Artificial Generational Intelligence: Cultural Accumulation in Reinforcement Learning

Jonathan Cook *
FLAIR, University of Oxford
jonathan.cook2@hertford.ox.ac.uk

Chris Lu *
FLAIR, University of Oxford
christopher.lu@eng.ox.ac.uk

Edward Hughes
Google DeepMind
edwardhughes@google.com

Joel Z. Leibo
Google DeepMind
jzl@google.com

Jakob Foerster
FLAIR, University of Oxford
jakob@eng.ox.ac.uk

Abstract

Cultural accumulation drives the open-ended and diverse progress in capabilities spanning human history. It builds an expanding body of knowledge and skills by combining individual exploration with inter-generational information transmission. Despite its widespread success among humans, the capacity for artificial learning agents to accumulate culture remains under-explored. In particular, approaches to reinforcement learning typically strive for improvements over only a *single* lifetime. Generational algorithms that do exist fail to capture the open-ended, emergent nature of cultural accumulation, which allows individuals to trade-off innovation and imitation. Building on the previously demonstrated ability for reinforcement learning agents to perform social learning, we find that training setups which balance this with independent learning give rise to cultural accumulation. These accumulating agents outperform those trained for a single lifetime with the same cumulative experience. We explore this accumulation by constructing two models under two distinct notions of a generation: *episodic generations*, in which accumulation occurs via *in-context learning* and *train-time generations*, in which accumulation occurs via *in-weights* learning. In-context and in-weights cultural accumulation can be interpreted as analogous to knowledge and skill accumulation, respectively. To the best of our knowledge, this work is the first to present general models that achieve emergent cultural accumulation in reinforcement learning, opening up new avenues towards more open-ended learning systems, as well as presenting new opportunities for modelling human culture.

1 Introduction

The capacity to learn skills and accumulate knowledge over timescales that far outstrip a single lifetime (i.e., across generations) is commonly referred to as *cultural accumulation* [Hofstede et al., 1994, Tennie et al., 2009]. Cultural accumulation has been considered the key to human success [Henrich, 2019], continuously aggregating new skills, knowledge and technology. It also sustains generational improvements in the behaviour of other species, such as in the homing efficiency of birds [Sasaki and Biro, 2017]. The two core mechanisms underpinning cultural accumulation are *social learning* [Bandura and Walters, 1977, Hoppitt and Laland, 2013] and *independent discovery* [Mesoudi, 2011]. Its effectiveness can be attributed to the flexibility with which participants engage in these two mechanisms [Enquist et al., 2008, Rendell et al., 2010, Mesoudi and Thornton, 2018].

*Equal contribution.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-1907

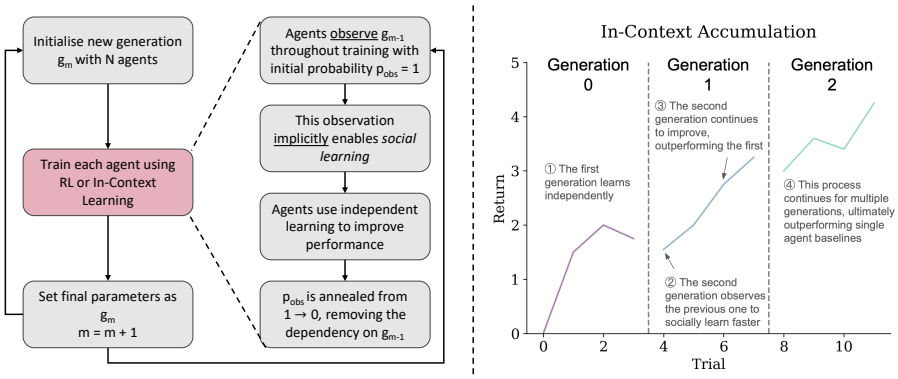

Figure 1: **Left**: A flow chart describing our RL model of cultural accumulation. **Right**: An annotated, illustrative plot demonstrating in-context accumulation as observed in our results.

Cumulative culture thus executes an evolutionary search in the space of behaviours, with higher rates of independent discovery corresponding to a higher mutation rate. Ultimately, how individuals should balance social vs. independent learning depends on the validity and fidelity of social learning signals, i.e., how successful other agents in the environment are at the same task.

Given the profound success of cultural accumulation in nature, it is natural to explore its applicability to artificial learning systems, which remains an under-explored research direction. In deep reinforcement learning (RL), the learning problem is usually framed as taking place over a single "lifetime". Generational methods that do exist, such as iterated policy distillation [Schmitt et al., 2018, Stooke et al., 2022] and expert iteration [Anthony et al., 2017], explicitly optimise new generations with hand-crafted and explicit imitation learning techniques, as opposed to learning accumulation implicitly, which should ultimately enable greater flexibility. Prior work has successfully demonstrated emergent social learning in RL [Borsa et al., 2017, Ndousse et al., 2021, Bhoopchand et al., 2023], showing that it helps agents solve hard exploration problems and adapt online to new tasks. However, these works have only considered one iteration of information transmission from an expert to a student. In this work, we therefore investigate how social learning and exploration can be balanced to achieve cultural accumulation in RL. This would lay the foundations for an open-ended, population-based self-improvement loop among artificial learning agents. It would also establish new tools for modelling cultural accumulation.

In humans, cultural accumulation takes place over a number of timescales due to the different rates at which knowledge, skills and technology are acquired [Perreault, 2012, Cochrane et al., 2023]. We therefore present two corresponding formulations of cultural accumulation in RL: *in-context* accumulation, which operates over fast adaptation to new environments, and *in-weights* accumulation, which operates over the slower process of updating weights (i.e., training). The in-context setting can be interpreted as analogous to short-term knowledge accumulation and the in-weights setting as analogous to long-term, skills-based accumulation.

We demonstrate the effectiveness of both the in-context and in-weights models by showing sustained generational performance gains on several tasks requiring exploration under partial observability[2]. On each task, we find that accumulating agents outperform those that learn for a single lifetime of the same total experience budget. This cultural accumulation emerges purely from individual agents maximising their independent rewards, without any additional losses.

2 Background

2.1 Partially-Observable Stochastic Games

A Partially-Observable Stochastic Game (POSG) [Kuhn, 1953] is defined by the tuple $M = \langle I, S, A, T, R, O, \gamma \rangle$. $I = \{1, \ldots, N\}$ defines the set of agents, S defines the set of states, and A defines the direct product of action spaces for each agent, i.e., $A = A^1 \times \cdots \times A^N$ where A^i is the action space for agent $i \in I$. T defines the stochastic transition function, $T : S \times A \times S \to [0, 1]$, R de-

[2]Code can be found at `https://github.com/FLAIROx/cultural-accumulation`.

fines the reward function, $R : S \times A \to \mathcal{R}$, and O defines the observation function, $O : \mathcal{O} \times S \to [0, 1]$ where $\mathcal{O}$ is the observation space. $\gamma \in (0, 1]$ refers to the discount factor of returns.

At each timestep t, each agent i samples an action from its policy $a_t^i \sim \pi_{\theta^i}(o_t^i)$, where θ_t^i corresponds to the policy parameters for agent i. These actions are aggregated to form the joint action $a_t = (a_t^1, \ldots, a_t^N)$. The environment then calculates the next state $s_{t+1} \sim T(s_t, a_t)$. The agents then receive the next observation $o_{t+1} \sim O(s_{t+1})$ and reward $r_t \sim R(s_{t+1})$ from the environment. The objective for each agent i in a POSG is to maximise its expected discounted sum of returns $J(\pi_{\theta^i}) = \mathbb{E}_{s_0 \sim \tilde{S}, a_{0:\infty} \sim \pi(s^i_{1:\infty}) \sim T}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$.

2.2 Partially-Observable Markov Decision Processes

A Partially-Observable Markov Decision Processes (POMDP) [Kaelbling et al., 1998] is a special case of the POSG with only one agent. For a given agent i in a POSG, if the policy parameters of all other agents θ^{-i} are fixed, the POSG can be reduced to a POMDP for agent i. Informally, this can be done by assuming the other agents are part of the environment. In other words, sampling from other agent policies would be part of the transition function.

To solve POMDPs, we make use of memory-based policy architectures that represent the prior episode interactions at timestep t using ϕ_t. Memory is needed to solve POMDPs because of the state aliasing introduced by partial-observability. For recurrent neural networks (RNNs) ϕ_t refers to the hidden state at timestep t. We now condition the policy on this state $a_t^i \sim \pi_{\theta^i}(o^i|\phi_t^i)$ and also update the hidden state with recurrence function h_{θ^i}, such that $\phi_{t+1} = h_{\theta^i}(o^i|\phi_t^i)$.

2.3 Meta-RL

In this paper, we consider meta-RL settings where an agent is tasked with optimising its return over a distribution of POMDPs $\mathcal{M}$. At each episode[3], a POMDP is sampled $M \sim \mathcal{M}$ and the agent is given K trials to optimise its cumulative return. This setting has also been called in-context RL [Laskin et al., 2022]. We adopt the setting of Ni et al. [2021], where meta-RL is formulated as another POMDP $\mathcal{M}$, in which the selection of M is now a part of the state s. The agent's last action a_{t-1}, reward r_{t-1} and trial termination are appended to the observation o_t.

One common approach to meta-RL uses gradient-based optimisation to train a meta-policy with respect to $\mathcal{M}$ using recurrent network architectures [Duan et al., 2016, Wang et al., 2016]. Within each episode, updates to the agent's internal state ϕ enable static agent parameters to implement an in-context RL procedure, allowing the agent to adapt to a specific M. During adaptation to a new POMDP, this procedure aggregates and stores information from sequential observations. Over long enough episodes, agents can therefore face an in-context exploration-exploitation tradeoff, where within-episode exploration of the environment could yield higher returns.

2.4 Generational Training

Generational training refers to training populations of agents in sequence, thus chaining together multiple learning processes [Stooke et al., 2022]. We define a generation $g_m = \{g_m^1, \ldots, g_m^{N_{\text{pop}}}\}$ as a population of N_{pop} agents. Each g_m^n can therefore be model weights, or an internal state in the case of in-context RL. We distinguish N_{pop} from the N used in Section 2.1, because each member of a generation of size N_{pop} may be within its own POSG of N agents. We use n to index a specific member of a generation. A new generation is generated by a process that conditions on the old generation $g_{m+1} = f(g_m)$. By modelling cultural accumulation over both in-context RL and RL training, f could capture generational accumulation in both knowledge and skills.

3 Problem Statement

In this work, we investigate how to achieve cultural accumulation in RL agents. We formalise a measure of success in Equation 1, where we compare the returns achieved in a held-out set of environments when learning is spread across G total generations to learning in a single lifetime with

[3]Our use of "episode" and "trial" aligns with Bauer et al. [2023], which is the reverse of Duan et al. [2016].

https://doi.org/10.52202/079017-1907

the same total experience budget. This comparison evaluates whether our models capture the benefits of cultural accumulation over singe lifetime learning.

$$R_T(\pi_G, \mathcal{M}|f(\pi_{G-1}|\ldots,\pi_1)) > R_{G\cdot T}(\pi_1, \mathcal{M}) \quad (1)$$

Policies are parameterised according to $\pi_\theta(\cdot|\phi)$ where θ are parameters of the trained network and ϕ is the agent's internal state. We define **in-context accumulation** as cultural accumulation during online adaptation to new environments. In this setting, θ are frozen and T is the length of an episode. We assume that a meta-RL training process has produced θ and that the internal state ϕ is used to distinguish between generations, which we elaborate on in Section 4.1.2. We define **in-weights accumulation** as cultural accumulation over training runs. Here, T is the number of environment steps used for training each generation and each successive generation is trained from randomly initialised parameters θ, meaning that θ instead are the substrate for accumulation.

3.1 Environments

Memory Sequence: To study processes of cultural accumulation among humans, Cornish et al. [2017] use a random sequence memorisation task in which sequences recalled by one participant become training data for the next in an iterated learning process. In doing so, they show a cumulative increase in recalled string length, thus simulating cultural accumulation. We thus present *Memory Sequence* as a simple environment for investigating basic cultural accumulation in RL agents. In *Memory Sequence*, there is an arbitrary sequence made up of digits that agents must infer over the course of in-context learning or training. Each action corresponds to predicting a digit. Episodes are fixed length and agents receive $+1$ reward for predicting the correct next digit and -1 reward for predicting an incorrect next digit. The sequence is randomly sampled and fixed during in-context adaptation for in-context accumulation, whilst it is fixed across training for in-weights accumulation.

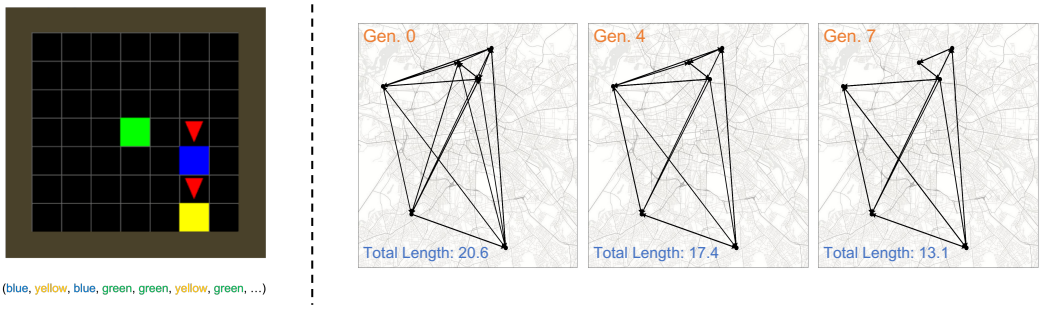

Figure 2: **Left**: A visualization of the Goal Sequence Environment. **Right**: Routes travelled get shorter across generations in the TSP environment. Visualisation implementation is based on Jumanji [Bonnet et al., 2024].

Goal Sequence: Previous work on social learning in RL uses a gridworld *Goal Cycle* environment [Ndousse et al., 2021]. In this environment, there are n identical goals and agents must figure out the correct cyclic ordering in which to navigate through them. Agents receive an egocentric partial observation of the environment state on each timestep. Other prior work uses a 3-dimensional *Goal Cycle* environment [Bhoopchand et al., 2023]. *Goal Cycle* is limited in its ability to evaluate generational improvement across learned agent behaviours, because once an agent has identified the correct cyclic order, it should simply repeat the same action sequence of cycling through the goals for the duration of an episode. We therefore introduce *Goal Sequence* (Figure 2) as a simple adaptation of *Goal Cycle*, but with more open-ended properties. Specifically, we replace the cycle with a non-repeating sequence of goals that agents must discover by navigating based on egocentric, partial observations. We use a 7×7 grid with 3 goal types and agents observe a 3×3 grid of cells directly in front of them. For in-context accumulation, the goal sequence and positions are randomly generated and fixed across in-context adaptation. For in-weights accumulation, the goal sequence is fixed across training, but goal positions are still randomised between episodes.

Travelling Salesperson: Finally, we use the Travelling Salesperson Problem (TSP). Given that cultural accumulation involves the transmission of information not otherwise immediately accessible

to new generations, we specifically focus on the partially-observable variant of TSP [Buck and Keller, 2008, Noormohammadi-Asl and Taghirad, 2019] in which some or all of the cities' coordinates are not observed. The agent's objective is to visit every city, minimising the total length of the route travelled. The positions of cities are randomly generated and fixed across in-context adaptation for in-context accumulation. City positions are fixed across all of training for in-weights accumulation.

4 Cultural Accumulation in RL

We model cultural accumulation as taking place within POSGs. Each member of a given generation g_{m+1} learns in parallel on an independent instance of the environment, which also contains *static* members of the previous generation, g_m. Therefore, from the perspective of a given agent n in g_{m+1} this is a POMDP, since the prior, static generation can be considered part of the environment. In this work, we focus on *generational improvement* and have thus introduced a separation between the *development phase* of an agent, where they are learning from and improving upon prior generations, and the *transmission phase*, where they are simply acting and being observed by the next generation. Under this formulation, we propose and investigate two distinct mechanisms for cultural accumulation in RL agents based on our *in-context* and *in-weights* dichotomy introduced in Section 3.

4.1 In-Context Accumulation

We first consider in-context accumulation, where we aim to achieve cultural accumulation via fast in-context adaptation following a separate phase of meta-RL training. Meta-RL training can be viewed as imbuing agents with multiple memory systems [Squire et al., 1993]. By updating ϕ, in-context RL executes a processes of knowledge acquisition, or *declarative* memory [Squire, 2004] updating. In-context accumulation extends this knowledge acquisition to operate across multiple generations. The corresponding results are in Section 5.1.

4.1.1 Training

Algorithm 1 Training Loop for In-Context Accumulation (changes to RL^2 in red)

$\hat{\phi}$:= init. agent state
$\hat{\theta}$:= init. agent parameters
θ^0 := oracle parameters
ϵ := oracle noise factor
$\theta^n \leftarrow \hat{\theta}$ // Initialise the agent parameters
for train step $t \in [0, T-1]$ **do**
  $s \sim \hat{S}$
  $\phi^n \leftarrow \hat{\phi}$
  B := trajectory buffer
  for each trial $k \in [0, K-1]$ **do**
    $p_{\text{obs}} = 1 - k/(K-1)$
    while trial not done **do**
      $o^n \sim O(s)$
      if `IsVisible` $\sim$ Bernoulli(p_{obs}) **then**
        Set oracle visible in o^n
      end if
      $a^0 \sim \pi_{\theta^0,\epsilon}(s)$, $a^n \sim \pi_{\theta^n}(o^n|\phi^n)$
      $\phi^n \leftarrow f_{\theta^n}(o^n|\phi^n)$
      $s, r \sim T(s, a)$
      $B \leftarrow (o^n, a^n, r^n)$ // Append transition to buffer
    end while
  end for
  $\theta^n \leftarrow \text{update}(\theta^n, B)$
end for

To train agents capable of in-context accumulation, we require three attributes: **(1)** agents must learn to learn from the behaviour of other agents (i.e., social learning) within the context of a single episode,

https://doi.org/10.52202/079017-1907

(2) agents must be able to act independently by their last trial, so that their behaviour can provide useful demonstrations for the next generation, **(3)** agents must use sufficient independent exploration to improve on the the previous generation, or each new generation will only be as good as the last.

To facilitate **(1)** during training, but *not* at test time, we assume access to an oracle `O` with fixed parameters $\hat{\theta}^{\texttt{O}}$. `O` is able to condition on the full state s, which we refer to as *privileged information*. Agent n can observe the behaviours of `O` as they jointly act in an environment. To achieve **(2)**, we include this representation of `O` in o^n with probability p_{obs} at each timestep and linearly anneal p_{obs} across the K trials from $1 \rightarrow 0$. This ensures that the agent learns to act independently by the final trial. Perez et al. [2023] show that increasing opportunities to learn socially leads to less diversity in homogeneous populations, which could be seen as limiting independent discovery in our context, further motivating this constraint. For **(3)**, we add some random noise to the oracle policy according to a tunable parameter ϵ[4]. This encourages the agent to learn that the behaviour of other agents `O` in the environment may be sub-optimal, in which case agent n should engage in *selective* social learning [Poulin-Dubois and Brosseau-Liard, 2016]. This approach could be seen as imposing an information rate limit [Prystawski et al., 2023] between the oracle and learning agent. We present this process altogether as Algorithm 1, with changes to standard RL^2 in red.

4.1.2 Evaluation

Algorithm 2 In-Context Accumulation During Evaluation

$\hat{\phi} :=$ init. agent state
$\theta :=$ parameters from training
$s_0 \sim \hat{S}$ // Sample initial state
$\tilde{\phi}_0^{n^*} \leftarrow \hat{\phi}$ // Define an initial reset state
for each generation $m \in [1, M]$ **do**
  for each population member $n \in [1, N_{\text{pop}}]$ **do**
    $\phi_m^n \leftarrow \hat{\phi}$
    $s^n \leftarrow s_0$
    for each trial $k \in [0, K-1]$ **do**
      if $k = K-1$ **then**
        $\tilde{\phi}_m^n \leftarrow \phi_m^n$ // Store agent state at beginning of last trial as reset state
      end if
      $p_{\text{obs}} = 1 - k/(K-1)$
      $\phi_{m-1}^n \leftarrow \tilde{\phi}_{m-1}^{n^*}$ // Set previous generation state to best reset state from that generation
      while trial not done **do**
        $o_{m-1}^n, o_m^n \sim O(s)$
        if `IsVisible` $\sim$ Bernoulli(p_{obs}) **then**
          Set previous generation visible in o_m^n
        end if
        $a_{m-1}^n \sim \pi_\theta(o_{m-1}^n | \phi_{m-1}^n)$, $a_m^n \sim \pi_\theta(o_m^n | \phi_m^n)$
        $\phi_{m-1}^n \leftarrow f_\theta(o_{m-1}^n | \phi_{m-1}^n)$, $\phi_m^n \leftarrow f_\theta(o_m^n | \phi_m^n)$
        $s^n, r^n \sim T(s^n, a^n)$
      end while
    end for
    $n^* = \text{argmax}_{n \in [1, N_{\text{pop}}]} \sum_t \mathbb{I}[k = K-1] r_m^n$ // Select best population member based on final trial performance
  end for
end for

Having trained agents via Algorithm 1, we evaluate their in-context accumulation. During this evaluation phase, `O` is replaced by the best[5] of the previous generation n^*, as we do not assume

[4]For the oracle to capture sub-optimal, but goal-directed behaviour, we inject correlated noise by corrupting the privileged information in $o^{\texttt{O}}$.

[5]This is an external selection mechanism. We investigate the emergence of selective social learning in Appendix C.

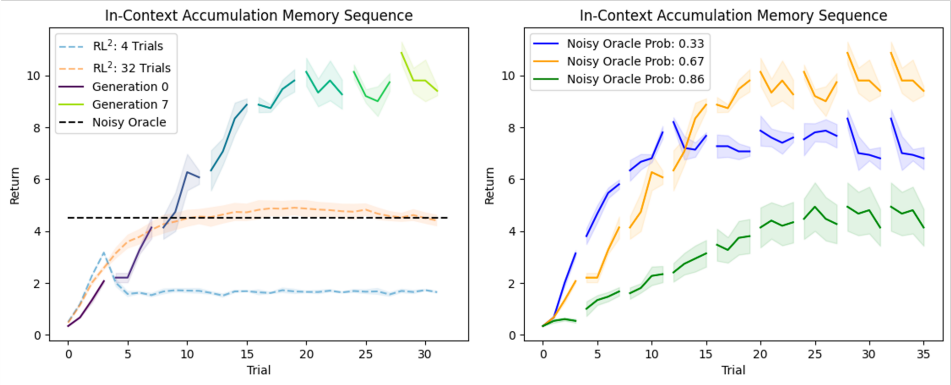

Figure 3: **Left**: In-context accumulation during evaluation on *Memory Sequence*. **Right**: Evaluation results following training with different oracle accuracies.

access to privileged information at test time. This previous best agent, now in *transmission phase*, has its internal state reset between trials. A member of the new generation, in *development phase*, continuously updates its internal state across trials. The same meta-parameters θ from training are used across generations and only the internal states ϕ are used to update each generation. Generations are made up of N agents that interact with N independent environments.

For each generation g_m, we store the internal state of each agent on the first step of its final trial and call this the "reset state" $\tilde{\phi}_m^n$. After the final trial, we then keep the previously stored reset state of the best performing agent $\tilde{\phi}_m^{n^*}$ in terms of rewards received in its last trial. When introducing the next generation g_{m+1}, generation g_m transitions into transmission phase and we represent the adapted behaviour of the best performing agent by resetting the internal states $\phi_m^{1:N} \leftarrow \tilde{\phi}_m^{n^*}$ at the beginning of each trial. Thus, the next generation g_{m+1} can in essence observe the best of g_m repeating its final trial. In practice, we mask g_{m+1} from the observations of g_m in order to reflect the fact that agents do not expect to see other agents in their final trial, as guaranteed by training attribute **(2)** above. Since the observations of g_{m+1} can include information about the learned behaviours of g_m, we implicitly have $g_{m+1} = f(g_m)$, establishing a purely in-context equivalent to generational training (Section 2.4). This process is presented in Algorithm 2.

4.2 In-Weights Accumulation

Finally, we present an algorithm for cultural accumulation to take place over the course of training successive generations of policies (i.e., in-weights) rather than through solely updating the internal state (i.e., in-context). This is akin to generational training (Section 2.4), but without modifying the policy-gradient loss, by simply including the previous generation within environments on which the new generation is training, so that their actions can be observed and learned from. This alternative model of cultural accumulation assumes an agent's lifetime to be an entire training run, rather than a single episode. Here, we anneal the probability of observing the previous generation p_{obs} on a given environment timestep linearly over training. The corresponding approach is detailed in Algorithm 3, presented in Appendix A. In-weights RL (i.e., RL training) can be interpreted as updating *procedural* memory [Johnson, 2003] via skill acquisition. We thus interpret in-weights accumulation as gradual, skills-based accumulation. Note that there is no separation between "training" and "evaluation" in this case, because the in-weights algorithm views cultural accumulation as part of the training. This is in contrast to in-context accumulation, which is an evaluation-time algorithm executed by appropriately trained meta-policies. The results that correspond to this setting are in Section 5.2.

5 Results

For each of our experiments, we use a Simplified Structured State Space Model (S5) [Smith et al., 2022] modified for RL [Lu et al., 2024] to encode memory, building on the PureJaxRL codebase [Lu et al., 2022a]. This model architecture runs asymptotically faster than Transformers in sequence length and outperforms RNNs in memory tasks [Morad et al., 2024]. PPO [Schulman et al., 2017] is

https://doi.org/10.52202/079017-1907

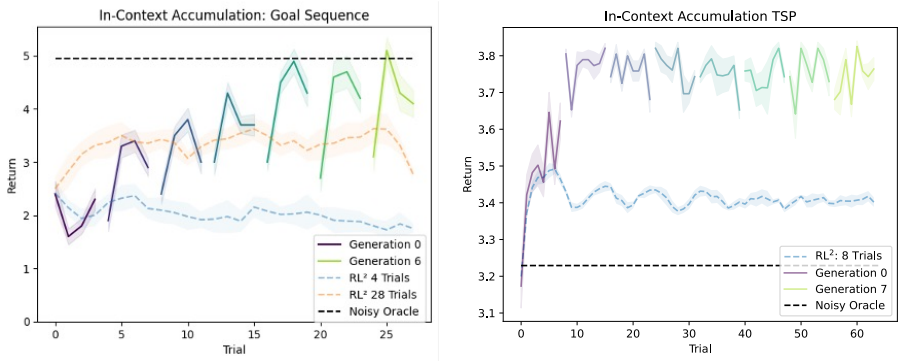

Figure 4: **Left**: In-context accumulation during evaluation on *Goal Sequence*. **Right**: In-context accumulation during evaluation on TSP.

used as the RL training algorithm. For *Goal Sequence* experiments, observations are processed by a CNN, whilst for *Memory Sequence* and TSP, they are processed by feed forward layers. Architectural details are provided in Appendix E and algorithmic hyperparameters in Appendix F. For all in-context experiments, a one-hot encoding of the reward received on each timestep and a one-hot encoding of the current trial number are included each agent's observation. We provide further details on the configuration of each environment in Appendix D. All results are averaged across 10 seeds and the shaded regions of plots show standard error.

5.1 In-Context Results

As single-lifetime baselines, we use RL^2 trained on: (a) episodes of equivalent length to one generation and (b) episodes of equivalent length to all cumulative generations. All in-context evaluations are performed on held-out environment instances for the same number of trials.

Memory Sequence: In Figure 3, we show that in-context learners trained according to Algorithm 1 are capable of accumulating beyond single-lifetime RL^2 baselines and even beyond the performance of the noisy oracles with which they were trained when evaluated on a new sequence. Interestingly, when evaluating the accumulation performance of agents trained with oracles of different noise levels, we see that it degrades when oracles are too accurate. This is directly inverse to results in imitation learning [Sasaki and Yamashina, 2021], where the quality of expert trajectories positively impacts the performance achieved when training on these trajectories. We attribute this result to an over-reliance on social learning when oracles are too accurate, which therefore impedes the progress of independent in-context learning during training. Conversely, if oracles are too random, agents do not acquire the necessary social learning skills to effectively make use of prior generations.

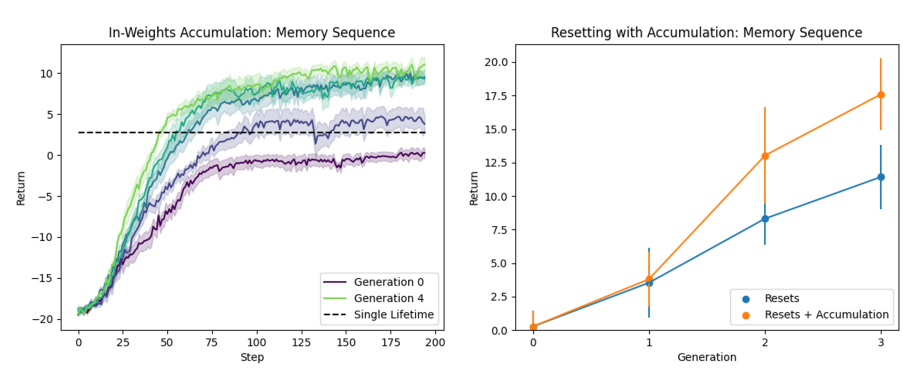

Figure 5: **Left**: In-weights accumulation on *Memory Sequence*. **Right**: In-weights accumulation compounds with resetting. Error bars represent 95% confidence intervals.

Goal Sequence: Figure 4 (left) shows that in-context accumulation also significantly outperforms single-lifetime RL^2 when evaluated on a new goal sequence. On this more challenging, partially-observable navigation task, we found that higher, but still imperfect oracle accuracies during training produced the most effective accumulating agents. This is likely due to the fact that learning to learn socially in this environment is a much harder problem that involves finding and actively following another agent. We additionally observe that the performance of each generation typically drops slightly on the last trial or two, where agents are entirely or mostly independent. This indicates that although agents are able to independently recall and navigate most of the sequence they have learned, improving on the previous generation, they are still somewhat over-reliant on demonstrations.

TSP: We present these results in Figure 4 (right). Again, we see that cultural accumulation enables sustained improvements considerably beyond RL^2 over a single continuous context (i.e., lifetime). We only show the RL^2 baseline trained on shorter episodes, as the baseline trained on longer episodes failed to make meaningful learning progress. In Figure 2, we show routes traversed by different generations for an example set of city locations. Notably, these routes become more optimised across generations, with later generations exploiting a decreasing subset of edges.

5.2 In-Weights Results

Memory Sequence: Figure 5 (left) shows that agents can accumulate over the course of training via Algorithm 3. For in-weights accumulation, the sequence is kept fixed over training, meaning that agents are trained to learn as much of a single sequence as possible. After one generation of accumulation, these agents outperform single-lifetime training for a duration equivalent to 5 complete generations. This demonstrates that single-lifetime learners succumb to primacy bias [Nikishin et al., 2022] and converge prematurely. In light of this, we investigate whether in-weights accumulation can be paired with other methods to overcome primacy bias. In particular, we use the simple approach of resetting some policy network layers [Nikishin et al., 2022], opting for resetting the final two feed-forward layers, and present these results in Figure 5 (right). We observe that resetting alone helps mitigate premature convergence, but that when we apply resetting to accumulating agents[6], the improvements are compounded and agents reach higher returns than via either method independently.

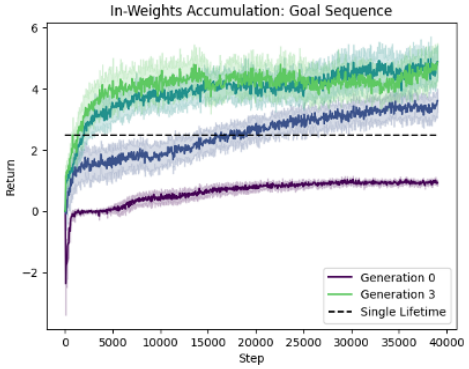

Figure 6: In-weights accumulation on *Goal Sequence*.

Goal Sequence: Figure 6 shows that in-weights accumulation exceeds the performance of training a single agent for a continuous run equivalent to the total duration of 4 accumulation steps.

TSP: We show results for in-weights accumulation in TSP in Figure 7, within Appendix B.

6 Related Work

Social Learning: Considerable focus has been placed on using social learning to overcome hard exploration problems, imitate human behaviour, and generalise to unseen environments [Ndousse et al., 2021, Bhoopchand et al., 2023, Ha and Jeong, 2023]. Ndousse et al. [2021] trains agents that achieve reasonable performance without a demonstrator by increasing the proportion of episodes with no demonstrator in handpicked increments over training. We employ a similar approach for training agents to eventually act independently by annealing the probability of observing the previous generation across training. To obtain independently capable agents in-context, we anneal the probability of observing the previous generation between trials, which is similar to the expert dropout used in Bhoopchand et al. [2023].

Machine Learning Models of Cumulative Culture: A growing body of work uses advances in machine learning to model cultural accumulation *in-silico* [Prystawski et al., 2023, Perez et al., 2023,

[6]We initialise the second generation entirely from scratch so that it can learn to learn socially and use resetting with accumulation thereafter.

https://doi.org/10.52202/079017-1907

	Explicit Imitation Learning	Implicit Third-Person Social Learning
Non-Generational	Behavioral Cloning [Pomerleau, 1988] Inverse RL [Russell, 1998]	Observational Learning by RL [Borsa et al., 2017] Emergent Social Learning [Ndousse et al., 2021] Learning Few-Shot Imitation [Bhoopchand et al., 2023]
Generational Training	Kickstarting [Schmitt et al., 2018] X-Land [Stooke et al., 2022]	**Our Work**

Table 1: Comparisons to prior work. Generational Training helps overcome local optima to tackle open-ended tasks [Stooke et al., 2022]. Implicit third-person social learning is more general than hand-crafted imitation learning algorithms. Our work combines both to achieve cultural accumulation.

2024]. Perez et al. [2024] use Bayesian RL with constrained inter-generational communication to reproduce social learning processes documented in human populations. This communication is represented in a domain specific language. Rather than requiring an explicit communication channel between agents, we use social learning to facilitate accumulation in a more domain agnostic manner, demonstrating performance gains in multiple distinct settings. Prystawski et al. [2023] demonstrate cultural evolution in populations of Large Language Models, where language-based communication is used as the mechanism for knowledge transfer between generations.

Generational RL: The generational process of cultural accumulation can be seen as iterations of *implicit* policy distillation and improvement. Iterative policy distillation [Schmitt et al., 2018, Stooke et al., 2022] is therefore related to our work. Unlike these methods, we do not assume that the new learner has access to the policies of experts or past generations, but only observations of their behaviour as they interact with a shared environment. We also do not explicitly modify the learning objective, leaving the agent free to learn how much or how little to imitate past generations.

7 Conclusion

We take inspiration from the widespread success of cumulative culture in nature and investigate cultural accumulation in reinforcement learning. We construct an in-context model of accumulation, which operates on episodic timescales, and an in-weights model, which operates over entire training runs. We define successful cultural accumulation as a generational process that exceeds the performance of independent learning with the same total experience budget. We then present in-context and in-weights algorithms that give rise to successful cultural accumulation on several tasks requiring exploration under partial observability. We find that in-context accumulation can be impeded by training agents with oracles that are either too reliable or too unreliable, highlighting the need to balance social learning and independent discovery. We also show that in-weights accumulation effectively mitigates primacy bias and is further improved by network resets.

Limitations and Future Work: An interesting future direction would be to use learned auto-curricula [Dennis et al., 2020, Jiang et al., 2021] for deciding when agents should learn socially or independently, instead of linearly annealing the observation probability. Another extension would be to investigate cumulative culture in settings with different incentive structures, such as heterogeneous or cooperative rewards [Rutherford et al., 2023]. Future work could also study ways we can evolve the process of cultural accumulation itself [Lu et al., 2023a] or learn to influence the learning process of other agents [Lu et al., 2022b, 2023b] for cultural transmission. Finally, we note that whilst understanding cumulative culture has benefits for both machine learning and modelling human society, the consequences of self-improving systems should warrant consideration.

Acknowledgments

Jonathan Cook is supported by the ESPRC Centre for Doctoral Training in Autonomous Intelligence Machines and Systems EP/S024050/1. Jakob Foerster is partially funded by the UKI grant EP/Y028481/1 (originally selected for funding by the ERC). Jakob Foerster is also supported by the JPMC Research Award and the Amazon Research Award.

References

Thomas Anthony, Zheng Tian, and David Barber. Thinking fast and slow with deep learning and tree search, 2017.

Albert Bandura and Richard H Walters. *Social learning theory*, volume 1. Englewood cliffs Prentice Hall, 1977.

Jerome H Barkow, Akinsola A Akiwowo, Tushar K Barua, MRA Chance, Eliot D Chapple, Gouranga P Chattopadhyay, Daniel G Freedman, WR Geddes, BB Goswami, PAC Isichei, et al. Prestige and culture: a biosocial interpretation [and comments and replies]. *Current Anthropology*, 16(4):553–572, 1975.

J. Bauer, K. Baumli, S. Baveja, et al. Human-timescale adaptation in an open-ended task space. In *International Conference on Machine Learning*, 2023.

A. Bhoopchand, B. Brownfield, A. Collister, et al. Learning few-shot imitation as cultural transmission. In *Nature Communications 14*, page 7536, 2023.

Clément Bonnet, Daniel Luo, Donal Byrne, Shikha Surana, Sasha Abramowitz, Paul Duckworth, Vincent Coyette, Laurence I. Midgley, Elshadai Tegegn, Tristan Kalloniatis, Omayma Mahjoub, Matthew Macfarlane, Andries P. Smit, Nathan Grinsztajn, Raphael Boige, Cemlyn N. Waters, Mohamed A. Mimouni, Ulrich A. Mbou Sob, Ruan de Kock, Siddarth Singh, Daniel Furelos-Blanco, Victor Le, Arnu Pretorius, and Alexandre Laterre. Jumanji: a diverse suite of scalable reinforcement learning environments in jax, 2024. URL `https://arxiv.org/abs/2306.09884`.

Diana Borsa, Bilal Piot, Rémi Munos, and Olivier Pietquin. Observational learning by reinforcement learning. *arXiv preprint arXiv:1706.06617*, 2017.

A. R. Buck and J. M. Keller. A myopic monte carlo strategy for the partially observable travelling salesman problem. 2008.

Aaron Cochrane, Chris R Sims, Vikranth R Bejjanki, C Shawn Green, and Daphne Bavelier. Multiple timescales of learning indicated by changes in evidence-accumulation processes during perceptual decision-making. *npj Science of Learning*, 8(1):19, 2023.

H. Cornish, R. Dale, S. Kirby, and M. H. Christiansen. Sequence memory constraints give rise to language-like structure through iterated learning. In *PLoS ONE*, 2017.

Michael Dennis, Natasha Jaques, Eugene Vinitsky, Alexandre Bayen, Stuart Russell, Andrew Critch, and Sergey Levine. Emergent complexity and zero-shot transfer via unsupervised environment design. *Advances in neural information processing systems*, 33:13049–13061, 2020.

Y. Duan, J. Schulman, X. Chen, P. L. Bartlett, I. Sutskever, and P. Abbeel. RL^2: Fast reinforcement learning via slow reinforcement learning. arXiv preprint, arXiv:1611.02779, 2016.

M. Enquist, S. Ghirlanda, and A. Jarrick. Why does human culture increase exponentially? 2008.

Seungwoong Ha and Hawoong Jeong. Social learning spontaneously emerges by searching optimal heuristics with deep reinforcement learning. In *International Conference on Machine Learning*, pages 12319–12338. PMLR, 2023.

J. Henrich. The secret of our success: How culture is driving human evolution, domesticating our species, and making us smarter. Princeton University Press, 2019.

G. Hofstede, G. J. Hofstede, and M. Minkov. Cultures and organizations: Software of the mind. McGraw-Hill Professional, 1994.

William Hoppitt and Kevin N Laland. *Social learning: an introduction to mechanisms, methods, and models*. Princeton University Press, 2013.

Victoria Horner, Darby Proctor, Kristin E Bonnie, Andrew Whiten, and Frans BM de Waal. Prestige affects cultural learning in chimpanzees. *PloS one*, 5(5):e10625, 2010.

 https://doi.org/10.52202/079017-1907

Minqi Jiang, Michael Dennis, Jack Parker-Holder, Jakob Foerster, Edward Grefenstette, and Tim Rocktäschel. Replay-guided adversarial environment design. *Advances in Neural Information Processing Systems*, 34:1884–1897, 2021.

Addie Johnson. Procedural memory and skill acquisition. *Handbook of psychology*, pages 499–523, 2003.

L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. 101:99–134, 1998.

H. W. Kuhn. Extensive games and the problem of information. 28:193–216, 1953.

M. Laskin, L. Wang, J. Oh, et al. In-context reinforcement learning with algorithm distillation. In *Advances in Neural Information Processing Systems*, 2022.

Chris Lu, Jakub Kuba, Alistair Letcher, Luke Metz, Christian Schroeder de Witt, and Jakob Foerster. Discovered policy optimisation. *Advances in Neural Information Processing Systems*, 35:16455–16468, 2022a.

Chris Lu, Sebastian Towers, and Jakob Foerster. Arbitrary order meta-learning with simple population-based evolution. In *ALIFE 2023: Ghost in the Machine: Proceedings of the 2023 Artificial Life Conference*. MIT Press, 2023a.

Chris Lu, Timon Willi, Alistair Letcher, and Jakob Nicolaus Foerster. Adversarial cheap talk. In *International Conference on Machine Learning*, pages 22917–22941. PMLR, 2023b.

Chris Lu, Yannick Schroecker, Albert Gu, Emilio Parisotto, Jakob Foerster, Satinder Singh, and Feryal Behbahani. Structured state space models for in-context reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

Christopher Lu, Timon Willi, Christian A Schroeder De Witt, and Jakob Foerster. Model-free opponent shaping. In *International Conference on Machine Learning*, pages 14398–14411. PMLR, 2022b.

Alex Mesoudi. Variable cultural acquisition costs constrain cumulative cultural evolution. *PloS one*, 6(3):e18239, 2011.

Alex Mesoudi and Alex Thornton. What is cumulative cultural evolution? *Proceedings of the Royal Society B*, 285(1880):20180712, 2018.

Steven Morad, Chris Lu, Ryan Kortvelesy, Stephan Liwicki, Jakob Foerster, and Amanda Prorok. Revisiting recurrent reinforcement learning with memory monoids. *arXiv preprint arXiv:2402.09900*, 2024.

K. Ndousse, D. Eck, S. Levine, and N. Jaques. Emergent social learning via multi-agent reinforcement learning. In *International Conference on Machine Learning*, 2021.

T. Ni, B. Eysenbach, and R. Salakhutdinov. Recurrent model-free rl can be a strong baseline for many pomdps. 2021.

E. Nikishin, M. Schwarzer, P. D'Oro, P-L. Bacon, and A. Courville. The primacy bias in deep reinforcement learning. In *International Conference on Machine Learning*, 2022.

A. Noormohammadi-Asl and H. D. Taghirad. Multi-goal motion planning using traveling salesman problem in belief space. 2019.

J. Perez, M. Sánchez-Fibla, and C. Moulin-Frier. Cultural evolution in populations with heterogenous and variable preferences. 2023. URL `https://inria.hal.science/hal-04408019`.

J. Perez, C. Léger, M. Ovando-Tellez, C. Foulon, J. Dussauld, P. Oudayer, and C. Moulin-Frier. Cultural evolution in populations of large language models. 2024.

Charles Perreault. The pace of cultural evolution. 2012.

Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

D. Poulin-Dubois and P. Brosseau-Liard. The developmental origins of selective social learning. *Current Directions in Psychological Science*, 25(1):60–64, 2016.

B. Prystawski, D. Arumugam, and N. D. Goodman. Cultural reinforcement learning: a framework for modeling cumulative culture on a limited channel. 2023.

Luke Rendell, Robert Boyd, Daniel Cownden, Marquist Enquist, Kimmo Eriksson, Marc W Feldman, Laurel Fogarty, Stefano Ghirlanda, Timothy Lillicrap, and Kevin N Laland. Why copy others? insights from the social learning strategies tournament. *Science*, 328(5975):208–213, 2010.

Stuart Russell. Learning agents for uncertain environments. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 101–103, 1998.

Alexander Rutherford, Benjamin Ellis, Matteo Gallici, Jonathan Cook, Andrei Lupu, Gardar Ingvarsson, Timon Willi, Akbir Khan, Christian Schroeder de Witt, Alexandra Souly, et al. Jaxmarl: Multi-agent rl environments in jax. *arXiv preprint arXiv:2311.10090*, 2023.

F. Sasaki and R. Yamashina. Behavioural cloning from noisy demonstrations. In *International Conference on Learning Representations*, 2021.

T. Sasaki and D. Biro. Cumulative culture can emerge from collective intelligence in animal groups. 2017.

Simon Schmitt, Jonathan J Hudson, Augustin Zidek, Simon Osindero, Carl Doersch, Wojciech M Czarnecki, Joel Z Leibo, Heinrich Kuttler, Andrew Zisserman, Karen Simonyan, and S. M. Ali Eslami. Kickstarting deep reinforcement learning. *arXiv preprint, arXiv:1803.03835*, 2018.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. arXiv preprint, arXiv:1707.06347, 2017.

J. T. H. Smith, A. Warrington, and S. W. Linderman. Simplified state space layers for sequence modelling. arXiv preprint, arXiv:2208.04933, 2022.

Larry R Squire. Memory systems of the brain: a brief history and current perspective. *Neurobiology of learning and memory*, 82(3):171–177, 2004.

Larry R Squire, Barbara Knowlton, and Gail Musen. The structure and organization of memory. *Annual review of psychology*, 44(1):453–495, 1993.

A. Stooke, A. Mahajan, C. Barros, et al. Open-ended learning leads to generally capable agents. arXiv preprint, arXiv:2107.12808, 2022.

C. Tennie, J. Call, and M. Tomasello. Ratcheting up the ratchet: On the evolution of cumulative culture. In *Philosophical Transactions of the Royal Society B: Biological Sciences*, 2009.

Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*, 2016.

 https://doi.org/10.52202/079017-1907

A In-Weights Accumulation

Algorithm 3 In-Weights Accumulation

$\hat{\theta}$:= init. agent parameters
$\tilde{\theta}_0^{n^*} \leftarrow \hat{\theta}$ // Initialise the "random" first generation parameters
for each generation $m \in [1, G]$ **do**
  for each population member $n \in [1, N_{\text{pop}}]$ **do**
    $B :=$ trajectory buffer
    $\theta_m^n \sim \hat{\theta}$
    for train step $t \in [0, T-1]$ **do**
      $p_{\text{obs}} = 1 - t/(T-1)$
      $s \sim \hat{S}$
      while not done **do**
        $o_{m-1}^n, o_m^n \sim O(s)$
        if $p \sim$ Bernoulli(p_{obs}) **then**
          Set previous generation visible in o_m^n
        end if
        $a_{m-1}^n \sim \pi_{\tilde{\theta}_{m-1}^{n^*}}(o_{m-1}^n)$
        $a^n \sim \pi_{\theta_m^n}(o_m^n)$
        $s, r \sim T(s, a)$
        $B \leftarrow (o_m^n, a_m^n, r_m^n)$ // Append transition to buffer
      end while
      $\theta_m^n \leftarrow$ update(θ_m^n, B)
    end for
    $n^* = \text{argmax}_{n \in [1, N_{\text{pop}}]} \sum_{t=T-1} r_m^n$ // Select best population member based on final episode performance
  end for
end for

B In-Weights Accumulation in TSP

In this setting, after two generations, performance exceeds agents trained for a single lifetime equivalent to 5 generations. However, we do not observe as much improvement on subsequent generations.

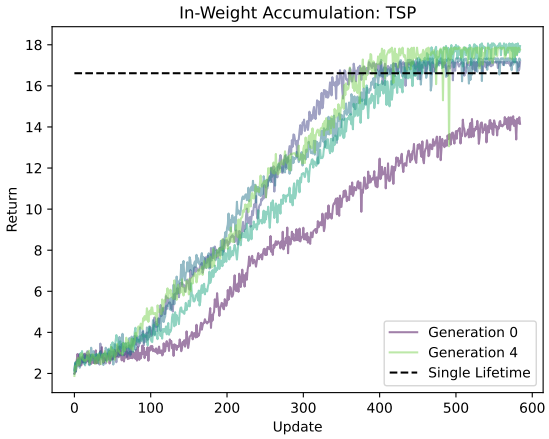

Figure 7: In-weights accumulation on TSP.

https://doi.org/10.52202/079017-1907

C Selective Social Learning

In the Goal Sequence experiments, we select the *best performing* agent of the last generation for the current generation of agents to observe during training, automating the selection process. In human and animal cultural accumulation, this selection is instead *learned* through *prestige cues* Barkow et al. [1975], Horner et al. [2010]. Thus, in the Memory Sequence experiments, we do not automatically select the best of the past generation for the agents to observe. Instead, the agents can observe the *entire* last generation. For In-Context Accumulation, we sort the observed oracles and agents by their performance during pre-training and evaluation. Thus, the agents ideally learn to weight the last generation by their relative performance. For In-Weights Accumulation, sorting does not make a difference, as random network initialization is agnostic to the observation ordering. Thus, the agent learns implicitly through observation, which of the past generation is the best to imitate.

D Further Environment Details

D.1 Memory Sequence

For in-context accumulation in Memory Sequence, we use sequences comprised of three digits, with a maximum length of 24, and give the agents four trials per episode. Each generation has a population size of three. We perform initial meta-training for 8e6 timesteps. Agents get a reward of 1.0 for a correct response and a reward for -1.0 for an incorect one. The trial ends when an incorrect response is given or the maximum length is reached.

For in-weight accumulation in Memory Sequence we use sequences comprising of ten digits, with a maximum length of 24. Agents train for 1e5 timesteps. Each generation has a population size of five. Agents get a reward of 1.0 for a correct response and a reward for -1.0 for an incorect one. The episode ends when the maximum length is achieved.

D.2 Goal Sequence

We use 7×7 grids with 3 goal types. Agents observe a 3×3 grid of cells directly in front of them. Observations are symbolic with 4 channels (one for each goal type and one for agents). On each timestep, an agent can take one of three actions: `move forward`, `turn left`, `turn right`. Agents receive $+1$ reward for hitting the correct next goal and -1 reward for hitting the incorrect next goal. For in-context experiments, trials are 30 steps long and there are 4 trials in an episode. For in-weights experiments, episodes are 50 steps long.

D.3 TSP

For in-context accumulation in TSP, we use six cities and provide eight trials per episode. The city positions are uniformly sampled in the unit square. Each generation has a population size of three. We perform initial meta-training for 8e6 timesteps. Agents get a reward of $\frac{(\sqrt{2}-\text{dist(cur city,next city)})}{\sqrt{2}}$ if the selected next city is valid. Agents get a reward of -1.0 and the trial ends if they select an already-visited city.

For in-weight accumulation in TSP, we use twenty-four cities. Each generation has a population size of eight. Agents train for 3e5 timesteps. The reward setup is equivalent to above.

https://doi.org/10.52202/079017-1907

E Architecture Details

E.1 *Memory Sequence* and TSP

The policy and value networks share two fully layers and a GRU layer. Values and policies are computed with two fully connected layers. All layers use leaky ReLU activations. Here, we take the example of a 4-dimensional input for the 4-dimensional embedding of an action observation and assume the in-weights setting, so no trials. The final output is 10-dimensional in the case of a random sequence made up of 10 distinct digits.

- Shared input layers:
  - FC (4, 128)
  - FC (128, 256)
  - FC (256, 256)
  - GRU (256, 256)
- Value MLP:
  - FC (256, 128)
  - FC (128, 128)
  - FC (128, 1)
- Policy MLP:
  - FC (256, 128)
  - FC (128, 128)
  - FC (128, 10)

E.2 *Goal Sequence*

The policy and value networks share three convolutional layers, a fully connected layer, and an S5 layer. Values and policies are computed with two fully connected layers. Convolutions use leaky ReLU activation functions and all other layers use tanh activation functions. Inputs are 5-dimensional for one-hot encodings of 3 goal types, walls and agents. A 3-dimensional vector for one-hot encodings of reward is concatenated to the output of the last convolutional layer. A one-hot encoding of the trial would also be included in the in-context setting.

- Shared input layers:
  - Conv (5, 32), 1×1 filters, stride 1, padding 0
  - Conv (32, 64), 1×1 filters, stride 1, padding 0
  - Conv (64, 64), 1×1 filters, stride 1, padding 0
  - FC (576 + 3, 256)
  - S5 (256, 256)
- Value MLP:
  - FC (256, 64)
  - FC (64, 64)
  - FC (64, 1)
- Policy MLP:
  - FC (256, 64)
  - FC (64, 64)
  - FC (64, 3)

F Hyperparameters

F.1 *Memory Sequence* and TSP

population size	5
learning rate	2.5×10^{-5}
batch size	4
rollout length	128
update epochs	4
minibatches	4
γ	0.99
λ_{GAE}	0.95
ϵ clip	0.2
entropy coefficient	0.01
value coefficient	0.5
max gradient norm	0.5
anneal learning rate	False

F.2 *Goal Sequence*

population size	4
learning rate	1×10^{-5}
batch size	128
rollout length	32
update epochs	8
minibatches	8
γ	0.99
λ_{GAE}	0.95
ϵ clip	0.2
entropy coefficient	0.01
value coefficient	0.5
max gradient norm	0.5
anneal learning rate	False

https://doi.org/10.52202/079017-1907

G Compute Resources

Memory Sequence and TSP experiments were run on a single NVIDIA RTX A40 GPU (40GB memory) in under 20 minutes. Training of in-context learners in *Goal Sequence* was run in under 8 minutes on 4 A40s (oracle training runs in the same time). In-weights accumulation in *Goal Sequence* was run in 30 minutes on 4 A40s.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction motivate the study of cultural accumulation in artificial learning agents and claim that, under our models, cultural accumulation can be achieved over RL training and in-context RL in the settings we consider, both of which are backed up by our results.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In our conclusion we outline the limitations of this work and highlight directions for future research.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

https://doi.org/10.52202/079017-1907

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Experimental details required to understand and interpret the results are included in the main text and all further details required to reproduce the results are included in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide an anonymous open-sourced repository containing all code used to run experiments and instructions needed to reproduce results.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Key training and test details needed to interpret results are in the main text any further details are included in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All plots include shaded regions indicating standard error or error bars representing 95% confidence intervals.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

https://doi.org/10.52202/079017-1907

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute resources used for each experiment are provided in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This research was conducted with strict adherence to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: In the paper's conclusion, we discuss the potential positive and negative societal impacts of this work.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not include the risk of any data or models that have risk for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: No existing assets were used in this paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

https://doi.org/10.52202/079017-1907

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The open-sourcing of our code includes the release of the environments used with supporting documentation.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing, nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing, nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-1907