nature machine intelligence

Perspective https://doi.org/10.1038/s42256-024-00800-2

A collective AI via lifelong learning and sharing at the edge

In the format provided by the authors and unedited

Supplementary Information

This Supplementary Information describes ShELL algorithms and their implementations (Section 1). Following, additional technical details are provided on application scenarios and performance metrics (Section 2).

Contents

1 Examples of ShELL algorithms and implementations

In this section, we provide summaries of a few exemplary ShELL algorithms and implementations that achieve the objectives O1 to O3 via the abilities A1 to A4 as listed in Section 3. In some cases, they are tested on edge devices and include approaches and considerations for budgeted computation and deployment on SWaP-constrained devices. While their TRL levels may still be low, they demonstrate that ShELL algorithms are technically feasible and provide support to the intuitions and vision that are detailed in this Perspective paper. When available, links to the source code of the implementations are provided.

1.1 Multi-agent distributed lifelong learning for collective knowledge acquisition [1].

In this study, the authors introduce a multi-agent collaborative lifelong learning system. Each agent builds a local knowledge base and shares it with neighboring agents. Agents update their knowledge bases to align with their neighbors. They create task-specific sparse combinations of their local knowledge base to solve new tasks, update their knowledge as needed, and share it. With a connected communication graph, this framework converges to uniform agents with general expertise across all tasks. Benchmarks included land mine detection, facial expression recognition, and student score prediction.

1.2 Collaborative learning through shared collective knowledge and local expertise [2].

The previous work [1] was extended in [2], allowing agents to possess local expertise, resulting in non-uniform agents with a shared global knowledge base and individual local expertise. The approach decomposes the task-specific parameters into an overcomplete dictionary $D_i \in R^{\mathrm{d} \times \mathrm{u}}$ and the corresponding sparse vector representation $s_i^{(t)} \in R^{\mathrm{u}}$. The algorithm employs a broadcast knowledge approach to sharing within the network of agents, where D_i is shared once task learning is complete. The algorithm was tested on facial expression recognition with 21 total tasks with a 3-agent population where each agent learns 7 tasks, and landmine detection with 29 total tasks and a population of 2 agents collaborating per task.

1.3 Sharing lifelong reinforcement learning knowledge via modulating masks [3].

This study proposes a lifelong RL distributed and decentralized collective of agents in which agents query each other for modulating masks and transfer those that are relevant to the current task on a peer-to-peer basis. The core LL engine exploits modulating masks, based on a type of parameter isolation LL method, that are used to learn one task for each mask [9]. This partial model parameter sharing has limited LL overhead due to the parameter isolation approach. The masks can be linearly combined to learn new tasks, or transferred across the collective. The collective uses an on-demand mechanism as each agent queries the collective for masks when learning one specific task. The study also suggests a competitive mechanism in which the better-performing masks for one task are used to overwrite less-performing masks. Experiments were conducted on simulated RL problems that included the Minigrid environment and the CT-graph environment. The system was tested on a range of servers and embedded devices (Nvidia Jetson series including Nano, TX2, AGX, and Orin). Experiments ran with up to 32 agents, and 32 tasks, although such limits were determined by available devices rather than algorithmic constraints. Agents ran asynchronously with agents connected over the Internet and running in locations across different countries and continents. Implementation code available at: `https://github.com/DMIU-ShELL/deeprl-shell.git`

- *Lifelong learning algorithm*: **Lifelong reinforcement learning with modulating**

masks [9]. This study proposes and tests learnable masks and learnable linear combinations of masks for knowledge reuse in deep lifelong reinforcement learning. The LL overhead is limited to a memory requirement with the increase of stored masks. However, masks can be made arbitrarily sparse and can be binarized to reduce memory use. Implementation code available at: `https://github.com/dlpbc/mask-lrl`.

1.4 CoDeC: Communication-Efficient Decentralized Continual Learning [4].

The CoDeC method extends the Gradient Projection Memory approach [10] to a decentralized continual learning (DCL) setup, where a collection of networks (or agents) learns from training data that is distributed across space as well as time. In this setup, each agent computes gradients on its local data and shares that with its neighbors. Sharing across agents happens every training iteration and can be performed in two ways. The first methodology is a static Gossip mechanism, where an agent communicates with a fixed user-defined number of random agents in the system. The second methodology is an active communication model where an agent is communicating with others if their GPM distance is greater than a user-defined threshold. In this framework, each agent maintains a GPM and follows a two-step model update process. First, it updates the local gradient with gossip (following either the static or active model) averaging [11] using the neighbors' gradients, which ensures effective decentralized learning. Next, to minimize forgetting, it projects the averaged gradients on its GPM and updates the model. CoDeC reported results on three image classification benchmarks where up to 16 agents were trained and tested using the directed ring and torus topology. In the DCL setup, CoDeC obtained SoTA accuracy with minimum forgetting achieving up to 4.8x reduction in communication cost.

- *Lifelong learning algorithm*: **Gradient projection memory for continual learning [10]**. Gradient Projection Memory (GPM) [10] casts catastrophic forgetting problems in continual learning as gradient interference problems among sequential tasks. GPM partitions each layer's gradient space into two orthogonal subspaces: Core Gradient Space (CGS) and Residual Gradient Space (RGS). Important gradient directions (CGS) for previous tasks are stored in GPM, and gradient updates for the new tasks are taken along RGS to minimize interference. GPM operates in a single network (or agent) setup, where each task uses a shared network backbone.

1.5 Lightweight learner for shared knowledge lifelong learning [5].

Shared Knowledge Lifelong Learning (SKILL) deploys a decentralized asynchronous population of Lightweight Lifelong Learning (LLL) agents that each sequentially learn different tasks, with all agents operating independently and in parallel. After learning their respective tasks, agents share and consolidate their knowledge over a decentralized communication network, so that, in the end, all agents can master all tasks.

The goal of Lightweight Lifelong Learning (LLL) agents is to facilitate efficient sharing by minimizing the fraction of the agent that is specialized for any given task. Each

LLL agent thus consists of a common task-agnostic immutable part, where most parameters are, and individual task-specific modules that contain fewer parameters but are adapted to each task. The population of agents uses a broadcasting mechanism, where each agent shares its knowledge with all others each time a new task has been learned. Agents share their task-specific modules (Beneficial Bias (BB)), plus summary information ("task anchors") representing their tasks in the common task-agnostic latent space of all agents. Receiving agents register each received task-specific module using the corresponding anchor. Thus, every agent improves its ability to solve new tasks each time new task-specific modules and anchors are received. If all agents can communicate with all others, eventually all agents become identical and can solve all tasks. On a new, very challenging SKILL-102 data set with 102 image classification tasks (5,033 classes in total, 2,041,225 training, 243,464 validation, and 243,464 test images), this new approach achieves much higher (and SOTA) accuracy over 8 lifelong learning baselines, while also achieving near-perfect parallelization. LLL is edge-friendly because the number of parameters learned for each task is small (typically $<$ 2 MBytes). Implementation code available at: `https://github.com/gyhandy/Shared-Knowledge-Lifelong-Learning.git`

- *Lifelong learning algorithm*: **Beneficial perturbation network for designing general adaptive artificial intelligence systems [12]** - Beneficial perturbation network (BPN) is a biologically plausible type of deep neural network with extra, out-of-network, task-dependent biasing units to accommodate dynamic situations in task-incremental lifelong learning. Biasing units are programmed by leveraging beneficial perturbations (opposite to well-known adversarial perturbations) for each task. Beneficial perturbations for a given task bias the network toward that task, without affecting the weights of the network, essentially switching the network into a different mode to process that task. This largely eliminates catastrophic interference between tasks. When multiple agents each learn different tasks, task-dependent biasing units can be shared among agents, giving rise to the LLL approach described in [5]. BPN was validated on Incremental MNIST, CIFAR-10, and 8-dataset with one agent, and provided significant improvement in average task accuracy.

- *Lifelong learning algorithm*: **CLR: Channel-wise Lightweight Reprogramming for Continual Learning [13]** - Channel-wise Lightweight Reprogramming (CLR) is another method to help convolutional neural networks (CNNs) overcome catastrophic forgetting during continual learning. A CNN model trained on an old task (or self-supervised proxy task) could be "reprogrammed" to solve a new task by using inexpensive reprogramming parameters. The reprogramming is implemented in additional layers which simply transform a set of incoming CNN feature maps into a new set by applying 3x3 convolutions channel-wise to each map. The 3x3 reprogramming kernels are learned for each new task.

  CLR aims to improve the stability-plasticity trade-off to solve continual learning problems: To maintain stability and retain previous task ability, we use a common task-agnostic immutable part as shared core parameter set. We then add task-specific lightweight reprogramming parameters to reinterpret the outputs of the immutable parts, to enable plasticity and integrate new knowledge. To learn sequential tasks, we only train the lightweight reprogramming parameters to learn each new task. Reprogramming parameters are task-specific and exclusive to each task, which makes this method immune to catastrophic forgetting. In a ShELL setting, the task-specific lightweight reprogramming parameters (CLR-Layers) could be shared among agents.

To minimize the parameter requirement of reprogramming to learn new tasks, we make reprogramming lightweight by only adjusting essential kernels and learning channel-wise linear mappings from anchor parameters to task-specific domain knowledge. We show that, for general CNNs, the CLR parameter increase is less than 0.6% for any new task. Our method outperforms 13 state-of-the-art continual learning baselines on a new challenging sequence of 53 image classification datasets. Implementation available at: `https://github.com/gyhandy/Channel-wise-Lightweight-Reprogramming.git`

1.6 A distillation-based approach integrating continual learning and federated learning for pervasive services [6].

This study investigated the combination of Continual Learning (CL) techniques and Federal Learning (FL) to allow for edge devices (nodes) that are tuned to the specificity of the deployed domain as they encounter new streams of data. The neural model in each edge device is trained using its locally owned data set that corresponds to the current task, while data from previous task is unavailable. The learned models across edge devices are communicated (shared) synchronously to the central server during each communication round. Before the introduction of CL techniques, the sharing of knowledge across all nodes proved beneficial and sometimes enabled nodes to acquire knowledge of a task before it is learned locally. To prevent catastrophic forgetting locally on each edge device, a knowledge distillation CL method was employed. Inspired by Learning without Forgetting (LwF) [14] approach, the study employed two teachers to serve as regularizers for the distillation process. The previous version of a local model and the centralized server model served as teachers for the current local model in each edge device during training on a task. The approach was evaluated in a class incremental setup, using the Human Activity Recognition (HAR) dataset [15] and demonstrated the mitigation of catastrophic forgetting in the FL framework.

1.7 A Rule-based Shield: Accumulating Safety Rules from Catastrophic Action Effects [7].

Representing a shield as a compilation of experienced catastrophic mistakes can promote safer behaviors. However, this approach exhibits limitations in terms of generalizability, particularly in continuous domains where the likelihood of encountering the same state twice is virtually nonexistent. The challenge is tackled by categorizing errors into equivalence classes, each of which is addressed through a set of *safety rules* (predicates) accumulated from learned catastrophic outcomes. For example, when observing a rear-end collision in an autonomous driving task the aim is to obtain a set of rules that prevent all similar future rear-end collisions. In practice, prior to selecting each action for execution by the policy, actions that contravene any safety rule pertaining to the current state are systematically excluded and not considered for selection. In a multi-agent setting, agents collaborate by sharing safety rules instead of pairs of unsafe states and actions. The effectiveness of this approach was evaluated using an autonomous driving simulator where ten decentralized agents encountered a series of randomized scenarios across six driving tasks. The rule-based shield variant, PPO-RS, notably decreased unsafe outcomes and enhanced cumulative rewards compared to the original PPO, ShieldPPO, and other safety-focused RL benchmarks.

- *Lifelong learning algorithm:* **Learning a shield from catastrophic action effects: Never repeat the same mistake [16]**. In unfamiliar environments, learning agents may make errors with potentially catastrophic consequences. While acknowledging the inevitability of agents occasionally making catastrophic errors, this study addresses the crucial task of preventing the *repetition* of such mistakes by autonomous agents. The proposed method employs a *shield*, which restricts agents from executing specific actions in specific states. Catastrophic mistakes made by an agent are recorded, and the agent is prohibited from repeating these mistakes in the future. In a multi-agent setting, catastrophic mistakes made by one agent are shared with the entire group, ensuring that no other agents repeat the same mistakes. The shield, being focused on catastrophic mistakes and independent of the reward function, is task-agnostic and adaptable to lifelong learning scenarios with evolving tasks. The method was evaluated in a grid-based domain, where ten decentralized agents faced a series of navigation challenges in changing environments. The agents employed a variant of the Proximal Policy Optimization (PPO) [17] algorithm, called ShieldPPO, that integrates the shield into the policy to mask unsafe actions. Results indicate that ShieldPPO surpasses PPO and baseline methods from the safe reinforcement learning literature across various settings. Additionally, the sharing of mistakes among agents leads to a linear reduction in catastrophic mistakes relative to the number of agents.

1.8 Asynchronous decentralized federated lifelong learning for landmark localization in medical imaging [8].

Tackling a privacy-preserving medical imaging problem, this study investigated the use of Federated Learning (FL) to learn landmark localization in the brain tumor segmentation (BraTS) dataset [18] with a focus on applicability to edge devices. To prevent catastrophic forgetting, the selective experience replay lifelong learning method was employed [19] through the SERIL lifelong DRL approach described in related paper 1 [20]. Specifically, selected experiences of past tasks were sampled from a replay buffer and interleaved with the current task data during training. A key trait of the FL setup employed was the breakaway from a centralized model, thus allowing a decentralized setup. Furthermore, asynchronous updates between devices were also employed. Thus, the asynchronous decentralized setup enabled data and model heterogeneity and allowed for robustness to connection drops between devices as there is no central point of failure. To prevent excessive communication and sharing overhead, a set of predefined hub devices (nodes) that supported communication only with spatially adjacent devices was set up. During training, each lifelong learner model on a device learns using the current task data, experiences from its own replay buffer, and experiences from the replay buffer (ERB) of other devices. Each agent's personal experiences are shared when training is completed. The ERB datasets are shared from nodes to the agents to help in the training process on-demand. A maximum collective size of 4 was tested across DGX-1 and Google Cloud T4-based systems.

- *Lifelong learning algorithm:* **Multi-environment lifelong deep reinforcement learning for medical imaging [20]**. The nature of medical imaging environments requires deep reinforcement learning agents to constantly evolve and adapt to changes in aspects such as orientation, sequences, and pathologies. This work introduces a novel lifelong DRL framework, SERIL, to learn and adapt to evolving landmark localization tasks on brain MRI images. SERIL is based on a lifelong learning algorithm known as selective experience replay [19]. The approach is tested in a single-agent setup on the 2017 BraTS image dataset.
- *Lifelong learning algorithm:* **Low-compute compression mechanism for edge: A framework for dynamically training and adapting deep reinforcement learning models to different, low-compute, and continuously changing radiology deployment environments [21]**. Low-compute edge devices are heavily used in the medical imaging field. Imaging environments often evolve rapidly and this requires models to continually adapt and adjust. This is difficult to achieve on low-compute devices. This study introduces three novel coreset-based compression algorithms used to improve the per-epoch running time performance for medical imaging localization tasks on low-compute devices i.e., edge devices: neighborhood averaging coreset, neighborhood sensitivity-based sampling coreset, and maximum entropy coreset. These were applied in conjunction with the selective experience replay [19] lifelong learning method to perform localization tasks in DIXON water and DIXON fat MRI images.

2 Application Scenarios and performance of ShELL systems

2.1 ShELL advantages in multiple scenarios.

Fig. 1 expands with graphical illustrations the four scenarios described in Section 5 of the main paper. Constraints, environmental conditions, and the advantages of ShELL are described in conjunction with the graphical representations. The variety of scenarios, also highlighted in Table 1 of the main paper, suggests that performance metrics may vary across those different scenarios. In the following sections we highlight common aspects of performance across ShELL scenarios.

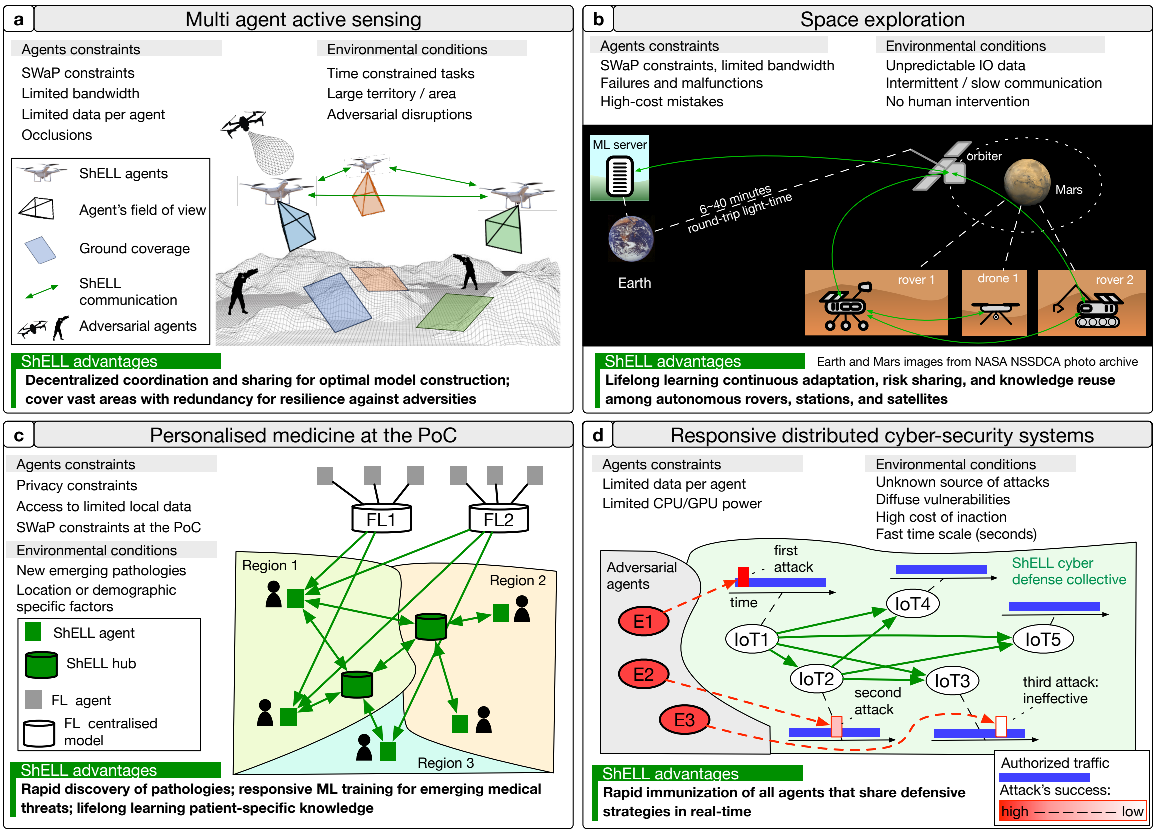

Figure 1: Illustration of ShELL-suitable domains. These four domains were chosen for the potential advantages of lifelong learning and sharing. The illustrations cannot provide all details concerning specific ML tasks that are performed in such domains, but highlight the particular constraints and environmental conditions that make lifelong learning and sharing a highly desirable enhancement to current ML approaches.

2.2 Target performance.

An intuitive idea is that, with an optimal algorithm, n agents may be n times better or n times faster, or a combination of such factors, than the single agent. However, a more careful examination of the problem reveals that a deviation from the n factor is possible in both directions (Figure 2(**c**)). Improved performance or speed that is less than a factor n can derive from efficiency losses, e.g., agents learning the same knowledge from the same data, or communication and integration being time-consuming, delayed, or less effective than learning from their own data. A counter-intuitive notion is that n agents can be better than n times the single agent. This condition may occur in reinforcement learning with large search spaces or sparse rewards where agents can explore different policies in parallel. When an agent finds a sparse reward, this can be shared, and the entire collective can resume learning from the vantage point of one lucky agent, a concept explored in [3]. Similarly, if one agent faces an adverse condition, e.g., hardware damage as a consequence of a policy, such a mistake can be instantly shared to prevent it from happening again to other members of the collective [16, 7]. These learning dynamics bear resemblance with parallel search processes in evolutionary computation [22].

The increased learning speed can be measured with respect to levels of performance on a given number of tasks (Figure 2(**c**, bottom graph)). The acceleration in learning speed can be higher, equal, or lower than n times faster than the single agent, with n being the number of deployed agents. The increase in the speed of learning is particularly desired when rapid responses to new conditions are required.

Other system settings can affect the ShELL metrics, e.g., the available bandwidth and the frequency of communication, the limitation of which could lead to reduced performance. Other more subtle factors relate to the curricula, such as the ratio of the number of tasks over the number of agents, the duration of a learning effort on a given task before switching to other tasks, and the order of tasks. Given n agents and k tasks, having each agent learn a different task could be beneficial in certain domains, while having all agents learn the same task could be better in other scenarios. The ability of an agent to switch to new data, i.e., see a new task, once it has maximized performance on the current tasks, will also affect ShELL metrics. In addition, active learning and coordination can improve the efficiency of collective learning and thus ShELL metrics.

2.3 Measuring performance.

How can the overall performance of a collective of n agents on k tasks be measured? The decentralized nature of ShELL allows this question to be answered in different ways because of the absence of a central integrating server. Even if special nodes are designed to serve as communication hubs, the performance of a ShELL system is intrinsically diffuse across all agents. Testing all agents on all tasks requires considerable computation and stops the collective from learning. Moreover, not all agents might be required to perform all tasks. An interesting possibility is to exploit the ShELL communication capabilities and test only one connected agent on all tasks: this agent will query the collective to gather the necessary information to solve the tasks on which it is being tested. The performance of this agent can be used to assess the collective. Such an agent was named the "evaluation" agent in [3] because it is devoted solely to testing. Using one or more evaluation agents means that the distinction between evaluation blocks and learning blocks [23, 24] is across different agents (evaluation or learning agents). This implies that each learning agent does not need to be stopped at any time for testing. Crucially, the evaluation agent is connected to the collective and can fetch knowledge while being tested, but is invisible to the collective itself, i.e., it does not share knowledge. Thus, the evaluation agent returns the performance of the collective with minimal interference. Other ways to measure performance may be necessary in specific scenarios.

References

[1] M. Rostami, S. Kolouri, K. Kim, and E. Eaton, "Multi-agent distributed lifelong learning for collective knowledge acquisition," in *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018, IFAAMAS, 9 pages*, 2017.

[2] J. Mohammadi and S. Kolouri, "Collaborative learning through shared collective knowledge and local expertise," in *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, 2019.

[3] S. Nath, C. Peridis, E. Ben-Iwhiwhu, X. Liu, S. Dora, C. Liu, S. Kolouri, and A. Soltoggio, "Sharing lifelong reinforcement learning knowledge via modulating masks," in *Second Conference on Lifelong Learning Agents (CoLLAs) 2023*, 2023.

[4] S. Choudhary, S. A. Aketi, G. Saha, and K. Roy, "Codec: Communication-efficient decentralized continual learning," *arXiv preprint arXiv:2303.15378*, 2023.

[5] Y. Ge, Y. Li, D. Wu, A. Xu, A. M. Jones, A. S. Rios, I. Fostiropoulos, P.-H. Huang, Z. W. Murdock, G. Sahin, *et al.*, "Lightweight learner for shared knowledge lifelong learning," *Transactions on Machine Learning Research*, 2023.

[6] A. Usmanova, F. Portet, P. Lalanda, and G. Vega, "A distillation-based approach integrating continual learning and federated learning for pervasive services," *arXiv preprint arXiv:2109.04197*, 2021.

[7] S. S. Shperberg, B. Liu, A. Allievi, and P. Stone, "A rule-based shield: Accumulating safety rules from catastrophic action effects," in *Conference on Lifelong Learning Agents*, pp. 231–242, PMLR, 2022.

[8] G. Zheng, M. A. Jacobs, V. Braverman, and V. S. Parekh, "Asynchronous decentralized federated lifelong learning for landmark localization in medical imaging," *arXiv preprint arXiv:2303.06783*, 2023.

[9] E. Ben-Iwhiwhu, S. Nath, P. K. Pilly, S. Kolouri, and A. Soltoggio, "Lifelong reinforcement learning with modulating masks," *Transactions on Machine Learning Research*, 2023.

[10] G. Saha, I. Garg, and K. Roy, "Gradient projection memory for continual learning," in *International Conference on Learning Representations (ICLR)*, 2021.

[11] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[12] S. Wen, A. Rios, Y. Ge, and L. Itti, "Beneficial perturbation network for designing general adaptive artificial intelligence systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 8, pp. 3778–3791, 2021.

[13] Y. Ge, Y. Li, S. Ni, J. Zhao, M.-H. Yang, and L. Itti, "Clr: Channel-wise lightweight reprogramming for continual learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18798–18808, 2023.

[14] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[15] D. Anguita, A. Ghio, L. Oneto, X. Parra, J. L. Reyes-Ortiz, *et al.*, "A public domain dataset for human activity recognition using smartphones.," in *Esann*, vol. 3, p. 3, 2013.

[16] S. S. Shperberg, B. Liu, and P. Stone, "Learning a shield from catastrophic action effects: Never repeat the same mistake," *arXiv preprint arXiv:2202.09516*, 2022.

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

[18] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, *et al.*, "The multimodal brain tumor image segmentation benchmark (brats)," *IEEE transactions on medical imaging*, vol. 34, no. 10, pp. 1993–2024, 2014.

[19] D. Isele and A. Cosgun, "Selective experience replay for lifelong learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, Apr. 2018.

[20] G. Zheng, S. Lai, V. Braverman, M. A. Jacobs, and V. S. Parekh, "Multi-environment lifelong deep reinforcement learning for medical imaging," *arXiv preprint arXiv:2306.00188*, 2023.

[21] G. Zheng, S. Lai, V. Braverman, M. A. Jacobs, and V. S. Parekh, "A framework for dynamically training and adapting deep reinforcement learning models to different, low-compute, and continuously changing radiology deployment environments," *arXiv preprint arXiv:2306.05310*, 2023.

[22] T. Bäck, D. B. Fogel, and Z. Michalewicz, "Handbook of evolutionary computation," *Release*, vol. 97, no. 1, p. B1, 1997.

[23] A. New, M. Baker, E. Nguyen, and G. Vallabha, "Lifelong learning metrics," *arXiv preprint arXiv:2201.08278*, 1 2022.

[24] M. M. Baker, A. New, M. Aguilar-Simon, Z. Al-Halah, S. M. Arnold, E. Ben-Iwhiwhu, A. P. Brna, E. Brooks, R. C. Brown, Z. Daniels, A. Daram, F. Delattre, R. Dellana, E. Eaton, H. Fu, K. Grauman, J. Hostetler, S. Iqbal, C. Kent, N. Ketz, S. Kolouri, G. Konidaris, D. Kudithipudi, E. Learned-Miller, S. Lee, M. L. Littman, S. Madireddy, J. A. Mendez, E. Q. Nguyen, C. Piatko, P. K. Pilly, A. Raghavan, A. Rahman, S. K. Ramakrishnan, N. Ratzlaff, A. Soltoggio, P. Stone, I. Sur, Z. Tang, S. Tiwari, K. Vedder,

F. Wang, Z. Xu, A. Yanguas-Gil, H. Yedidsion, S. Yu, and G. K. Vallabha, “A domain-agnostic approach for characterization of lifelong learning systems,” *Neural Networks*, vol. 160, pp. 274–296, 3 2023.