this generalization *after* training — with frozen weights. Whether task diversity also improves an agent’s ability to *continue* learning across 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. Paradoxically, while increasing diversity improves systematic transfer, we find that *too much* diversity inhibits a learner’s ability to continue optimizing performance on new task distributions. As diversity increases, learners plateau in the success rate they achieve on a new task distribution while continuing to improve on earlier task distributions—more diverse early task distributions interfere with later task distributions. We release Banyan as a domain for studying continual RL in the many-task regime on an academic budget.

1 Introduction

Recent work has demonstrated that task diversity in training can lead to more capable reinforcement learning (RL) agents. RL agents trained on millions of different games can generalize zero-shot to held-out games constructed from the same task primitives (Team et al., 2021; Nikulin et al., 2024); combining meta-learning with vast task diversity produces agents that adapt entirely in context to novel, open-ended problems (Bauer et al., 2023); and recent multi-agent work shows that diverse tasks can support generalization to coordination with new partners (Jha et al., 2025). However, these works primarily evaluate fixed policies or in-context adaptation after large-scale training, leaving open whether diversity also improves an agent’s ability to continue learning through weight updates as the task distribution changes.

This is the continual RL setting, in which agents sequentially learn across changing task distributions (Abel et al., 2023). Unlike zero-shot evaluation, continual RL does not freeze the policy; unlike in-context adaptation, it does not hold the agent’s weights fixed. Distribution shifts do not necessarily

*Equal contribution. †Corresponding author: wcarvalho@g.harvard.edu.

wait for convergence. Instead, an agent may enter a new task distribution while it is still learning the previous one. 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 do not isolate this question. Benchmarks for continual RL study sequential learning but typically contain at most hundreds of tasks (Wolczyk et al., 2021; Tomilin et al., 2023). Meanwhile, GPU-accelerated environments with procedurally generated tasks or environments contain millions or billions of tasks (Nikulin et al., 2024; Matthews et al., 2024), but they are not designed to isolate how the amount and type of diversity affect continual learning across controlled distribution shifts.

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. 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, if this diversity grows exponentially, we see that it quickly hinders learning on new distributions.

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 controlled benchmark for continual RL

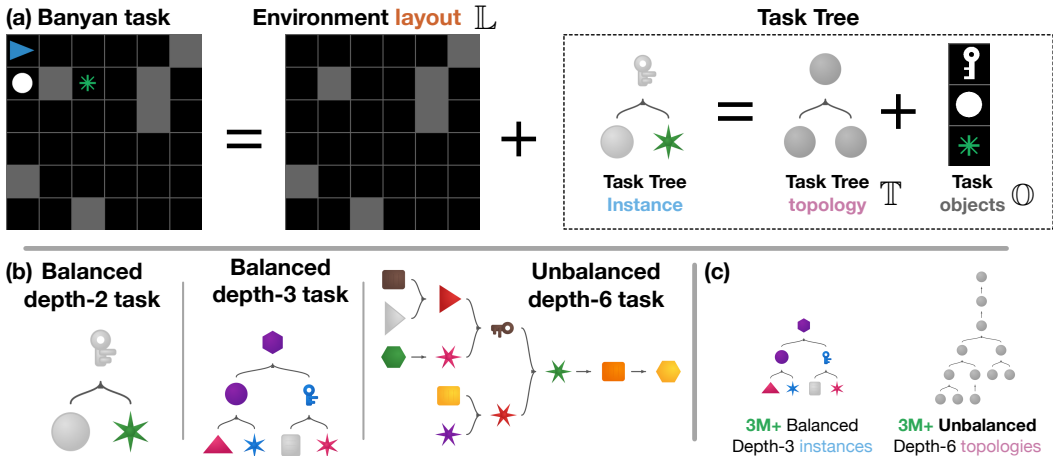


Figure 1: Overview of Banyan. (a) A task decomposes into an environment layout and a task tree; each task tree further decomposes into a topology and an instance over task objects. (b) Tree depth and topology control the horizon and dependency structure of a task. (c) Procedural generation yields millions 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 combines two items or transforms a single item in a way that is valid under the merge rules but leads to an unsolvable state for the particular tree instance the agent was solving. **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.

How are task trees related 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 does changing a layout, instance, or topology affect 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. Some experiments also compare shared-object and disjoint-object shifts: shared-object shifts preserve the object vocabulary across distributions, whereas disjoint-object shifts introduce new object identities and item-specific rules while preserving the same abstract rule structure.

4 Experiments

Does controlled task diversity translate into transferable learning, and does that transfer compound across distribution shifts? We address this in two regimes. §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 ten 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, $S_{\text{end}}(d_{i-1})$, and its initial success rate on the next, $S_{\text{start}}(d_i)$:

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

A value near zero means the agent enters d_i at the performance level it reached on d_{i-1} ; larger values measure the disruption at the boundary. For **backward transfer**, we ask how later training changes performance on an earlier distribution. Let $S_{\text{end}}(d_i; t)$ denote the agent’s success rate on d_i when evaluated after training on phase t , with the convention $S_{\text{end}}(d_i; i) = S_{\text{end}}(d_i)$. We define

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

where positive values indicate positive backward transfer and negative values indicate forgetting.

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).

4.1 Diversity produces systematic transfer

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 5 seeds and each PQN line has 3 seeds.

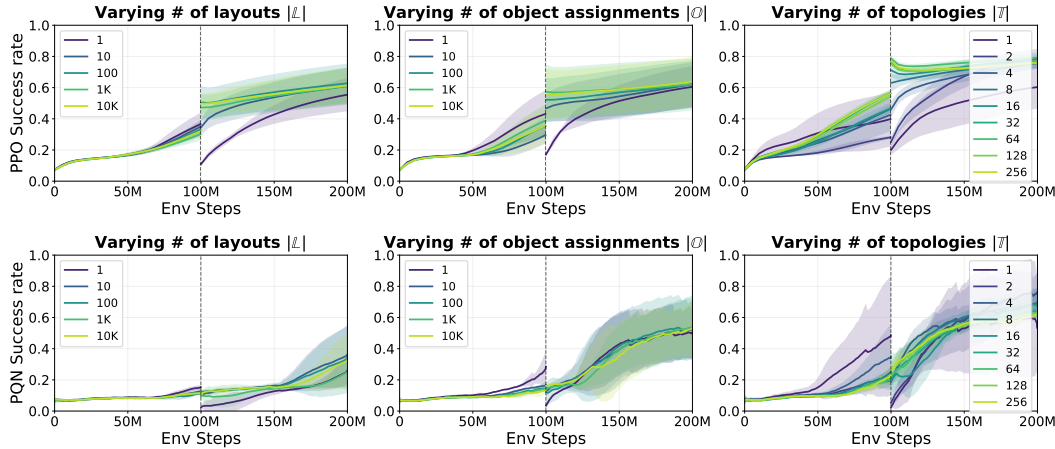


Figure 2: **Increasing task diversity in d_1 causes the transfer gap Δ_2 to approach 0.** Increasing the number of layouts \mathbb{L} , object assignments \mathbb{O} , or topologies \mathbb{T} leads agents to begin d_2 close to their performance on d_1 . This is true for both PPO (top) and PQN (bottom).

Question 1: How does increasing task diversity in d_1 affect forward transfer to d_2 ? Across all dimensions we vary, as we increase the number of examples in d_1 , we see that Δ_2 quickly goes to 0 (Figure 2). 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.

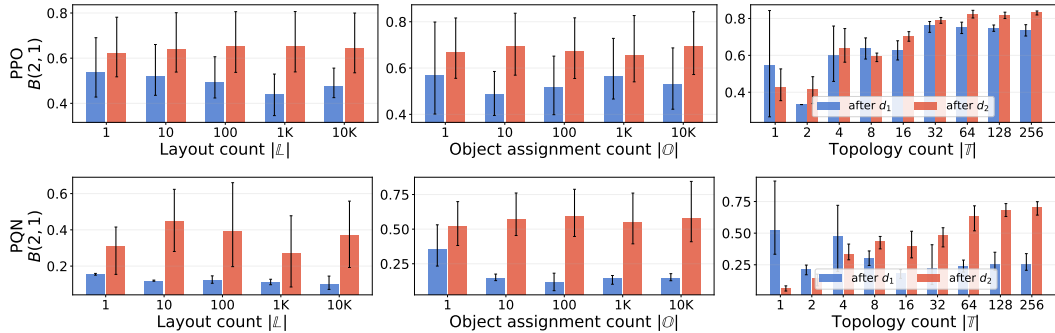


Figure 3: **Increasing task diversity in d_1 improves backward transfer to d_1 after d_2 , but only when varying topology count $|\mathbb{T}|$.** This is true for both PPO (top) and PQN (bottom).

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. As with R1, we find that varying layout count and varying object assignment count have a similar effect that is qualitatively different from varying topology count.

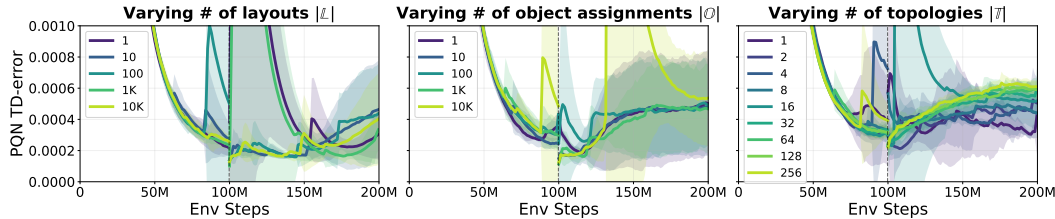


Figure 4: **Higher task diversity yields lower TD-error at the $d_1 \rightarrow d_2$ boundary.** PQN TD-error over training as diversity increases along each axis: layouts $|\mathbb{L}|$ (left), object assignments $|\mathbb{O}|$ (middle), topologies $|\mathbb{T}|$ (right). Dashed lines mark the $d_1 \rightarrow d_2$ 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 $d_1 \rightarrow d_2$ boundary — the first d_2 data the agent’s d_1 -trained predictions are evaluated on. Figure 4 shows that higher diversity along every axis yields both a more performant initial policy *and* a smaller initial TD-error at the onset of d_2 . 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? To test whether R1’s diversity effect depends on Banyan’s discrete gridworld control, we re-run R1 in a continuous-control substrate.

We adapt Point Mass (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. Reward, horizon, and termination match Banyan. 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. The R1 effect replicates: Δ_2 falls from .95 at $n = 1$ to .5 at $n = 1000$, and $B(2, 1)$ falls from ≈ 0.8 to ≈ 0.01 across the same range (Figure 5).

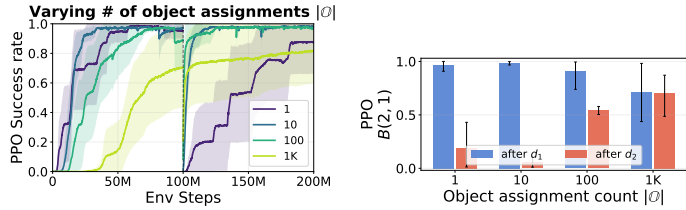


Figure 5: Task diversity also improves forward and backward transfer in a continuous control domain.

4.2 Systematic transfer does not enable continual reinforcement learning

Experimental setup. The single-shift study showed that diversity along any one axis closes the forward transfer gap to d_2 ($\Delta_2 \rightarrow 0$) and, in the case of topologies \mathbb{T} , improves backward transfer to d_1 . Whether either effect compounds across a longer sequence is unclear, as gains at one boundary need not survive nine more. Here we extend the protocol to ten distributions d_1, \dots, d_{10} . We use Continual Backprop (CBP; Dohare et al., 2024) as our baseline, which targets the loss-of-plasticity failure mode that arises under repeated distribution shifts.

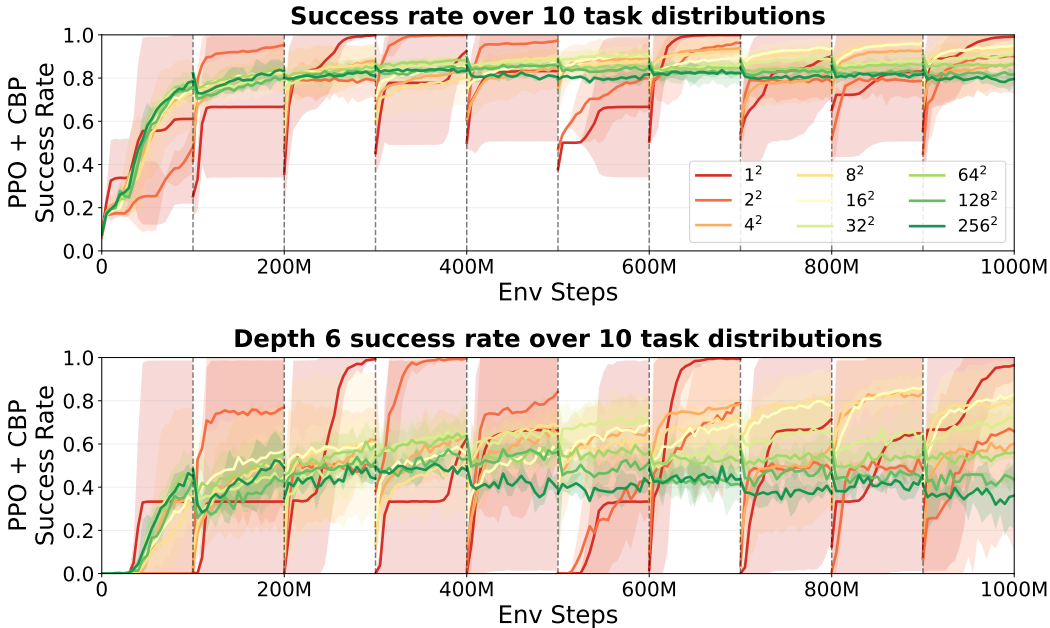


Figure 6: Diversity closes every transfer gap yet stalls long-run learning. PPO with Continual Backprop across the ten-distribution sequence d_1, \dots, d_{10} , sweeping co-varied layout/topology diversity n from 1 to 256. Top: success averaged over depths 1–6; bottom: depth 6 alone.

Question 5: How does task diversity affect transfer across a sequence of 10 task distributions? We sweep diversity along each axis from $n = 1$ to $n = 256$, yielding up to $256 \times 256 = 65,536$ unique tasks per distribution. As in R1, increasing diversity drives $\Delta_i \rightarrow 0$ 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 ≈ 0.74 at d_1 to ≈ 0.95 at d_{10} . At the highest diversity, this within-sequence improvement disappears: at $n = 256$, $S_{\text{end}}(d_i)$ stays near 0.80 across all ten distributions. The endpoint $S_{\text{end}}(d_{10})$ in fact declines monotonically as diversity grows past $n = 16$, falling from 0.95 at $n = 16$ through 0.91, 0.86, 0.82 to 0.80 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 $n = 256$, $S_{\text{end}}(d_1; 1) \approx 0.80$ rises to $S_{\text{end}}(d_1; 10) \approx 0.95$ (Figure 7). 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.

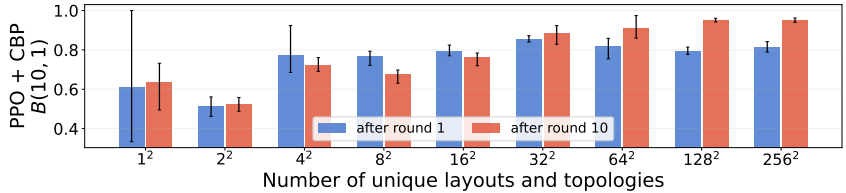


Figure 7: **Increasing task diversity improves backward transfer $B(10, 1)$.** Success on d_1 after round 1 versus after round 10, averaged over depths 1–6, against the number of unique layouts and topologies n^2 .

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 ten 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. We considered two accounts for the plateau — optimization, and interference. This result continues even with continual backprop (Dohare et al., 2024) suggesting that interference is the most likely candidate. The signature is that the agent’s performance on d_1 keeps climbing ($B(i, 1)$ grows) while $S_{\text{end}}(d_i)$ flatlines, as updates refine structure shared across distributions at the expense of fitting what each new one makes new. What stalls is not learning but specialization.

To ask what diversity was buying, we examined PQN’s value predictions at the boundary between distributions. PQN derives its policy from a single value function, so its TD-error at the $d_1 \rightarrow d_2$ boundary measures how well features trained on d_1 predict d_2 before the agent has trained on d_2 . Higher diversity yielded both a more performant initial policy and a smaller initial TD-error (Figure 4), evidence that diversity shapes features that generalize off-distribution.

To test whether these effects depend on the discrete grid-world control of Banyan, we replicated the single-shift sweep in a continuous point-mass domain. The diversity effect held: Δ_2 fell from 0.95 at $n = 1$ to 0.5 at $n = 1000$, and $B(2, 1)$ approached zero across the same range (Figure 5). Whatever diversity is buying for the agent, it is bought at a level above the specifics of the control problem.

Limitations. Banyan is a grid-world environment, so the ecological validity of these effects remains unclear; our continuous-control replication is reassuring, but we do not vary visual diversity here. Our agents rely on recurrent rather than attention-based architectures throughout, leaving open whether transformers are similarly inhibited by diversity under repeated distribution shift. We also restrict ourselves to minimal PPO and PQN baselines, so it is an open question whether mechanisms that already help low-diversity continual learning — replay, modular policies, model-based planning — can recover the compounding benefits of diversity we find disappear here, or whether the trade-off is intrinsic to weight-sharing function approximation.

Conclusion. If we want minimal algorithms that benefit from scaling data, the question is how to use diversity to promote continual reinforcement learning rather than to inhibit it. Our results suggest that the same diversity that smooths individual transitions also stalls the longer-run learning it is meant to enable. Learning and scale, on their own, may not be not enough.

Acknowledgements

We thank Natasha Jaques, Andrew Lampinen, and the Computational Cognitive Neuroscience group at Harvard for helpful discussions and feedback. This work was supported by a gift from the Chan Zuckerberg Initiative Foundation to establish the Kempner Institute for the Study of Natural and Artificial Intelligence.

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=ZS9WEWYbD>.
- 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.
- 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-Schmiege, 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.
- 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>.
- 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 Mahmoode, 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](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.
- 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>.
- 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.
- 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.
- 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=zg8dpAG11I>.
- 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>.
- 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.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 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](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>.