Policy Search, Retrieval, and Composition via Task Similarity in Collaborative Agentic Systems

Saptarshi Nath[1], Christos Peridis[1], Eseoghene Benjamin[4], Xinran Liu[2], Soheil Kolouri[2], Peter Kinnell[1], Zexin Li[3], Cong Liu[3], Shirin Dora[1], Andrea Soltoggio[1]

[1]Department of Computer Science, Loughborough University, Loughborough, UK
[2]Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, USA
[3]Department of Electrical Engineering and Computer Engineering, University of California, Riverside, USA
[4]Alan Turing Institute, London, UK
{s.nath, c.peridis, p.kinnell, s.dora, a.soltoggio}@lboro.ac.uk, {xinran.liu, soheil.kolouri}@vanderbilt.edu, {zli536, congl}@ucr.edu, ebenjamin@turing.ac.uk

Abstract

Agentic AI aims to create systems that set their own goals, adapt proactively to change, and refine behavior through continuous experience. Recent advances suggest that, when facing multiple and unforeseen tasks, agents could benefit from sharing machine-learned knowledge and reusing policies that have already been fully or partially learned by other agents. However, how to query, select, and retrieve policies from a pool of agents, and how to integrate such policies remains a largely unexplored area. This study explores how an agent decides what knowledge to select, from whom, and when and how to integrate it in its own policy in order to accelerate its own learning. The proposed algorithm, *Modular Sharing and Composition in Collective Learning* (MOSAIC), improves learning in agentic collectives by combining (1) knowledge selection using performance signals and cosine similarity on Wasserstein task embeddings, (2) modular and transferable neural representations via masks, and (3) policy integration, composition and fine-tuning. MOSAIC outperforms isolated learners and global sharing approaches in both learning speed and overall performance, and in some cases solves tasks that isolated agents cannot. The results also demonstrate that selective, goal-driven reuse leads to less susceptibility to task interference. We also observe the emergence of self-organization, where agents solving simpler tasks accelerate the learning of harder ones through shared knowledge.

Code — https://github.com/DMIU-ShELL/MOSAIC

1 Introduction

The ability to solve complex, open-ended problems is an increasingly desirable objective as agentic AI principles become central in AI architectures. In open-ended real-world settings, these systems face challenges in generalizing and adapting to an ever-evolving stream of data and tasks (De Lange et al. 2022; Parisi et al. 2019). Although new lifelong learning algorithms (Kudithipudi et al. 2022) are being introduced to provide continual evolution and adaptation, when agents learn in isolation, they can only benefit from their limited experiences, unlike humans who benefit from collaboration and sharing of experiences and skills. Recent studies suggest that a shift towards more collaborative types of learning, where agents selectively share and reuse modular knowledge from other agents, is key to reducing redundant learning, accelerating adaptation, and improving robustness (Tarale, Rietman, and Siegelmann 2025; Soltoggio et al. 2024; DARPA/MTO 2021).

In the context of self-adaptation, reinforcement signals provide an effective tool to achieve continuous improvement in a wide range of domains (Sutton and Barto 2018; Jaderberg et al. 2017). In addition, task similarities have been exploited to benefit learning in areas such as transfer learning and multi-task RL (Hendawy, Peters, and D'Eramo 2024; Sun et al. 2022; D'Eramo et al. 2020). Lifelong learning approaches also leverage task similarity when previously learned tasks benefit the learning of subsequent tasks, with an advantage often named *forward transfer* (Ben-Iwhiwhu et al. 2023; Kirkpatrick et al. 2017). However, many approaches to sharing knowledge, e.g., distributed RL (Sartoretti et al. 2019) or Federated Learning (Yoon et al. 2021), often assume some form of centralization and uniformity of tasks, which is ill-suited to the requirements of agentic AI where each agent acts independently, asynchronously, and likely on a large variety of different tasks. Recent studies have begun to investigate the sharing and reuse of policies in evolving multi-agent contexts (Nath et al. 2023; Tarale, Rietman, and Siegelmann 2025; Gerstgrasser, Danino, and Keren 2023). These developments reflect a growing recognition that scalable AI systems increasingly depend on self-directed selective collaboration among autonomous learners.

Knowledge and skill sharing among a set of independent and autonomous learners, while conceptually appealing, presents significant challenges. In particular, open research questions reflect the problems of how to identify what knowledge to acquire, from whom to acquire it, how such knowledge is best represented, and how to integrate it to manifest robust lifelong learning properties (DARPA/MTO 2021; Soltoggio et al. 2024). Furthermore, it is still unclear to what extent collaborative learning offers an advantage over isolated or centralized learning.

To address these questions, this study introduces *Modular Sharing and Composition in Collective Learning* (MO-

SAIC), which describes how agents can select appropriate knowledge from peers with the aim of implementing policy composition and reuse. Problem analysis and the following ablation studies indicate that essential components of such a system are (1) knowledge selection through task similarity, (2) modular and transferable neural representations, and (3) policy integration, composition, and fine-tuning.

Specific algorithmic choices in MOSAIC to implement such components are Wasserstein embeddings for task similarity, transferable binary network masks for sharing, and learnable linear combinations of those to achieve integration and fine tuning. In principle, analogous methods could be used to adapt MOSAIC to a variety of domains, e.g., LoRA-based modular composition for foundation models and LLMs (Hu et al. 2022; He et al. 2022) and embedding methods or descriptors to express task alignment (Achille et al. 2019; Grover et al. 2018; Rakelly et al. 2019).

Simulations indicate that selectively acquiring relevant knowledge is essential to effectively reuse policies. The composition of policies is best achieved with a similarity-driven and reward-aware weighting. The results show how communicating MOSAIC agents learn faster than isolated learners by reusing knowledge from peers on similar tasks. In some cases, communicating agents can solve tasks that they could not learn alone. In addition, the analysis reveals an implicit self-organization, where agents discover and exploit curriculum structures in the available knowledge, resulting in agents building on skills hierarchically, from simpler tasks to learn more complex ones.

In summary, MOSAIC supports the hypothesis that task-relevant policy transfer and reuse, guided by agent-centered optimization objectives, is feasible and effective with advantages over learning in isolation or sharing global parameters, which could lead to task interference. To the best of our knowledge, this is the first study that combines modular and transferable task-specific knowledge with similarity-based selection and integration. Such a combination of components is sufficient to significantly improve sample efficiency and learning through collective knowledge sharing.

2 Related Work

Recent work in modular knowledge reuse for multi-agent RL explores various approaches to enable efficient knowledge transfer. Federated learning methods (Yoon et al. 2021) communicate task-specific masks or binary representations. These approaches lack principled task similarity metrics for selective reuse. MOSAIC addresses this gap through Wasserstein-based similarity measures that guide knowledge selection based on performance signals.

Several frameworks allow parameter-efficient knowledge sharing through masking (Ge et al. 2023; Nath et al. 2023; Gerstgrasser, Danino, and Keren 2023), prototype aggregation (Tan et al. 2022), or decentralized training (Douillard et al. 2024; Jaghouar and Hagemann 2024). Communication-efficient methods reduce information exchange through knowledge preservation and selective sharing (McMahan et al. 2017). These approaches rely on predefined sharing protocols or require synchronous updates. MOSAIC's binary mask representation enables asynchronous, selective reuse and maintains modularity while minimizing communication overhead.

Policy composition methods like PaCo (Sun et al. 2022) interpolate within shared subspaces. These methods lack mechanisms for similarity-driven integration. Peer-to-peer frameworks (Tarale, Rietman, and Siegelmann 2025; Yu, Vincent, and Schwager 2022) and model-based approaches (Jiang, Narayanan, and Li 2021) enable decentralized coordination. These frameworks do not combine selection with modular representations. MOSAIC combines Wasserstein-based task similarity with binary mask modularity and similarity-driven linear policy combinations. Further discussion of related works is provided in Appendix B.

3 Methodology

The following section describes the components of the MOSAIC algorithm. A high level illustration is provided in Figure 1. Pseudocode can be found in the Appendix F.

3.1 Policy Representations

Isolating task-specific knowledge in compact representations (Alet, Lozano-Perez, and Kaelbling 2018) is a key to making policy transfer easier and more lightweight across agents. Lifelong learning parameter isolation approaches can achieve this goal (Rusu et al. 2016; Mallya and Lazebnik 2018; Wortsman et al. 2020; Ben-Iwhiwhu et al. 2023).

This study adopts neural network masks (Ben-Iwhiwhu et al. 2023; Wortsman et al. 2020) that have shown strong results in lifelong reinforcement learning and supervised learning settings, and supports composition through linear combinations of mask modules. PPO is used for the experiments in this paper; however, the proposed modular composition mechanism is, in principle, agnostic to the choice of RL algorithm. Mask representations rely on a frozen backbone network Φ that is homogeneous across all agents. Each policy π_τ is implicitly parameterized through a sparse mask ϕ_τ, such that $\pi_\tau = \pi_{\Phi \odot g(\phi_\tau)}$, where $\odot$ denotes the Hadamard product, and $g(\cdot)$ is a binarization function applied during the forward pass. The mask ϕ_τ consists of a set of real-valued score vectors, one per layer of the shared backbone network Φ, which select a sparse task-specific subnetwork from Φ without modifying its parameters. Scores are binarized during the forward pass to select a discrete subnetwork from the backbone, given by the thresholding function,

$$g(\phi_\tau)_\ell = \begin{cases} 1 & \text{if } (\phi_\tau)_\ell > \epsilon \\ 0 & \text{otherwise} \end{cases}, \quad \text{with } \epsilon = 0 \quad , \qquad (1)$$

where $g(\phi_\tau)$ denotes the complete binary mask and ℓ indexes the layers of the backbone network Φ. During the backward pass, gradients are computed with respect to the real-valued mask parameters ϕ_τ, and updates are performed using the straight-through estimator (STE) (Wortsman et al. 2020) to enable learning through the non-differentiable binarization step.

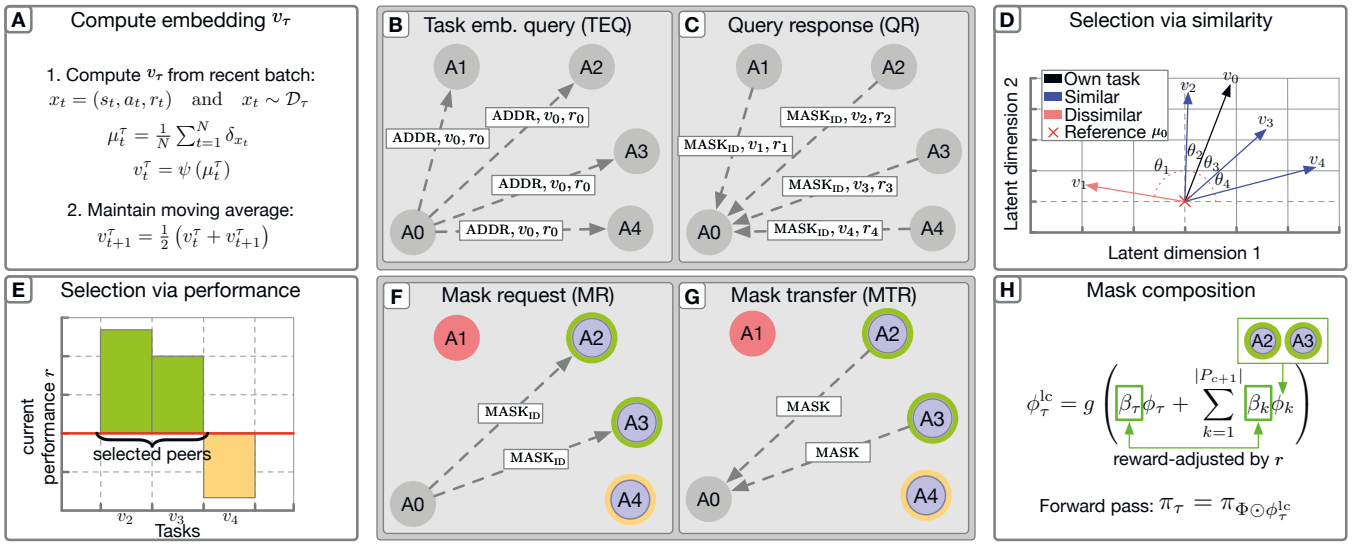

Figure 1: High-level illustration of the main MOSAIC algorithmic steps. (A) A Wasserstein task embedding v_τ is computed from a SAR batch by Agent A0 (representing any agent in the collective). (B) Periodically, the agent A0 broadcasts a task embedding query (TEQ) to all known peers. (C) Peers send back a query response (QR) that contains their task's v_τ, r, and the corresponding mask ID; (D) Agent A0 selects relevant embeddings using cosine similarity on the Wasserstein embeddings. (E) A0 further selects relevant embeddings using Criterion 1 and Criterion 2. (F) A0 sends mask requests (MR). (G) The contacted agents respond by sending the requested mask through a mask transfer (MTR). (H) Incoming masks from A2 and A3 are incorporated into A0. The training of the agent's policy occurs in parallel (not represented in the figure).

3.2 Wasserstein Task Embeddings for Online Reinforcement Learning

To identify suitable policies for a given task, MOSAIC extends Liu et al. (2025) to RL by computing embeddings over batches of state-action-reward (SAR) data collected online (see Figure 1(A)). For task τ, the agent maintains an empirical task distribution

$$\mu_\tau = \frac{1}{N}\sum_{t=1}^{N} \delta_{x_t} \in \mathcal{P}(\mathbb{R}^d), \tag{2}$$

where $x_t = (s_t, a_t, r_t)$ and $x_t \sim \mathcal{D}_\tau$. Here, $\mathcal{D}_\tau$ denotes the replay buffer for τ, and each $x_t \in \mathbb{R}^d$ is a state-action-reward (SAR) tuple. The tuple dimensionality is $d = d_s + d_a + d_r$, where d_s, d_a, and d_r are the dimensions of the state, action, and scalar reward, respectively. We use $N = 128$ samples in our experiments. δ_{x_t} denotes the Dirac measure centered on the sample x_t. A fixed synthetic reference distribution μ_0 is defined as

$$\mu_0 = \frac{1}{M}\sum_{m=1}^{M} \delta_{x_m^0}, \tag{3}$$

where $x_m^0 \sim \text{Uniform}(-1, 1)^d$ is sampled once during initialization and fixed thereafter. We fix $M = 50$ as the number of reference points in the synthetic distribution. This shared reference is used across all agents and enables them to align their embeddings in a shared latent space without centralized training or supervision. The 2-Wasserstein distance between μ_τ and μ_0 is computed by solving the optimal transport problem,

$$\gamma^* = \arg\min_{\gamma \in \mathbb{R}^{N\times M}} \sum_{t=1}^{N}\sum_{m=1}^{M} \gamma_{tm}\|x_t - x_m^0\|^2 \quad \text{s.t. } \gamma_{tm} \geq 0,\ \gamma \mathbf{1}_M = \tfrac{1}{N}\mathbf{1}_N,\ \gamma^\top \mathbf{1}_N = \tfrac{1}{M}\mathbf{1}_M, \tag{4}$$

where $\mathbf{1}_N$ and $\mathbf{1}_M$ are vectors of ones in $\mathbb{R}^N$ and $\mathbb{R}^M$, respectively. The Wasserstein task embedding operator ψ maps the task distribution μ_τ to a vector $v_\tau \in \mathbb{R}^{M\times d}$ via the barycenter projection of γ^* on $\{x_t\}_{t=1}^N$,

$$v_\tau = \psi(\mu_\tau) = \left[\sum_{t=1}^{N} \gamma_{tm}^* x_t\right]_{m=1}^{M}. \tag{5}$$

This mapping places each task τ in a shared latent space, where metrics can be used to quantify task relationships. A moving average is maintained to smooth fluctuations as the agent's policy evolves, r.

3.3 Knowledge Selection and Sharing

Agents communicate by maintaining a list of IP addresses and ports for other agents, which allows them to be located anywhere on the Internet. Each agent $i \in \mathcal{I}$ can communicate directly with all other agents $j \in \mathcal{I},\ j \neq i$ at any time.

Phase 1: Embedding queries At any time, an agent i can initiate a task embedding query (TEQ) by broadcasting its current task embedding v_i, iteration performance $\overline{r}_i$, and its IP/port (see Figure 1(B)(C)). Upon receiving the TEQ, each peer agent $j \neq i$ responds with its most recent task embedding v_j and associated performance $\overline{r}_j$.

Phase 2: Policy selection. Once the agent has received query responses (QR) (see Figure 1(F)(G)) with relevant embeddings from other agents, it computes the cosine similarity between its own embedding and those of each peer,

$$\cos(v_i, v_j) = \frac{\langle \text{vec}(v_i), \text{vec}(v_j) \rangle}{|\text{vec}(v_i)| \cdot |\text{vec}(v_j)|}, \quad \cos \in [-1, 1]. \quad (6)$$

To identify useful policies, agent i applies two heuristic filters based on similarity and performance:

$$\mathbb{I}_{\text{align}}(i,j) = \begin{cases} 1 & \text{if } \cos(v_i, v_j) > \theta, \\ 0 & \text{otherwise} \end{cases} \quad \text{(Criterion 1)}$$

$$\mathbb{I}_{\text{perf}}(i,j) = \begin{cases} 1 & \text{if } \overline{r}_j > \overline{r}_i, \\ 0 & \text{otherwise} \end{cases} \quad \text{(Criterion 2)}$$

with $\theta = 0.5$ in all experiments. The first criterion promotes semantic alignment between tasks, revealing latent similarities that can lead to effective transfer (see Section 4.4). The second ensures that agents only acquire masks from peers whose performance exceeds their own, avoiding noisy or under-trained policies that can degrade performance. Criterion 2 guards against the frequent false positives seen when comparing poorly trained policies that yield deceptively similar embeddings.

The two phases and the selection criteria also result in a bandwidth-efficient approach, since policies are transferred only when considered potentially useful. The peer masks that pass both criteria are stored as the set $P_{c+1} = \{\phi_1, ..., \phi_K\}$, where each ϕ_k denotes a mask received from a selected peer.

3.4 Knowledge Composition and Fine Tuning

Masks received from other agents meeting the two selection criteria above have been pretrained on tasks that are likely related to or share similarities with the agent's current task. As previously shown in Ben-Iwhiwhu et al. (2023), a linear combination of masks, weighted with trainable parameters β, can lead to beneficial policy search in RL. MOSAIC exploits this idea by combining policies that have been trained on the local agent, plus those acquired from other agents. In this study, the β values are computed using softmax on a set of beta parameters, optimized in log-space, $\tilde{\beta} \in \mathbb{R}^{1+|P_c|}$, where P_c is the number of masks acquired during a communication event c. During the forward pass, the agent constructs the linearly combined, binarized mask,

$$\phi_\tau^{\text{lc}} = g\left(\beta_\tau \phi_\tau + \sum_{k=1}^{|P_{c+1}|} \beta_k \phi_k\right), \quad (7)$$

where ϕ_τ is the agent's own task mask, and ϕ_k is the k-th peer mask. The parameters β_τ and β_k are the softmax-normalized values derived from $\tilde{\beta}$, for ϕ_τ and ϕ_k, respectively. $|P_{c+1}|$ is the number of masks acquired in the next communication event. The resulting binary mask, ϕ_τ^{lc}, is then used to modulate the backbone parameters, which gives the final policy π_τ. As training continues on the local task, the agent updates the β parameters and the real value ϕ_τ via backpropagation (keeping all ϕ_k masks fixed) thus fine tuning the overall resulting policy determined by ϕ_τ^{lc}.

Reward-Guided Initialization (RGI). The trainable parameters, β, allow gradient descent to determine which policies are most useful for the current task. The initial weighting as the masks are received is given by,

$$\beta_\tau = 0.5 + 0.5\overline{r}, \quad \beta_k = \frac{0.5(1-\overline{r})}{|P_{c+1}|} \quad , \quad (8)$$

where $\overline{r} \in [0, 1]$ is the agent's normalized return from the last iteration. This scheme biases low-performing agents toward external knowledge and high-performing agents toward their own policies. Ablation studies show that such an initialization provides a strong advantage, even though β is later tuned by gradient descent.

Before integrating a new set of peer masks P_{c+1}, the agent consolidates its current task mask ϕ_τ with the previous peer masks P_c using a weighted linear combination,

$$\phi_\tau \leftarrow \beta_\tau \phi_\tau + \sum_{k=1}^{|P_c|} \beta_k \phi_k \quad . \quad (9)$$

The consolidation collapses multiple masks into one without affecting the policy, managing memory scalability.

4 Experiments

MOSAIC is evaluated on three sparse-reward reinforcement learning benchmarks. The CT-graph (Soltoggio et al. 2023) and the MiniHack MultiRoom (Samvelyan et al. 2021) are used to assess the advantage of communicating agents versus isolated agents. The MiniGrid Crossing (Chevalier-Boisvert, Willems, and Pal 2018) is used to assess the performance of MOSAIC against the chosen baselines. Each benchmark includes tasks with similar, dissimilar, and interfering tasks to assess whether MOSAIC can identify and leverage the available task similarities, while avoiding interfering policies. Each agent is assigned a unique task and trained in parallel with other agents. PPO (Schulman et al. 2017) was used in all experiments, with an FCN for MiniGrid and CT-graph, and a CNN for MiniHack. Individual results are shown in Appendix E, Figures 14, 15, and 16. Details on computing infrastructure, hyperparameters, architectures and libraries are reported in the Appendix F, Tables 5, 6, 7, and 8. Appendix A, Table 3 reports significance testing.

4.1 Image Sequence Learning

The image sequence learning (ISL) benchmark, implemented with the Configurable Tree Graph (CT-Graph) environment (Soltoggio et al. 2023), consists of procedurally generated tree navigation problems where each node is an RL state encoded as an image. We define a curriculum of 28 tasks organized into four independent image sets, each containing seven related tasks of increasing tree depth (2–8). This configuration produces four distinct and unrelated task groups with internal hierarchies of difficulty, from easy (depth 2) to very hard (depth 8). Sparse rewards and exponential branching make the benchmark challenging, with reward probabilities as low as $\approx 7.74 \times 10^{-9}$ for depth-8 tasks (Soltoggio et al. 2023).

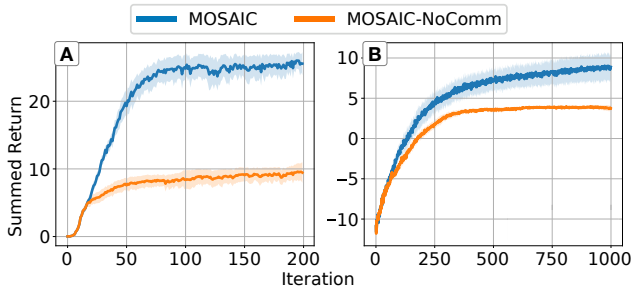

Figure 2: Performance of communicating MOSAIC agents versus isolated agents on the same tasks. (A) Image sequence learning, 28 tasks, five seeds/task: average of 140 runs with 95% confidence intervals (Colas, Sigaud, and Oudeyer 2018). (B) MiniHack Multiroom, 14 tasks, five seeds/task: average of 70 runs with 95% confidence intervals. MOSAIC agents achieve relative gains of 170.8% and 128.2% over the isolated baseline in the image sequence and MiniHack benchmarks, respectively.

Results in Figure 2(A) show that MOSAIC achieves significantly faster learning and broader task coverage, outperforming MOSAIC-NoComm (no sharing among agents) by 2.7×, reaching a maximum total return of 26.0 across all 28 tasks. MOSAIC reaches 50% performance (total return of 14) in 37 iterations (18,944 steps). MOSAIC-NoComm plateaus at a maximum of 9.6. On average, MOSAIC-NoComm fails on 18 tasks, whereas MOSAIC fails on only 2.

4.2 MiniHack MultiRoom

MiniHack is a grid-based navigation task with sparse rewards and pixel observations. Agents must traverse connected rooms to reach a final goal. We evaluate MOSAIC on 14 tasks grouped into two difficulty clusters: room sizes of 4×4 and 6×6. Each level adds a room connected by a closed but unlocked door. Agents receive +1 for task completion and -0.01 for collisions. Task layouts are randomized every episode to enable learning of behavioral strategies as opposed to trajectories. As shown in Figure 2(B), MOSAIC agents reach zero reward by iteration 132 (270,336 steps) versus iteration 176 (360,448 steps) for MOSAIC-NoComm, which is a 25% reduction in the required samples. Over the full run, MOSAIC attains a maximum total return of 9.04 across all tasks, compared to 3.96 for MOSAIC-NoComm.

4.3 MiniGrid Crossing

The MiniGrid benchmark is a sparse-reward grid-world navigation task with symbolic observations. MOSAIC is evaluated on 14 tasks spanning seven SimpleCrossing and seven LavaCrossing variants, which differ in layout, object placement, and room structure. Total return curves for all methods are shown in Figure 3, with final and intermediate metrics in Table 1. MOSAIC is compared against the following baselines: Multi-Task PPO (MTPPO) uses a shared backbone without modularity or interference mitigation. Multi-DQN (MDQN) (D'Eramo et al. 2020) attaches task-specific Q-heads to a shared encoder but has no mechanism to avoid interference. PCGrad+MoE (Yu et al. 2020) combines a shared encoder, expert subnetworks, and gradient projection to resolve conflicts. Mixture of Orthogonal Experts (MOORE) (Hendawy, Peters, and D'Eramo 2024) promotes expert diversity via orthogonalization with learned gating to reduce overlap. MOSAIC achieves the highest final return (11.67), outperforming all baselines. MTPPO peaks at 4.64 (30% of the theoretical maximum), while MDQN fails to learn any tasks, peaking at 0.31. MOORE and PCGrad+MoE improve faster early, reaching 50% thresholds in 24 (49,152 steps) and 54 iterations (110,592 steps) respectively, but plateau well below MOSAIC and never exceed 75%. We speculate that centralized models with gradient sharing (MTPPO, PCGrad+MoE) achieve early gains through shared features but suffer interference that limits long-term specialization.

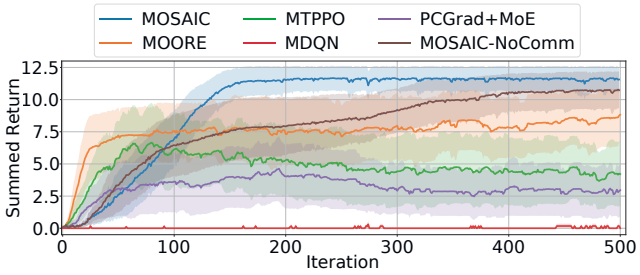

Figure 3: Comparison of MOSAIC to baseline approaches on the 14 task MiniGrid curricula made up of SimpleCrossing and LavaCrossing task variations: average performance for 70 runs with 95% confidence intervals.

Method	Final perf	25% perf	50% perf	75% perf
MOSAIC (ours)	**11.67**	55	90	119
MTPPO	4.64	48	-	-
MDQN	0.31	-	-	-
PCGrad+MoE	6.66	24	54	-
MOORE	8.83	**13**	**24**	-
MOSAIC-NoComm	10.78	46	87	291

Table 1: Performance metrics on the MiniGrid Crossing benchmark. The table reports the final average return and the number of iterations to reach 25%, 50% and 75% of the theoretical maximum. Dashes indicate that the thresholds were not reached during the training.

4.4 How Similarity Helps Targeted Policy Transfers

Figure 4 illustrates the average cosine similarity in the CT-graph benchmark between task embeddings and the converged consolidation parameters $\hat{\beta}_j^i$ (see Eq. (9)), which provides an indication of the influence of policies derived from task j when learning task i.

The cosine similarities of tasks presented in random order (Figure 4(A)), when clustered with WPGMA (Figure 4(C)),

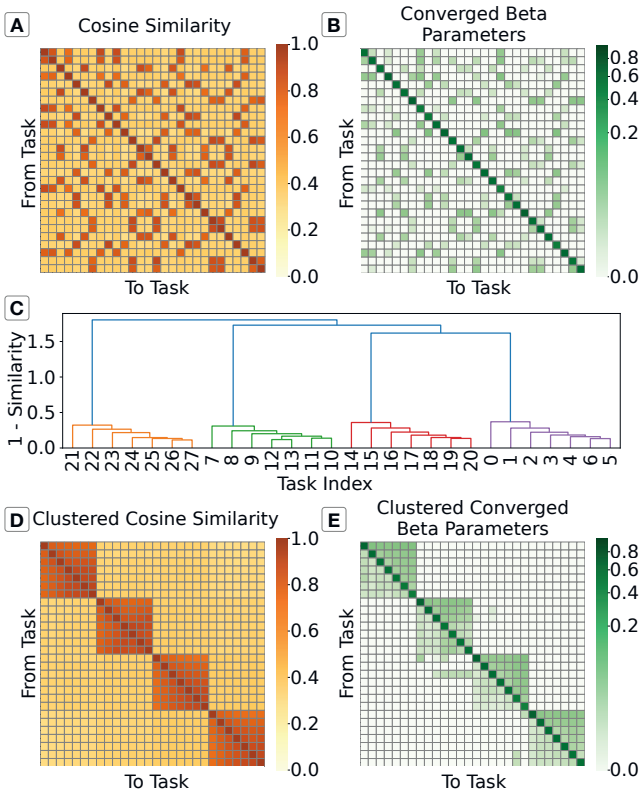

Figure 4: Pairwise cosine similarity and $\hat{\beta}$ statistics in the image sequence learning benchmark. (A) Cosine similarity matrix. (B) $\hat{\beta}$ values indicating policy use per task. (C) Cosine similarities clustered using WPGMA, shown as a dendrogram. (D) Cosine similarity matrix reordered by clustering. (E) Clustered $\hat{\beta}$ values show that similar tasks exhibit the highest policy reuse. Annotated heatmaps of individual clusters are in Appendix E.1, Figure 18.

allow for the reconstruction of the four groups of tasks, and even the inter-group difficulty relationship. By reordering the matrix using a such clustering, Figure 4(D) illustrates visually the successful reconstruction of task relationships. The $\hat{\beta}$ matrices (Figures 4(B) and (E)) confirm that policy reuse and task similarity remain similar even after fine tuning of the policies. Note that, unlike the symmetric similarity matrix, $\hat{\beta}$ values are asymmetric, reflecting source-to-target policy reuse.

Figure 5 presents the performance of agents in the CT-graph grouped by difficulty levels. Interestingly, comparing with isolated learning (B) suggests that the policies of easier tasks are progressively shared with agents solving harder tasks. This coordination enabling those agents to find successful policies that the same RL algorithm fails to discover when learning in isolation. This analysis helps explain the 170.8% performance gain of communicating agents over isolated agents (Figure 2(A)): agents tackling harder tasks retrieve, combine, and refine policies from simpler tasks to solve complex problems.

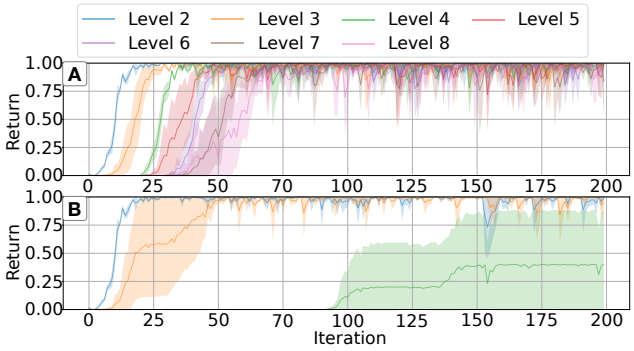

Figure 5: Performance grouped by task complexity, i.e., seven different levels, corresponding to different graph depths, in the image sequence learning problem. Average performance for 20 runs (four tasks and five seeds/task) with 95% confidence intervals. (A) Communicating MOSAIC agents are compared with (B) isolated MOSAIC agents . Communicating agents solve tasks progressively, from the simplest to the hardest tasks. Isolated agents only manage to solve the two simplest tasks, and partially the third, but fail on the four most complex tasks.

4.5 Ablation Studies

Ablation studies were conducted to assess the effect on performance of MOSAIC's knowledge selection criteria and reward-guided initialization (RGI) on performance. Figure 6 compares three ablated variants of MOSAIC: without cosine similarity-based selection (¬ Criterion 1), without performance-based selection (¬ Criterion 2), and without reward-guided weight initialization (¬ RGI). MOSAIC-NoComm is included as a reference baseline.

¬ Criterion 1 reaches a total return of 20.1, whereas MOSAIC reaches 26.0, outperforming it by a factor of 1.29. We speculate that this gap could widen further in settings in which the number of unrelated tasks grows, i.e., where selecting the most relevant knowledge becomes critical.

¬ Criterion 2 performs comparable to MOSAIC at the end of training (25.6), but exhibits significantly slower learning. This behavior makes sense because as agents improve their performance, prioritizing higher reward becomes less critical. These two ablations ¬ Criterion 1 and ¬ Criterion 2 reach 50% of the theoretical maximum (14.0) in 76 (38,912 steps) and 71 iterations (36,352 steps), respectively. MOSAIC reaches this threshold in 37 iterations (18,944 steps), reaching 50% performance 1.9× and 2.0× faster, respectively.

¬ RGI removes reward-guided initialization and instead uses a fixed weighting of 0.5 for the local policy and 0.5 for external policies. This experiment performs comparably with MOSAIC; however, the average return shows periodic instability, characterized by sharp performance drops coinciding with communication events, resembling a sawtooth pattern. This behavior suggests that RGI is crucial to learning stability when integrating external masks.

Further ablation studies are shown in Appendix D to test the impact of query frequency on the performance of MO-

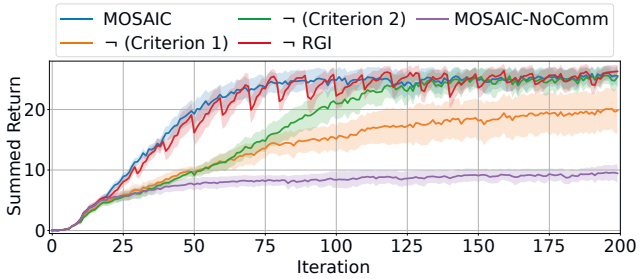

Figure 6: Ablation studies showing average performance over 140 runs (28 tasks, five seeds per task), with 95% confidence intervals. Variants tested include MOSAIC with no similarity criterion (¬ Criterion 1), no reward criterion (¬ Criterion 2), or no reward-guided initialization (¬ RGI). MOSAIC-NoComm is included as a baseline. Removing any of these components significantly degrades performance, highlighting their combined importance.

SAIC, and the impacts of the number of samples and size of the reference distribution on embedding accuracy and performance.

5 Discussion

The results show that MOSAIC's principles are effective across diverse settings. Agents achieved faster and more successful learning dynamics than isolated agents. In particular, transferring masks from some tasks accelerated learning in other tasks, consistent with findings in centralized lifelong learning (Ben-Iwhiwhu et al. 2023). These gains were most evident in reward-sparse and hierarchical environments, where agents that learned simpler tasks supported those that learned harder ones (Florensa et al. 2018; Nachum et al. 2018).

The ablation study highlights the importance of selection based on relevance and utility and of reward-adjusted weighting when integrating new policies. Naïve sharing overwrites stable learning (Isele and Cosgun 2018), deteriorating performance. These results emphasize the importance of selection when the available tasks and curriculum are unknown (Foerster et al. 2016). The effect of reward-guided initialization suggests that careful policy weighting is important to avoid performance disruption before fine tuning.

It is worth noting that MOSAIC's design takes inspiration from lifelong learning algorithms. Thus, policy composition can easily be extended to previously learned policies of previous tasks. Such an extension could deal with continuous streams of evolving tasks (Serra et al. 2018), supporting scalable and open-ended lifelong learning.

Limitations. Although the experiments were conducted in simulation, MOSAIC is designed for decentralized, bandwidth-limited deployments (e.g., robot fleets, on-device perception). Agents exchange only compact embeddings and mask scores while inference stays on-device. Key deployment considerations are cross-site normalization, private/authenticated exchange, and tuning query frequency and size to fit bandwidth.

MOSAIC relies on raw per-iteration reward as a selection and weighting signal, which limits generalization to environments with differing reward functions. Reward-normalized scores or task-progress metrics could improve environment-agnostic reuse. Adaptive or sparse topologies would better mitigate bandwidth scalability constraints (Tang et al. 2024). The policy composition used in MOSAIC is limited to positive combinations only. Other composition or mixture approaches could be tested (He et al. 2022). Alternative modular skill representations could be used for applications to larger neural networks, e.g., LoRA-like modules and adapters for transformer architectures (Hu et al. 2022; He et al. 2022). In all such cases, the requirement for a shared backbone or foundation model can reduce the pool of agents able to share knowledge and limit long-term development of progressively more complex skills (Houlsby et al. 2019).

Communication and selection mechanisms in MOSAIC were introduced as a proof-of-concept to test whether selective transfer is essential for agentic knowledge reuse (DARPA/MTO 2021; Soltoggio et al. 2024). Allowing policies to learn when and how to communicate could further increase autonomy (Foerster et al. 2016), but dynamic communication policies raise safety concerns. Agents could learn uncooperative strategies, exploit collective knowledge without contributing, or share harmful policies. Such risks mirror classic game-theoretic challenges and require safeguards. MOSAIC-like systems may also face adversarial attacks that exploit the agent-to-agent nature through model poisoning or policy extraction.

6 Conclusion

This work introduced Modular Sharing and Composition in Collective Learning (MOSAIC), where agents search, retrieve, and compose policies obtained from peers learning other tasks. The collective was shown to significantly outperform isolated agents by selecting and fine-tuning task-aligned knowledge. Results show that policy sharing is most effective when implemented selectively, with policy weighting and fine-tuning. Selection also results in the discovery of implicit curricula where simpler tasks help agents learn complex ones faster.

MOSAIC improves interpretability, as composed policies are traceable to their sources and the approach could extend to domains such as language, perception, and control. However, autonomous knowledge sharing carries risks: flawed or misaligned policies may spread rapidly. Future work must ensure that transfers remain safe and aligned with agent objectives. Using selective and modular reuse, MOSAIC can support scalable, adaptive, and collaborative agentic AI in real-world applications.

Acknowledgements

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under contract No. HR001121901 (Shared Experience Lifelong Learning) and the Industrial Robots-as-a-Service (IRaaS) project funded by the EPSRC (EP/V050966/1).

References

Achille, A.; Lam, M.; Tewari, R.; Ravichandran, A.; Maji, S.; Fowlkes, C.; Soatto, S.; and Perona, P. 2019. Task2Vec: Task Embedding for Meta-Learning. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 6429–6438.

Alet, F.; Lozano-Perez, T.; and Kaelbling, L. P. 2018. Modular Meta-Learning. In Billard, A.; Dragan, A.; Peters, J.; and Morimoto, J., eds., *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, 856–868. PMLR.

Ben-Iwhiwhu, E.; Nath, S.; Pilly, P. K.; Kolouri, S.; and Soltoggio, A. 2023. Lifelong Reinforcement Learning with Modulating Masks. *Transactions on Machine Learning Research*.

Chevalier-Boisvert, M.; Willems, L.; and Pal, S. 2018. Minimalistic Gridworld Environment for OpenAI Gym. https://github.com/maximecb/gym-minigrid.

Colas, C.; Sigaud, O.; and Oudeyer, P.-Y. 2018. How Many Random Seeds? Statistical Power Analysis in Deep Reinforcement Learning Experiments. *arXiv preprint arXiv:1806.08295*.

DARPA/MTO. 2021. Artificial Intelligence Exploration (AIE) Opportunity DARPA-PA-20-02-11 Shared-Experience Lifelong Learning (ShELL). https://sam.gov/opp/1afbf600f2e04b26941fad352c08d1f1/view. Accessed 2023/10/10.

De Lange, M.; Aljundi, R.; Masana, M.; Parisot, S.; Jia, X.; Leonardis, A.; Slabaugh, G.; and Tuytelaars, T. 2022. A Continual Learning Survey: Defying Forgetting in Classification Tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7): 3366–3385.

D'Eramo, C.; Tateo, D.; Bonarini, A.; Restelli, M.; and Peters, J. 2020. Sharing Knowledge in Multi-Task Deep Reinforcement Learning. In *International Conference on Learning Representations*.

Douillard, A.; Feng, Q.; Rusu, A. A.; Kuncoro, A.; Donchev, Y.; Chhaparia, R.; Gog, I.; Ranzato, M.; Shen, J.; and Szlam, A. 2024. DiPaCo: Distributed Path Composition. arXiv:2403.10616.

Florensa, C.; Held, D.; Geng, X.; and Abbeel, P. 2018. Automatic Goal Generation for Reinforcement Learning Agents. In Dy, J.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 1515–1528. PMLR.

Foerster, J. N.; Assael, Y. M.; de Freitas, N.; and Whiteson, S. 2016. Learning to Communicate with Deep Multi-Agent Reinforcement Learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, 2145–2153. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781510838819.

Ge, Y.; Li, Y.; Wu, D.; Xu, A.; Jones, A. M.; Rios, A. S.; Fostiropoulos, I.; Huang, P.-H.; Murdock, Z. W.; Sahin, G.; et al. 2023. Lightweight Learner for Shared Knowledge Lifelong Learning. *Transactions on Machine Learning Research*.

Gerstgrasser, M.; Danino, T.; and Keren, S. 2023. Selectively Sharing Experiences Improves Multi-Agent Reinforcement Learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Grover, A.; Al-Shedivat, M.; Gupta, J.; Burda, Y.; and Edwards, H. 2018. Learning Policy Representations in Multiagent Systems. In Dy, J.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 1802–1811. PMLR.

He, J.; Zhou, C.; Ma, X.; Berg-Kirkpatrick, T.; and Neubig, G. 2022. Towards a Unified View of Parameter-Efficient Transfer Learning. In *International Conference on Learning Representations*.

Hendawy, A.; Peters, J.; and D'Eramo, C. 2024. Multi-Task Reinforcement Learning with Mixture of Orthogonal Experts. In *The Twelfth International Conference on Learning Representations*.

Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; and Gelly, S. 2019. Parameter-Efficient Transfer Learning for NLP. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 2790–2799. PMLR.

Hu, E. J.; yelong shen; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Isele, D.; and Cosgun, A. 2018. Selective Experience Replay for Lifelong Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Jaderberg, M.; Mnih, V.; Czarnecki, W. M.; Schaul, T.; Leibo, J. Z.; Silver, D.; and Kavukcuoglu, K. 2017. Reinforcement Learning with Unsupervised Auxiliary Tasks. In *International Conference on Learning Representations*.

Jaghouar, S.; and Hagemann, J. 2024. OpenDiLoCo: An Open-Source Framework for Globally Distributed Low-Communication Training. In *Workshop on Efficient Systems for Foundation Models II @ ICML2024*.

Jiang, W. C.; Narayanan, V.; and Li, J. S. 2021. Model Learning and Knowledge Sharing for Cooperative Multiagent Systems in Stochastic Environment. *IEEE Transactions on Cybernetics*, 51(12): 5717–5727.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13): 3521–3526.

Kudithipudi, D.; Aguilar-Simon, M.; Babb, J.; Bazhenov, M.; Blackiston, D.; Bongard, J.; Brna, A. P.; Chakravarthi Raja, S.; Cheney, N.; Clune, J.; Daram, A.; Fusi, S.; Helfer, P.; Kay, L.; Ketz, N.; Kira, Z.; Kolouri, S.; Krichmar, J. L.; Kriegman, S.; Levin, M.; Madireddy, S.; Manicka, S.; Marjaninejad, A.; McNaughton, B.; Miikkulainen, R.; Navratilova, Z.; Pandit, T.; Parker, A.; Pilly, P. K.; Risi, S.; Sejnowski, T. J.; Soltoggio, A.; Soures, N.;

Tolias, A. S.; Urbina-Meléndez, D.; Valero-Cuevas, F. J.; Van de Ven, G. M.; Vogelstein, J. T.; Wang, F.; Weiss, R.; Yanguas-Gil, A.; Zou, X.; and Siegelmann, H. 2022. Biological underpinnings for lifelong learning machines. *Nature Machine Intelligence*, 4(3): 196–210.

Liu, X.; Bai, Y.; Lu, Y.; Soltoggio, A.; and Kolouri, S. 2025. Wasserstein Task Embedding for Measuring Task Similarities. *Neural Networks*, 181: 106796.

Mallya, A.; and Lazebnik, S. 2018. Packnet: Adding Multiple Tasks to a Single Network by Iterative Pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 7765–7773.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and Arcas, B. A. y. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Singh, A.; and Zhu, J., eds., *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, 1273–1282. PMLR.

Nachum, O.; Gu, S. S.; Lee, H.; and Levine, S. 2018. Data-Efficient Hierarchical Reinforcement Learning. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Nath, S.; Peridis, C.; Ben-Iwhiwhu, E.; Liu, X.; Dora, S.; Liu, C.; Kolouri, S.; and Soltoggio, A. 2023. Sharing Lifelong Reinforcement Learning Knowledge via Modulating Masks. In *Second Conference on Lifelong Learning Agents (CoLLAs) 2023*.

Parisi, G. I.; Kemker, R.; Part, J. L.; Kanan, C.; and Wermter, S. 2019. Continual Lifelong Learning with Neural Networks: A Review. *Neural Networks*, 113: 54–71.

Rakelly, K.; Zhou, A.; Finn, C.; Levine, S.; and Quillen, D. 2019. Efficient Off-Policy Meta-Reinforcement Learning via Probabilistic Context Variables. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 5331–5340. PMLR.

Rusu, A. A.; Rabinowitz, N. C.; Desjardins, G.; Soyer, H.; Kirkpatrick, J.; Kavukcuoglu, K.; Pascanu, R.; and Hadsell, R. 2016. Progressive Neural Networks. *arXiv preprint arXiv:1606.04671*.

Samvelyan, M.; Kirk, R.; Kurin, V.; Parker-Holder, J.; Jiang, M.; Hambro, E.; Petroni, F.; Kuttler, H.; Grefenstette, E.; and Rocktäschel, T. 2021. MiniHack the Planet: A Sandbox for Open-Ended Reinforcement Learning Research. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Sartoretti, G.; Wu, Y.; Paivine, W.; Kumar, T. S.; Koenig, S.; and Choset, H. 2019. Distributed reinforcement learning for multi-robot decentralized collective construction. In *Distributed Autonomous Robotic Systems: The 14th International Symposium*, 35–49. Springer.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*.

Serra, J.; Suris, D.; Miron, M.; and Karatzoglou, A. 2018. Overcoming Catastrophic Forgetting with Hard Attention to the Task. In *International Conference on Machine Learning*, 4548–4557. PMLR.

Soltoggio, A.; Ben-Iwhiwhu, E.; Braverman, V.; Eaton, E.; Epstein, B.; Ge, Y.; Halperin, L.; How, J.; Itti, L.; Jacobs, M. A.; Kantharaju, P.; Le, L.; Lee, S.; Liu, X.; Monteiro, S. T.; Musliner, D.; Nath, S.; Panda, P.; Peridis, C.; Pirsiavash, H.; Parekh, V.; Roy, K.; Shperberg, S.; Siegelmann, H. T.; Stone, P.; Vedder, K.; Wu, J.; Yang, L.; Zheng, G.; and Kolouri, S. 2024. A Collective AI via Lifelong Learning and Sharing at the Edge. *Nature Machine Intelligence*, 6(3): 251–264.

Soltoggio, A.; Ben-Iwhiwhu, E.; Peridis, C.; Ladosz, P.; Dick, J.; Pilly, P. K.; and Kolouri, S. 2023. The configurable tree graph (CT-graph): measurable problems in partially observable and distal reward environments for lifelong reinforcement learning. arXiv:2302.10887.

Sun, L.; Zhang, H.; Xu, W.; and Tomizuka, M. 2022. PaCo: Parameter-Compositional Multi-task Reinforcement Learning. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems*.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book. ISBN 0262039249.

Tan, Y.; Long, G.; Liu, L.; Zhou, T.; Lu, Q.; Jiang, J.; and Zhang, C. 2022. FedProto: Federated Prototype Learning across Heterogeneous Clients. In *AAAI Conference on Artificial Intelligence*.

Tang, S.; Ye, R.; Xu, C.; Dong, X.; Chen, S.; and Wang, Y. 2024. Decentralized and Lifelong-Adaptive Multi-Agent Collaborative Learning. *CoRR*, abs/2403.06535.

Tarale, P.; Rietman, E.; and Siegelmann, H. T. 2025. Distributed Multi-Agent Lifelong Learning. *Transactions on Machine Learning Research*.

Wortsman, M.; Ramanujan, V.; Liu, R.; Kembhavi, A.; Rastegari, M.; Yosinski, J.; and Farhadi, A. 2020. Supermasks in Superposition. *Advances in Neural Information Processing Systems*, 33: 15173–15184.

Yoon, J.; Jeong, W.; Lee, G.; Yang, E.; and Hwang, S. J. 2021. Federated Continual Learning with Weighted Inter-client Transfer. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 12073–12086. PMLR.

Yu, J.; Vincent, J. A.; and Schwager, M. 2022. DiNNO: Distributed Neural Network Optimization for Multi-Robot Collaborative Learning. *IEEE Robotics and Automation Letters*, 7(2): 1896–1903.

Yu, T.; Kumar, S.; Gupta, A.; Levine, S.; Hausman, K.; and Finn, C. 2020. Gradient Surgery for Multi-Task Learning. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 5824–5836. Curran Associates, Inc.