Lifelong Reinforcement Learning with Modulating Masks

Anonymous authors
Paper under double-blind review

Abstract

Lifelong learning aims to create AI systems that continuously and incrementally learn during a lifetime, similar to biological learning. Attempts so far have met problems, including catastrophic forgetting, interference among tasks, and the inability to exploit previous knowledge. While considerable research has focused on learning multiple input distributions, typically in classification, lifelong reinforcement learning (LRL) must also deal with variations in the state and transition distributions, and in the reward functions. Modulating masks, recently developed for classification, are particularly suitable to deal with such a large spectrum of task variations. In this paper, we adapted modulating masks to work with deep LRL, specifically PPO and IMPALA agents. The comparison with LRL baselines in both discrete and continuous RL tasks shows superior performance. We further investigated the use of a linear combination of previously learned masks to exploit previous knowledge when learning new tasks: not only is learning faster, the algorithm solves tasks that we could not otherwise solve from scratch due to extremely sparse rewards. The results suggest that RL with modulating masks is a promising approach to lifelong learning, to the composition of knowledge to learn increasingly complex tasks, and to knowledge reuse for efficient and faster learning.

1 Introduction

Lifelong reinforcement learning (LRL) is a recent and active area of research that seeks to enhance current RL algorithms with the following capabilities: learning multiple tasks sequentially without forgetting previous tasks, exploiting knowledge of previous tasks to accelerate learning of new tasks, or building solutions to increasingly complex tasks. The remarkable progress of RL in recent years, stemming in particular from the introduction of deep RL (Mnih et al., 2013), has demonstrated the potential of algorithms that can learn to act in an environment with complex inputs and rewards. However, the limitation of learning only one task means that such algorithms have a narrow focus, and might not be able to scale to complex tasks that first require learning of sub-tasks. It is arguable that acquiring knowledge of multiple tasks over a lifetime, generalizing across such tasks, exploiting acquired knowledge to master new tasks more quickly, or composing simple tasks to solve more complex tasks are necessary to develop more powerful AI systems that better mimic biological intelligence.

LRL shares some of the objectives of continual learning (CL) in classification and supervised learning (Parisi et al., 2019; Hadsell et al., 2020; De Lange et al., 2021), but due to the unique objective and domains of RL, it has developed into a separate field (Khetarpal et al., 2020). A variety of recently developed approaches in CL can be placed under three categories: replay methods, regularization methods, and parameter isolation methods. In LRL, Khetarpal et al. (2020) propose a taxonomy in which approaches can be classified as explicit knowledge retention, leveraging shared structures, and learning to learn. It can be appreciated that LRL exploits most advances in CL while also developing RL-specific algorithms.

Among the CL methods mentioned above, parameter isolation methods have shown state-of-the-art results in classification with approaches such as Progressive Networks (Rusu et al., 2016; Yoon et al., 2018), PackNet (Mallya & Lazebnik, 2018), Piggyback (Mallya et al., 2018) and Supermasks Wortsman et al. (2020). The key concept in such methodologies is to find smart and efficient ways to isolate parameters for each task. In Mallya & Lazebnik (2018), redundancies in large networks are exploited to free parameters to be used only for specific tasks, thus “packing” multiple tasks in one single network. Other approaches instead use the concept of masks (Zhou et al., 2019; Ramanujan et al., 2020; Wortsman et al., 2020) to select parts of a backbone network for each task. Instead of learning the parameters of a backbone network, which are set randomly, masking approaches focus on learning a multiplicative (or modulating) mask for each task. Masks can be effective in binary formats, enabling or disabling parts of the backbone network while requiring low memory. Interestingly, masks can be interpreted as *modulatory* mechanisms with biological inspiration (Kudithipudi et al., 2022) because they exploit a gating mechanism on parameters or activations.

Surprisingly, while the application of masks has been tested extensively in CL for classification, very little is known on their effectiveness in LRL. This study introduces the use of masks with policy optimization algorithms, specifically PPO (Schulman et al., 2017) and IMPALA (Espeholt et al., 2018). We explore (1) the effectiveness of masks to maximize lifelong learning evaluation metrics, and (2) the suitability of masks to reuse knowledge and accelerate learning (forward transfer). To demonstrate the first point, we test the approach on RL curricula with the Minigrid environment (Chevalier-Boisvert et al., 2018), the CT-graph (Soltoggio et al., 2019), Metaworld (Yu et al., 2020), and ProcGen (Cobbe et al., 2020), and assess the lifelong learning metrics (New et al., 2022; Baker et al., 2023) when learning multiple tasks in sequence. To demonstrate the second point, we exploit linear combinations of masks and investigate learning with curricula in which tasks have similarities and measure the forward transfer. To ensure reproducibility, the hyper-parameters for the experiments are reported in Appendix B. The code is anonymously published at `https://github.com/anon-6994/mmn-lrl`.

Contributions. This study introduces the use of modulating masks in LRL. In particular, we employ masks to learn curricula of RL tasks and assess lifelong learning metrics and forward transfer for both discrete and continuous problems. We show that (1) modulating masks are effective in LRL because they maximize lifelong learning metrics when they operate in combination with RL algorithms such as PPO and IMPALA; (2) the gradient search on parameters of a linear combination of masks results in the acceleration of learning of new tasks when such new tasks share properties with previously learned tasks.

2 Related work

2.1 Deep Reinforcement Learning (DeepRL)

The use of deep networks as function approximators in reinforcement learning (RL) has garnered widespread adoption, showcasing results such as learning to play video games (Mnih et al., 2013; Van Hasselt et al., 2016; Shao et al., 2018; Kempka et al., 2016; Mnih et al., 2016; Babaeizadeh et al., 2017; Espeholt et al., 2018), and learning to control actual and simulated robots (Lillicrap et al., 2016; Schulman et al., 2015; 2017; Fujimoto et al., 2018; Haarnoja et al., 2018). These algorithms enable the agent to learn how to solve a single task in a given evaluation domain. In a lifelong learning scenario with multiple tasks, they suffer from lifelong learning challenges such as catastrophic forgetting. Nevertheless, they serve as the basis for lifelong learning algorithms. For example, DQN (Mnih et al., 2013) was combined with EWC (Kirkpatrick et al., 2017) to produce a lifelong learning DeepRL agent (Kessler et al., 2022).

2.2 Lifelong (Continual) Learning

Several advances have recently been introduced in lifelong (continual) learning (Van de Ven & Tolias, 2019; Parisi et al., 2019; Hadsell et al., 2020; De Lange et al., 2021; Khetarpal et al., 2020), addressing challenges such as maintaining performance on previously learned tasks while learning a new task (overcoming forgetting), reusing past knowledge to rapidly learn new tasks (forward transfer), improving performance on previously learned tasks from newly acquired knowledge (backward transfer), the efficient use of model capacity to reduce intransigence, and reducing or avoiding interference across tasks (Kessler et al., 2022). A large body of work focused on lifelong learning in the supervised learning domain and overcoming the challenge of forgetting (Mendez et al., 2022).

Lifelong learning algorithms can be clustered into key categories such as synaptic consolidation approaches (Kirkpatrick et al., 2017; Zenke et al., 2017; Aljundi et al., 2018; Kolouri et al., 2019), memory approaches (Lopez-Paz & Ranzato, 2017; Zeng et al., 2019; Chaudhry et al., 2019; Rolnick et al., 2019; Lin et al., 2022), modular approaches (Rusu et al., 2016; Mallya & Lazebnik, 2018; Mallya et al., 2018; Wortsman et al., 2020; Mendez et al., 2022) or a combination of the above. Synaptic consolidation approaches tackle lifelong learning by discouraging the update of parameters useful for solving previously learned tasks through a regularization penalty. Memory approaches either store and replay samples of previous and current tasks (from a buffer or a generative model) during training (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019; Guo et al., 2020; von Oswald et al., 2020) or project the gradients of the current task being learned in an orthogonal direction to the gradients of previous tasks (Zeng et al., 2019; Farajtabar et al., 2020; Saha et al., 2021; Lin et al., 2022). Memory methods aim to keep the input-output mapping for previous tasks unchanged while learning a new task. Modular approaches either expand the network as new tasks are learned (Rusu et al., 2016; Yoon et al., 2018) or select sub-regions of a fixed-sized network (via masking) to learn tasks (Wortsman et al., 2020).

In the RL domain, a lifelong learner is usually developed by combining a standard deep RL algorithm, either on-policy or off-policy (for example, DQN, PPO, or SAC), with a lifelong learning algorithm. CLEAR (Rolnick et al., 2019) is a lifelong RL that combines IMPALA with a replay method and behavioral cloning. Progress & Compress (Schwarz et al., 2018) demonstrated the combination of IMPALA with EWC and policy distillation techniques. Other notable methods include the combination of a standard deep RL agent (SAC) with PackNet (Mallya & Lazebnik, 2018) as demonstrated in the Continual World robotics benchmark Wołczyk et al. (2021). In addition, Mendez et al. (2022) developed an algorithm combining neural composition, offline RL, and PPO to facilitate knowledge reuse and rapid task learning via functional composition of knowledge. To tackle a lifelong learning scenario with interference among tasks, current methods (Kessler et al., 2020; 2022) employ a multi-head policy network (i.e., a network with shared feature extractor layers connected to different output layers/heads per task) that combine a standard RL algorithm (e.g. DQN or SAC) with synaptic consolidation methods.

2.3 Modulation

Neuromodulatory processes (Avery & Krichmar, 2017; Bear et al., 2020) enable the dynamic alteration of neuronal behavior by affecting synapses or the neurons connected to synapses. Modulation in artificial neural networks (Fellous & Linster, 1998; Doya, 2002) draws inspiration from modulatory dynamics in biological brains that have proven particularly effective in reward-based environments (Schultz et al., 1997; Abbott & Regehr, 2004), in the evolution of reward-based learning (Soltoggio et al., 2008; 2018), and in meta RL (Ben-Iwhiwhu et al., 2022). Modulatory methods in lifelong learning are set up as masks that alter either the neural activations (Serra et al., 2018) or weights of neural networks (Mallya & Lazebnik, 2018; Mallya et al., 2018; Wortsman et al., 2020; Koster et al., 2022). The key insight is the use of a modulatory mask (containing binary or real values) to activate particular network sub-regions and deactivate others when solving a task. Therefore, each task is solved by different sub-regions of the network. For each task, PackNet (Mallya & Lazebnik, 2018) trains the model based on available network capacity and then prunes the network to select only parameters that are important to solving the task. A binary mask representing important and unimportant parameters is then generated and stored for that task, while ensuring that important parameters are never changed when learning future tasks. PiggyBack (Mallya et al., 2018) keeps a fixed untrained backbone network, while it trains and stores mask parameters per task. During learning or evaluation of a particular task, the mask parameters for the task are discretized and applied to the backbone network by modulating its weights. The Supermask (Wortsman et al., 2020) is a generalization of the PiggyBack method that uses a k-winner approach to create sparse masks.

3 Background

3.1 Problem Formulation

A reinforcement learning problem is formalized as a Markov decision process (MDP), with tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, p, r, \gamma \rangle$, where $\mathcal{S}$ is a set of states, $\mathcal{A}$ is a set of actions, $p : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is a transition probability distribution $p(s_{t+1}|s_t, a_t)$ of the next state given the current state and action taken at time t, $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is a reward function that produces a scalar value based on the state and the action taken at the current time step, $\gamma \rightarrow [0, 1]$ is the discount factor that determines how importance future reward relative to the current time step t. An agent interacting in the MDP behaves based on a defined policy (either a stochastic π or a deterministic μ policy). The objective of the agent is to maximize the expected cumulative discounted reward $\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$. It achieves this by learning an optimal policy.

In a lifelong learning setup, a lifelong RL agent is exposed to a sequence of tasks $\mathcal{T}$. Given N tasks, the agent is expected to maximize the RL objective for each task in $\mathcal{T} = \{\tau_1, \tau_2, \ldots, \tau_n\}$. As the agent learns one task after another in the sequence, it is required to maintain (avoid forgetting) or improve (backward transfer of knowledge) performance on previously learned tasks. A desired property of such an agent is the ability to reuse knowledge from previous tasks to rapidly learn the current or future tasks.

3.2 Modulating masks

The concept of modulating masks has recently emerged in the area of supervised classification learning (Mallya et al., 2018; Zhou et al., 2019; Ramanujan et al., 2020; Wortsman et al., 2020; Koster et al., 2022). Masks work by modulating the weights of the network, activating sub-regions of the network, while deactivating other regions. Given a network with layers $1, \ldots L$, with each layer l containing parameters $W^l \in \mathbb{R}^{m \times n}$, for each layer l, a score parameter $S^l \in \mathbb{R}^{m \times n}$ is defined.

During a forward pass (in training or evaluation), a binary mask $M^l \in \{0, 1\}^{m \times n}$ is generated from S^l based on an operation $g(S^l)$ according to one of the following: (i) a threshold ϵ (where values greater than ϵ are set to 1, otherwise 0) Mallya et al. (2018), or (ii) top-k values (the top $k\%$ values in S^l yields 1 in M^l, while the rest are set to 0) Ramanujan et al. (2020), or (iii) probabilistically sampled from a Bernoulli distribution, where the p parameter for the distribution is derived from $sigmoid(S^l)$ (Zhou et al., 2019). An alternative approach is the generation of ternary masks (i.e., with values $\{-1, 0, 1\}$) from S^l, as introduced in Koster et al. (2022). The binary or ternary mask modulates the layer's parameters (or weights), thus activating only a subset of the W^l.

Given an input sample $\mathbf{x} \in \mathbb{R}^m$, a forward pass through the layer is given as $f(\mathbf{x}, W^l, S^l) = (W^l \odot M^l) \cdot \mathbf{x}$, where $\odot$ is the element wise multiplication operation. Only the score parameters S^l are updated during training, while the weights W^l are kept fixed at their initial values. For brevity, the weights and scores across layers are denoted as W and S.

The training algorithm updating S^l depends on the operation used to generate the binary mask. When the binary masks are generated from Bernoulli sampling, standard backpropagation is employed. However, in the case of thresholding or selecting top $k\%$, an algorithm called edge-popup is employed, which combines backpropagation with the straight-through gradient estimator (i.e., where the gradient of the binary mask operation is set to identity to avoid zero-value gradients) (Bengio et al., 2013; Courbariaux et al., 2015).

In a lifelong learning scenario with multiple tasks, for each task τ_k, a score parameter S_k is learned and used to generate and store the mask M_k. During evaluation, when the task is encountered, the mask learned for the task is retrieved and used to modulate the backbone network. Since the weights of the network are kept fixed, it means there is no forgetting. However, this comes at the cost of storage (i.e., storing a mask for each task).

4 Methods

We first introduce the adaptation of modulating masks to RL algorithms (Section 3.2). Following, we hypothesize that previously learned masks can accelerate learning of unseen tasks by means of a linear combination of masks (Section 4.2). Finally, we suggest that continuous masks might be necessary to solve RL tasks in continuous environments (Section 4.3).

4.1 Modulatory masks in Lifelong RL

We posit that a stochastic control policy $\pi_{\theta,\Phi}$ can be parameterized by θ, the weights of a fixed backbone network, and $\Phi = \{\phi_1 \ldots \phi_k\}$, a set of mask score parameters for tasks $1 \ldots k$. ϕ_k are the scores of all layers of the network for task k, comprising the layers $1 \ldots L$, i.e., $\phi_k = \{S_k^1 \ldots S_k^L\}$. The weights θ of the network are randomly initialized using the signed Kaiming constant method (Ramanujan et al., 2020), and are kept fixed during training across all tasks. The mask score parameters Φ are trainable, but only ϕ_k is trained when the agent is exposed to task k. To reduce memory requirements, Wortsman et al. (2020) discovered that masks can be reduced to binary without significant loss in performance. We test this assumption by binarizing masks using an element-wise thresholding function

$$g(\phi_k) = \begin{cases} 1 & \phi_{k,\{i,j\}} > \epsilon \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where ϵ is set to 0. The resulting LRL algorithm is described in Algorithm 1. To assess this algorithm, we paired it with the on-policy PPO (Schulman et al., 2017) algorithm and the off-policy IMPALA (Espeholt et al., 2018) algorithm.

Algorithm 1 Lifelong RL Algorithm with modulating masks

```
Require: Number of tasks $N$, maximum training steps per task P
Require: Rollout length (steps) z
Initialize policy $\pi_{\theta,\Phi}$
for k=1 ... N do
    Set task $k$
    Set $steps = 0$
    while step < P do
        Rollout experiences $\{(s_1, a_1, r_1, s'_1) \dots (s_z, a_z, r_z, s'_z)\}$ using $\pi_{\theta,\phi_k}$.
        Update $steps = steps + z$
        Compute loss $\mathcal{L}_k(\pi_{\theta,\phi_k})$ based on an RL algorithm objective.
        Compute gradients $\nabla_{\phi_k}\mathcal{L}_k(\pi_{\theta,\phi_k})$ with respect to $\phi_k$.
        Update mask score parameter for task $k$, $\phi_k = \phi_k - \alpha\nabla_{\phi_k}\mathcal{L}_k(\pi_{\theta,\phi_k})$
    end while
    Store mask score $\phi_k$ or the binary mask $g(\phi_k)$
end for
```

4.2 Exploiting previous knowledge to learn new tasks

One assumption in LRL, often measured with metrics such as forward and backward transfer (New et al., 2022), is that some tasks in the curriculum have similarities. Thus, previously acquired knowledge, stored in masks, may be exploited to learn new tasks. To test this hypothesis, we propose an incremental approach to using previously learned masks when facing a new task. Rather than learning a large number of masks to be used to infer a linear combination for new tasks, as in Wortsman et al. (2020), we instead bootstrap the learning from the start with any small number of masks combined linearly, plus a trainable random mask $\phi^r = \{S^1_{k+1} \dots S^L_{k+1}\}$. The intuition is that strong linear combination parameters can be discovered quickly if similarities are strong, otherwise more weight is placed on the newly trained mask.

A new mask at layer l is given by

$$S^{l,lc} = \beta^l_{k+1} S^l_{k+1} + \sum_{i=1}^{k} \beta^l_i S^{l,*}_i \tag{2}$$

where $S^l_{k+1} \in \phi_{k+1}$ are the scores of layer l in task $k+1$, $S^{l,lc}$ denotes the transformed scores for task $k+1$ after the linear combination (lc) operation, $S^{l,*}_i$ denotes the optimal scores for previously learned task i, and $\beta^l_1, \dots \beta^l_{k+1}$ are the weights of the linear combination (at layer l). To maintain a normalized weighted sum, a Softmax function is applied to the linear combination parameters before they are applied in Equation 2. When no knowledge is present in the system, the first task is learned starting with a random mask. Task two is learned using $\bar{\beta}_1 = 0.5$, weighting task one's mask, and $\bar{\beta}_2 = 0.5$, weighting the new random mask. The third task will have $\bar{\beta}_1 = \bar{\beta}_2 = \bar{\beta}_3 = 0.33$, and so on.

In short, two approaches can be devised. The first one in which each mask is learned independently of the others. Experimentation of this approach will determine the baseline capabilities of modulating masks in

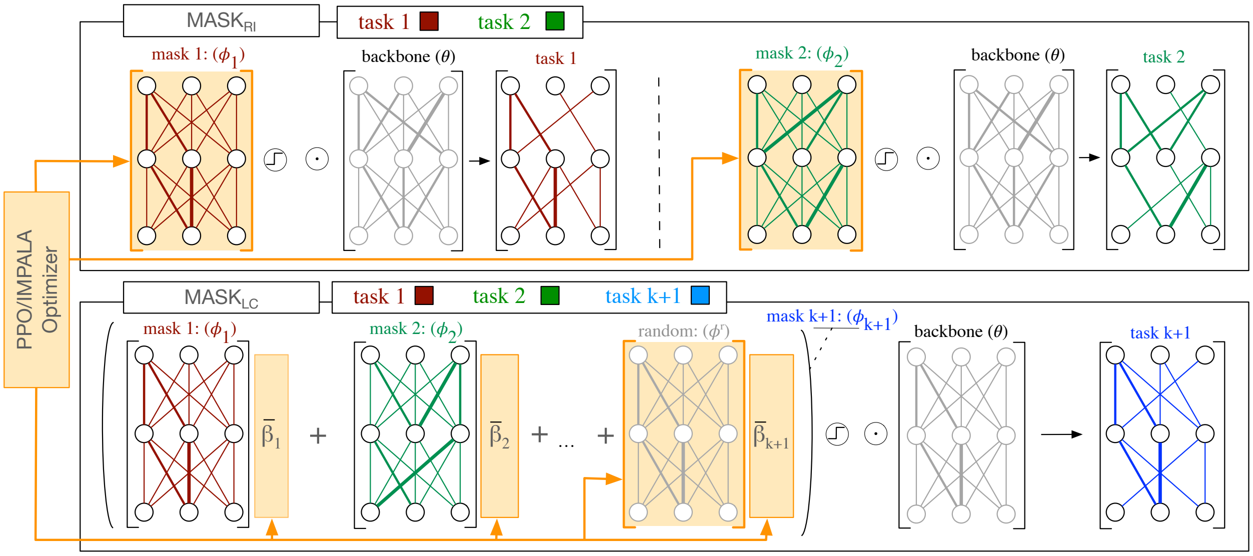


Figure 1: Graphical representations of the methods Mask$_{\mathrm{RI}}$ (top) and Mask$_{\mathrm{LC}}$ (bottom). Mask$_{\mathrm{RI}}$ searches for a new mask for each new task independently. Mask$_{\mathrm{LC}}$ attempts to exploit previous knowledge to learn a new task: known masks are linearly combined with a new randomly initialized mask while learning a new task. Gradient updates search for the parameters $\bar{\beta}_1, ..., \bar{\beta}_n$ and the new random mask.

LRL. We name this *Mask Random Initialization* (Mask$_{\mathrm{RI}}$). The second approach attempts to exploit the knowledge acquired so far during the curriculum learning by using a linear combination of masks to learn a new one. Experimentation of this second approach will determine the capabilities of modulating masks to exploit previously learned knowledge. We name this second approach *Mask Linear Combination* (Mask$_{\mathrm{LC}}$). The idea is graphically summarized in Figure 1.

It can be noted that, as the number of known tasks increases, the relative weight of each mask decreases in Mask$_{\mathrm{LC}}$. This could be a problem, particularly as the weight of the new random mask is reduced, possibly biasing the search excessively towards an average of previous policies. Therefore, we introduce a third approach that attempts to combine the benefits of both Mask$_{\mathrm{RI}}$ and Mask$_{\mathrm{LC}}$: we set the initial weight of the new random mask to 0.5, while the remaining 0.5 weight is shared by the masks of all known tasks. We name this third approach *Balanced Linear Combination* (Mask$_{\mathrm{BLC}}$). It must be noted that the difference between Mask$_{\mathrm{LC}}$ and Mask$_{\mathrm{BLC}}$ is only in the initialization of weights when a new task is encountered, where $\bar{\beta}_{k+1} = 1/(k+1)$ in Mask$_{\mathrm{LC}}$ and $\bar{\beta}_{k+1} = 0.5$ in Mask$_{\mathrm{BLC}}$. However, the parameters $\bar{\beta}$ can be modified arbitrarily by backpropagation during training.

For both Mask$_{\mathrm{LC}}$ and Mask$_{\mathrm{BLC}}$, updates are made only to S^l_{k+1} and $\bar{\beta}^l_i \dots \bar{\beta}^l_{k+1}$ across each layer l. After the training on task $k+1$ is completed, the knowledge from the linear combination is consolidated into the scores for the current task S^l_{k+1} by applying Equation 2. Therefore, the other masks and the linear combination parameters are no longer required. Algorithm 2 reports the forward pass operations for Mask$_{\mathrm{LC}}$.

4.3 Continuous Modulatory Masks for Continuous RL problems

In previous studies, binarization of masks was discovered to be effective in classification to reduce the memory requirement and improve scalability. As shown in our simulations, this approach was effective also in discrete RL problems. However, in continuous action space RL environments, we discovered that binary masks did not lead to successful learning. It is possible that the quantization operation hurts the loss landscape in such environments since the input-output mapping is continuous to continuous values. Therefore, a modification of the supermask method can be devised to support the use of continuous value masks. In the thresholding

Algorithm 2 Forward pass in network layer l in Mask_{LC}

```
Require: Number of task learned so far $k$
Require: $S^*_1 \dots S^*_k$, $S_{k+1}$, $\bar{\beta}_1 \dots \bar{\beta}_{k+1}$, $W$
procedure ForwardPass($\mathbf{x}$)
    if $k = 0$ then
        Set $P = S_1$
    else
        Compute the task score from linear combination: $P = \bar{\beta}_{k+1} S_{k+1} + \sum_{i=1}^{k} \bar{\beta}_i S^*_i$
    end if
    Compute mask from score $M_{k+1} = g(P)$
    Modulate weight $W^{mod} = (W \odot M_{k+1})$
    Compute output $\mathbf{y} = W^{mod} \cdot \mathbf{x}$
    return y
end procedure
```

mask generation operation (given a threshold ϵ), the modified version becomes

$$g(\phi_k) = \begin{cases} \phi_{k,\{i,j\}} & \phi_{k,\{i,j\}} > \epsilon \\ 0 & \text{otherwise} \end{cases} . \tag{3}$$

Such a modification still maintains the ability to learn sparse masks, but replaces the unitary positive values with continuous values. The results of the empirical investigation of the binary and continuous masks in a continuous action space environment is reported in Section 5.3.

5 Experiments

The three novel approaches, Mask_{RI}, Mask_{LC} and Mask_{BLC}, are tested on a set of LRL benchmarks and compared with a lifelong learning baseline, online EWC multi-head (EWC_{MH}), and with the non-lifelong learning algorithm PPO. Experiments with a PPO single task expert (STE) were also conducted to enable the computation of the forward transfer metric for each method, following the setup in Wołczyk et al. (2021). Control experiments with EWC single head, denoted as EWC_{SH}, performed poorly as a confirmation that our benchmarks contain interfering tasks (Kessler et al., 2022): we report those results in the Appendix F.1.

The benchmarks were chosen to assess the robustness of the method against the following aspects: discrete and continuous environments; variations across tasks in input, transition and reward distributions. The CT-graph (Soltoggio et al., 2019) (sparse reward, fast, scalable to large search spaces, variation of reward), Minigrid (Chevalier-Boisvert et al., 2018) (variation of input distributions and reward functions) and Continual World (Wołczyk et al., 2021) (continuous robot-control) were used to assess the approaches $\text{Mask}_{\text{RI/LC/BLC}} + \text{PPO}$. A complete set of hyper-parameters for each experiment is reported in Appendix B.

Additionally, we employed the ProcGen benchmark (Cobbe et al., 2020) that consists of a set of video games with high diversity, fast computation, procedurally generated scenarios, visual recognition and motor control. The masking methods were implemented with IMPALA: $\text{Mask}_{\text{RI/LC/BLC}} + \text{IMPALA}$ (Espeholt et al., 2018) and tested following the ProcGen lifelong RL curriculum presented in Powers et al. (2022) (a subset of games in ProcGen). The properties of the benchmark make the curriculum challenging (e.g., high dimensional RGB observations and procedural generations of levels). The curriculum was designed to test generalization abilities: for each task, the lifelong evaluations are carried out using procedurally generated levels that are unseen by the agents during training. Baselines for this benchmark include online EWC, P&C Schwarz et al. (2018), CLEAR (Rolnick et al., 2019), and IMPALA.

The metrics report a lifelong evaluation across all tasks at different points during the lifelong training, computed as the average sum of reward obtained across all tasks in the curriculum. The area under the curve (AUC) is reported in corresponding tables. A forward transfer metric, following the formulation

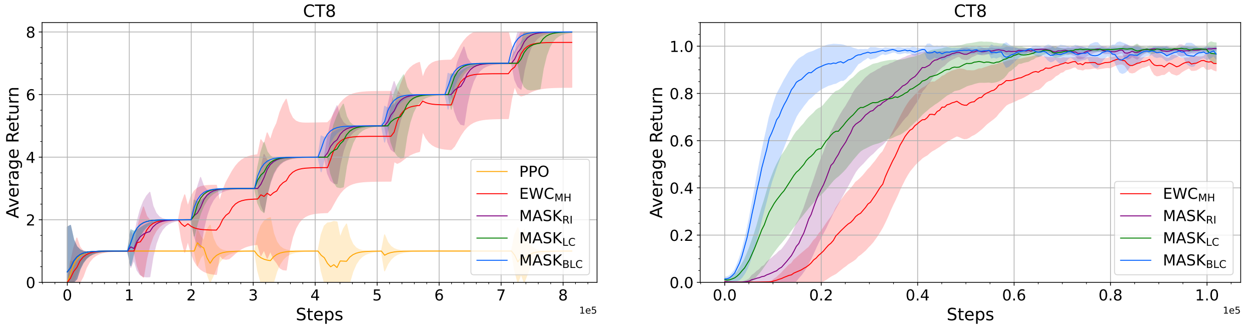


Figure 2: Performance in the *CT8* curriculum. (Left) Lifelong evaluation performance on all tasks (mean and 95% confidence interval on 3 seeds/samples). (Right) Training performance on each task, measured as the average return across all tasks and seeds runs (mean and 95% confidence interval on 8 tasks and 3 seeds, i.e., 24 samples). In the lifelong evaluation, a clear difference can be noted between the lifelong learning algorithms and the non-lifelong learning algorithm PPO, which is therefore not reported in the training plot (Right). The numerical values for the AUC are reported in Tables 1. The training curves show how Mask$_{\text{BLC}}$ is on average faster at learning new tasks, followed by Mask$_{\text{LC}}$ and Mask$_{\text{RI}}$. The performance on each individual task is reported in the Appendix in Figure 19

employed in Wołczyk et al. (2021), is computed for the CT-graph, Minigrid and Continual World. For each task, the forward transfer is computed as the normalized difference between the AUC of the training plot for the lifelong learning agent and the AUC for the reference single task expert. For the ProcGen experiments, the lifelong training and evaluation plot format reported in Powers et al. (2022) was followed to enable an easier comparison with the results in the original paper. As tasks are learned independently of other tasks in Mask$_{\text{RI}}$, there is no notion of forward transfer in the method. Therefore, the method is omitted when forward the transfer metrics are reported.

The results presented in the evaluation plots and the total evaluation metric reported in the tables below were computed as the mean of the 3 seed runs, with the error bars denoting the 95% confidence interval. While the sample size for the evaluation metric is 3, the sample size used for computing the mean and 95% confidence intervals for the forward transfer metric and the training plots is 3 seeds multiplied by the number of tasks per curriculum (i.e., 24, 36, 24, 30, 30 for *CT8*, *CT12*, *CT8 multi depth*, *MG10*, and *CW10* respectively). A significance test was conducted on the results obtained to validate the performance difference between algorithms. The result of the test is reported in Appendix A.

5.1 CT-graph

The configurable tree graph (CT-graph) Soltoggio et al. (2019) is a sparse reward, discrete action space environment with configurable parameters that define the search space by setting the depth and breadth of a tree graph. Due to the exponential growth of the search space with the depth of the tree graph, this benchmark can be set up to the appropriate complexity to test the limits of RL algorithms. Each instance of the environment contains a start state followed by a number of states (2D patterned images) that lead to leaf states and rewards only with optimal policies. A task is defined by setting one of the leaf states as a desired goal state that can be reached only via one trajectory. Two experimental setups were employed: the first with 8 leaf states (reward locations) that serves for 8 tasks, denoted as *CT8* curriculum (depth-3, breadth-2 graph with $2^3 = 8$ leaves). In the second setup, two graph types with 4 and 8 different reward locations, with depth 2 and 3 respectively, result in combined curriculum of 12 tasks, denoted as *CT12*. Such tasks have levels of similarities due to similar input distributions, but also interfere due to opposing reward functions and policies for the same inputs. Additionally, the 8-task graph has a longer path to the reward that introduces variations in both the transition and reward functions. Graphical illustrations of the graphs are provided in Appendix D.

Each task is trained for 102.4K time steps. Figures 2 and 3 report evaluations in the *CT8* and *CT12* curricula as agents are sequentially trained across tasks. The forward transfer and the total evaluation performance metrics are presented in Tables 1 and 2 respectively.

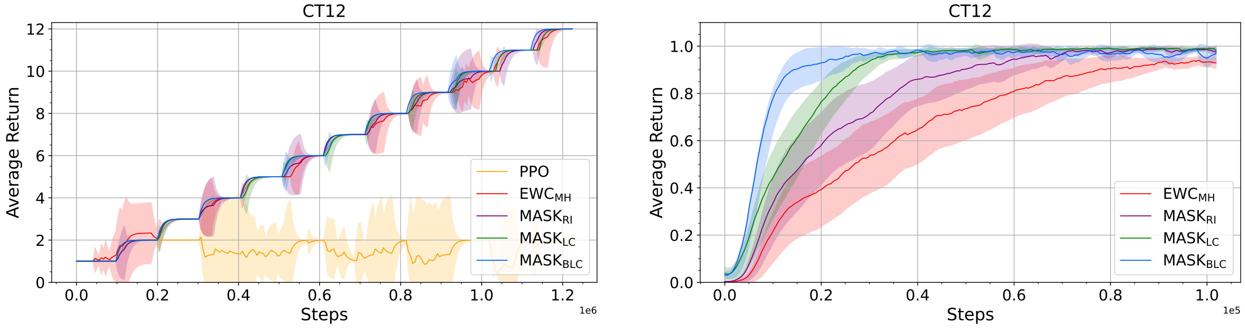


Figure 3: Performance in the *CT12* curriculum. (Left) Lifelong evaluation performance on all tasks (mean and 95% confidence interval on 3 seeds/samples). (Right) Training performance on each task, measured as the average return across all tasks and seeds runs (mean and 95% confidence interval on 12 tasks and 3 seeds, i.e., 36 samples).

Method	Total Eval	Fwd Trnsf.
PPO	151.00 ± 8.61	0.19 ± 0.20
EWC$_{\text{MH}}$	646.33 ± 166.67	-0.15 ± 0.22
Mask$_{\text{RI}}$	699.33 ± 12.75	–
Mask$_{\text{LC}}$	699.33 ± 9.40	0.40 ± 0.19
Mask$_{\text{BLC}}$	717.33 ± 2.87	0.68 ± 0.07

Table 1: Total evaluation return (AUC of the lifelong evaluation plot in Figure 2(Left)) and forward transfer during lifelong training in the *CT8*. Mean ± 95% confidence interval reported.

Method	Total Eval	Fwd Trnsf.
PPO	374.00 ± 39.04	-0.10 ± 0.23
EWC$_{\text{MH}}$	1531.33 ± 45.56	-0.14 ± 0.21
Mask$_{\text{RI}}$	1533.67 ± 26.33	–
Mask$_{\text{LC}}$	1542.67 ± 3.79	0.49 ± 0.11
Mask$_{\text{BLC}}$	1557.33 ± 7.59	0.63 ± 0.07

Table 2: Total evaluation return (AUC of the lifelong evaluation plot in Figure 3(Left)) and forward transfer during lifelong training in the *CT12*. Mean ± 95% confidence interval reported.

The plots show that the masking methods (Mask$_{\text{RI}}$, Mask$_{\text{LC}}$, and Mask$_{\text{BLC}}$) are capable of avoiding forgetting and obtain high evaluation performance, with significantly better forward transfer in comparison to EWC$_{\text{MH}}$ and PPO. On average, the Mask$_{\text{LC}}$ approach recovers performance faster than Mask$_{\text{RI}}$, and Mask$_{\text{BLC}}$ performs best. An expanded version of the training plots showing the learning curves per tasks and averaged across seed runs is reported in the Appendix F.2.

5.2 Minigrid

The Minigrid (Chevalier-Boisvert et al., 2018) is a discrete action space, sparse reward, grid-world navigation environment. The environment consists of a number of predefined partially observable tasks with varying levels of complexity. A consistent theme across all tasks is the requirement of an agent to navigate to a defined location in the grid-world. To achieve this, an agent may need to avoid obstacles such as walls, lava, moving balls, etc. The experiment protocol employs a curriculum of ten tasks (referred to as *MG10*), which consist of two variants of each of the following: SimpleCrossingS9N1, SimpleCrossingS9N2, SimpleCrossingS9N3, LavaCrossingS9N1, LavaCrossingS9N2. Screenshots of all tasks are reported in the Appendix (D.2). The variations across tasks include change in the state distribution and reward function. The results for the *MG10* experiments are presented in Figure 4 and Table 3. The masking methods obtained better performance in comparison to the baselines, with Mask$_{\text{BLC}}$ obtaining the best performance. Appendix F.2 provides a full experimental details.

5.3 Continual World

We evaluated the novel methods in a robotics environment with continuous action space, the Continual World (Wołczyk et al., 2021). The environment was adapted from the MetaWorld environment (Yu et al., 2020) that contains 50 classes of robotics manipulation tasks. The CW10 curriculum consists of 10 robotics tasks: visual screenshots are provided in the Appendix (D.3). The results for all methods were measured using the success rate metric introduced in Yu et al. (2020), which awards a 1 if an agent solves a task or 0 otherwise. For the masking methods in this curriculum, the standard quantization of masks into binary

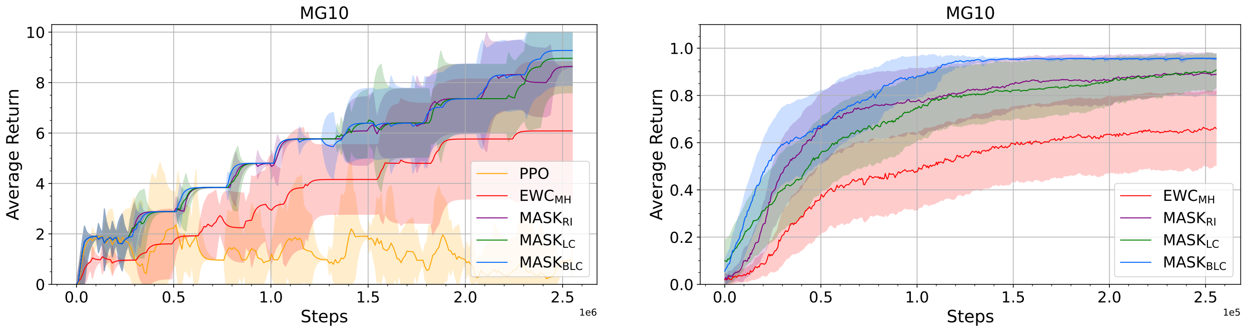


Figure 4: Performance in the *MG10* curriculum. (Left) Lifelong evaluation performance on all tasks (mean and 95% confidence interval on 3 seeds/samples). (Right) Training performance on each task, measured as the average return across all tasks and seeds runs (mean and 95% confidence interval on 10 tasks and 3 seeds, i.e., 30 samples).

Method	Total Eval	Fwd Trnsf.
PPO	296.85 ± 52.09	-0.40 ± 0.41
EWC$_{\text{MH}}$	933.56 ± 332.87	-1.09 ± 0.45
Mask$_{\text{RI}}$	1362.15 ± 119.14	–
Mask$_{\text{LC}}$	1360.80 ± 151.81	-0.19 ± 0.32
Mask$_{\text{BLC}}$	1388.09 ± 156.18	0.22 ± 0.22

Table 3: Total evaluation return (AUC of the lifelong evaluation plot in Figure 4(Left)) and forward transfer in the *MG10*. Mean ± 95% confidence interval reported.

performs poorly. To demonstrate this, two variants are run: the standard setting, where a binary mask is derived from the scores, denoted as Mask$_{\text{RI\_D}}$, and another where a continuous mask is derived from the scores (discussed in Section 4.3), denoted as Mask$_{\text{RI\_C}}$. The results from Figure 5 and Table 4 show that Mask$_{\text{RI\_C}}$ performs significantly better than Mask$_{\text{RI\_D}}$. Motivated by the results, the linear combination of masks method Mask$_{\text{LC}}$ presented for this curriculum also employed the use of continuous masks. Mask$_{\text{LC}}$ and Mask$_{\text{BLC}}$ performed markedly better than the baseline EWC$_{\text{MH}}$ that appears to struggle on this benchmark. Appendix F.2 reports the details for all methods.

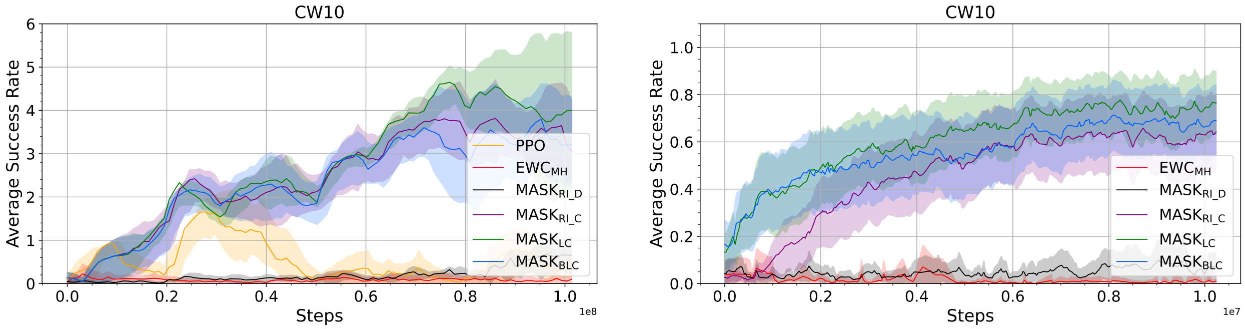


Figure 5: Performance in the *CW10* curriculum measured using the success rate metric. (Left) Lifelong evaluation performance on all tasks (mean and 95% confidence interval on 3 seeds/samples). (Right) Training performance on each task, measured as the average success rate across all tasks and seeds runs (mean and 95% confidence interval on 10 tasks and 3 seeds, i.e., 30 samples).

Method	Total Eval	Fwd Trnsf.
PPO	53.83 ± 72.32	-4.06 ± 2.58
EWC_{MH}	8.60 ± 12.91	-7.39 ± 3.76
$\text{Mask}_{\text{RI\_D}}$	20.45 ± 51.46	–
$\text{Mask}_{\text{RI\_C}}$	246.83 ± 157.39	–
Mask_{LC}	272.43 ± 124.92	-0.33 ± 0.36
Mask_{BLC}	237.00 ± 167.25	-0.61 ± 0.52

Table 4: Total evaluation return (AUC of the lifelong evaluation plot in Figure 5(Left)) and forward transfer in the *CW10*. Mean ± 95% confidence interval reported.

	Evaluation Performance	
Method	Train Tasks	Test Tasks
IMPALA	3687.57 ± 1194.23	2818.44 ± 1214.34
Online EWC	4645.87 ± 1963.42	3831.94 ± 1258.38
P&C	3403.59 ± 3428.37	3390.97 ± 1795.58
CLEAR	11706.34 ± 5271.45	8447.64 ± 2286.21
Mask_{RI}	12462.82 ± 1057.68	12222.10 ± 6064.18
Mask_{LC}	12488.31 ± 3416.90	12377.77 ± 1440.80
Mask_{BLC}	11683.21 ± 3172.61	11913.48 ± 3869.95

Table 5: Total evaluation return (AUC of the lifelong evaluation plot in Figure 6) on the train and test tasks in ProcGen curriculum. Mean ± 95% confidence interval reported.

5.4 ProcGen

The ProcGen (Cobbe et al., 2020) is a discrete action, procedurally generated environment that contains a number of visually diverse video games. It was proposed as a replacement of the Atari games benchmark for lifelong RL, while being computationally faster to simulate than Atari. The procedural nature of the environment facilitates testing of lifelong RL agents in unseen environments, thus evaluating also generalization capabilities. Variation in tasks exists across the state and transition distributions.

The experimental protocol employed follows the setup of Powers et al. (2022), with a sequence of six tasks (0 – Climber, 1 – Dodgeball, 2 – Ninja, 3 – Starpilot, 4 – Bigfish, 5 – Fruitbot) with five learning cycles. Screenshots of the games are reported in the Appendix (D.4). Several environment instances, game levels, and variations in objects, texture maps, layout, enemies, etc., can be generated within a single task. For each task, agents are trained on 200 levels. However, the evaluation is carried out on the distribution of all levels, which is a combination of levels seen and unseen during training.

IMPALA was used as the base RL optimizer on which we deployed the novel masking methods. The results are reported in Figure 6, following the presentation style used in Powers et al. (2022). The masking methods (Mask_{RI}, Mask_{LC}, and Mask_{BLC}) show better performance with respect to other baselines across most tasks, while maintaining generalization capabilities across training and evaluation environments. As the tasks are visually diverse, possibly resulting in less similarity across tasks, reusing previous knowledge may not offer much advantage. Nevertheless, the evaluation performance for each method reported in Table 5 illustrates a significant advantage of the masking methods with respect to the baselines, particularly in the test tasks, where Mask_{LC} is 44% better than the closest runner up (CLEAR) and over 300% better than P&C.

6 Analysis

The results of the previous section prompt the following questions: what linear coefficients emerge after learning? How is rapid learning in Mask_{LC} achieved? How is knowledge reused?

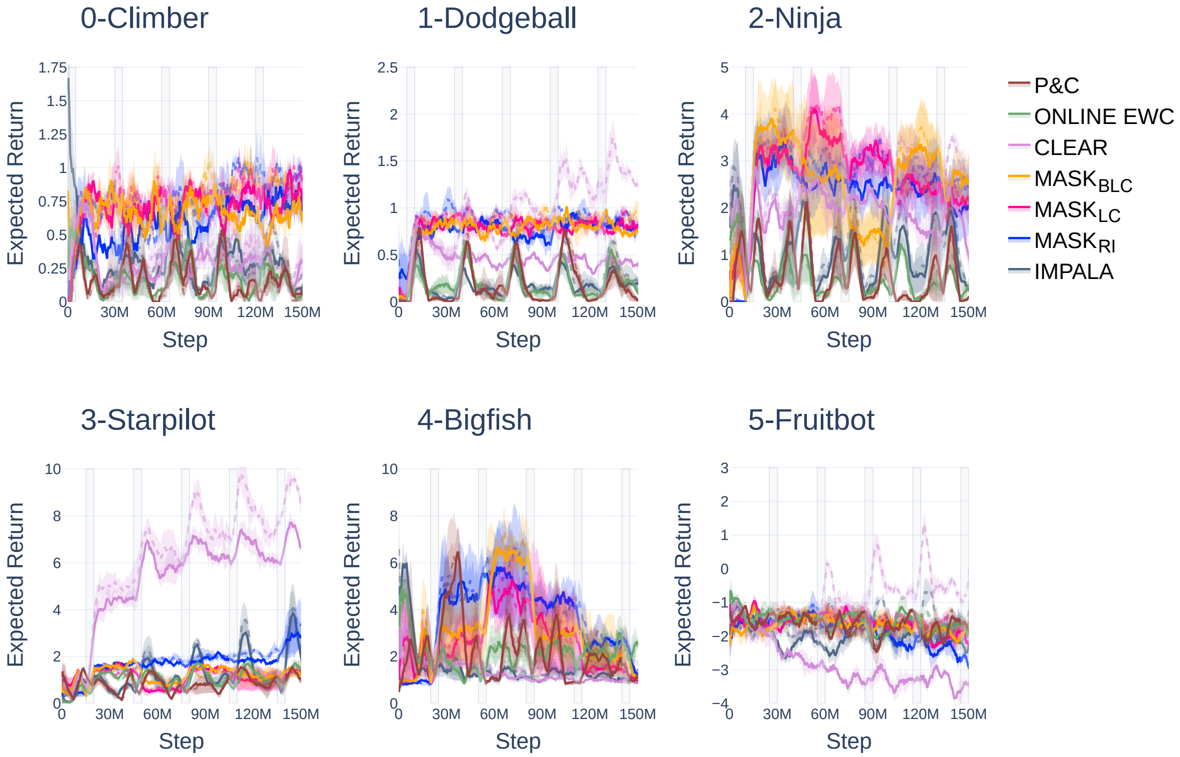


Figure 6: Evaluation results in the ProcGen environment (6 tasks, 5 cycles). The solid line represents evaluation on unseen environments, while the dotted line represents evaluation on the training environments. The gray shaded rectangles show at what point in time an agent is been trained on each task.

6.1 Coefficients for the linear combination of masks

Figure 7 presents the visualization of the coefficients (for the input and output layers) for a Mask$_{\text{LC}}$ agent trained in the *CT8*, *MG10*, and *CW10* curricula respectively. In each plot, each row reports the final set of coefficients after training on a task. For example, the third row in each plot represents the third task in the curriculum and reports three coefficients used for the linear combination of the masks for two tasks and the new mask. For the first task (first row), there are no previous masks to combine.

For the *CT8* and *MG10*, the plots show that the coefficients have similar weights for each task (i.e., row wise in Figure 7). This observation is consistent across the layers of the network (see Appendix E.2 for plots across all layers). This means that the knowledge from all previous tasks is equally important and reused when learning new tasks, possibly indicating that tasks are equally diverse or similar to each other.

Comparing the values across layers (i.e., column wise in Figure 7), we note that there is little variation. In other words, the standard deviation of each vector $\bar{\beta}$ is low, as all values are similar. From such an observation, it follows that the vector $\bar{\beta}$ could be replaced by a scalar for these two benchmarks.

A different picture emerges from the analysis of the coefficients in the CW10 curriculum (Figure 7 rightmost column). Here the coefficients appear to have a larger standard deviation both across masks and across layers. Particular values may suggest relationships between tasks. E.g., the input layer coefficient β_2^1 (from mask 2) is high in task 4 and 7. Similar diverse patterns can be seen in the output layer. We speculate that tasks in CW10 are more diverse and the optimization process is enhancing specific coefficients to reuse specific skills.

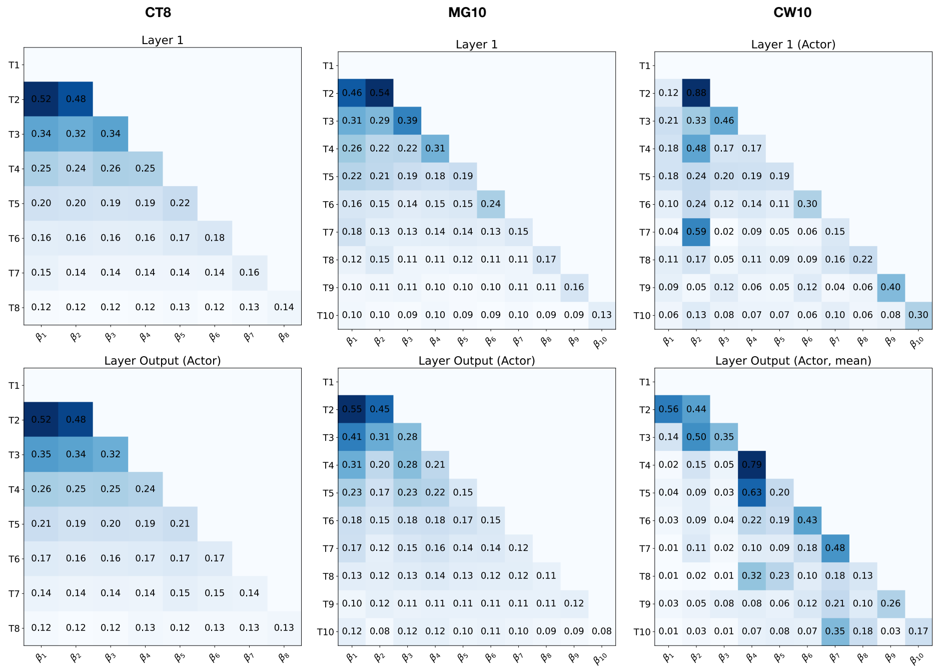


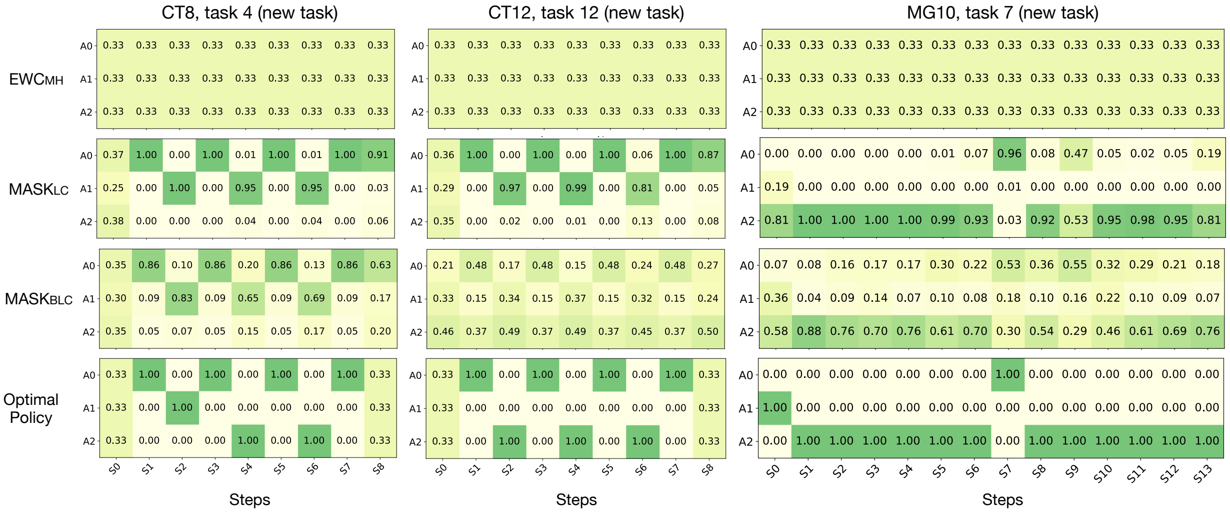
Figure 7: Coefficients $\bar{\beta}$ in Mask_{LC} after training on the *CT8*, *MG10*, and *CW10* curricula. Each row represents a task, and the values (which sums to 1) in it are the final set of coefficients after training on the task: the higher the value of a cell, the higher the level of importance of the corresponding mask in the linear combination operation. The figure only shows coefficients for the first layer and the output (actor) layer. See Appendix E.2 for plots of all other layers.

6.2 Exploitation of previous knowledge

The linear combination of masks appears to accelerate learning significantly as indicated in Figures 2(Right), 3(Right), 4(Right), and 5(Right). To investigate the causes of such learning dynamics, we plot the probabilities of actions during an episode in a new task. The idea is to observe what are the softmax probabilities that are provided by the linear combination of masks at the start of learning for a new task. A full analysis would require the unfeasible task of observing all trajectories: we therefore focus on the optimal trajectory by measuring probabilities when traversing such an optimal trajectory.

Figure 8 shows the analysis for the following cases: facing a fourth task after learning three tasks in the *CT8* benchmark; facing the 12th task after learning 11 tasks in the *CT12* benchmark; facing the 7th task after learning 6 tasks in the *MG10* benchmark. The first row in Figure 8 shows that EWC_{MH} has a purely random policy. Despite learning previous tasks, the new random head results in equal probabilities for all actions. On the contrary, both Mask_{LC} and Mask_{BLC} use previous masks to express preferences. In particular, in the *CT8* and *CT12*, the policy at steps 1, 3, 5 and 7 coincides with the optimal policy for the new task, likely providing an advantage. However, at steps 4 and 6 for the *CT8*, and 2, 4, and 6 for the *CT12*, Mask_{LC} has a markedly wrong policy: this is due to the fact that the new task has a different reward function and is therefore interfering. Due to the balanced combination of previous knowledge with a new mask, Mask_{BLC} seems to strike the right balance between trying known policies and exploring new ones. Such a balanced approach is also visible in the *MG10* task. Here, task 7 (see the Appendix D.2) consists of avoiding the walls and the lava while proceeding ahead, then turning left and reaching the goal: most such skills are similar to those acquired in previously seen tasks, and therefore task 7 is learned starting from a policy that is close to being optimal.

If Mask_{LC} and Mask_{BLC} are capable of exploiting previous knowledge, it is natural to ask whether such knowledge can be exploited to learn increasingly more difficult tasks. The CT-graph (Soltoggio et al., 2019)



CT8, task 4 (new task)

		S0	S1	S2	S3	S4	S5	S6	S7	S8
EWCMH	A0	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
	A1	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
	A2	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
MASKLC	A0	0.37	1.00	0.00	1.00	0.01	1.00	0.01	1.00	0.91
	A1	0.25	0.00	1.00	0.00	0.95	0.00	0.95	0.00	0.03
	A2	0.38	0.00	0.00	0.00	0.04	0.00	0.04	0.00	0.06
MASKBLC	A0	0.35	0.86	0.10	0.86	0.20	0.86	0.13	0.86	0.63
	A1	0.30	0.09	0.83	0.09	0.65	0.09	0.69	0.09	0.17
	A2	0.35	0.05	0.07	0.05	0.15	0.05	0.17	0.05	0.20
Optimal Policy	A0	0.33	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.33
	A1	0.33	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.33
	A2	0.33	0.00	0.00	0.00	1.00	0.00	1.00	0.00	0.33

CT12, task 12 (new task)

		S0	S1	S2	S3	S4	S5	S6	S7	S8
EWCMH	A0	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
	A1	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
	A2	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
MASKLC	A0	0.36	1.00	0.00	1.00	0.00	1.00	0.06	1.00	0.87
	A1	0.29	0.00	0.97	0.00	0.99	0.00	0.81	0.00	0.05
	A2	0.35	0.00	0.02	0.00	0.01	0.00	0.13	0.00	0.08
MASKBLC	A0	0.21	0.48	0.17	0.48	0.15	0.48	0.24	0.48	0.27
	A1	0.33	0.15	0.34	0.15	0.37	0.15	0.32	0.15	0.24
	A2	0.46	0.37	0.49	0.37	0.49	0.37	0.45	0.37	0.50
Optimal Policy	A0	0.33	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.33
	A1	0.33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.33
	A2	0.33	0.00	1.00	0.00	1.00	0.00	1.00	0.00	0.33

MG10, task 7 (new task)

		S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13
EWCMH	A0	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
	A1	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
	A2	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
MASKLC	A0	0.00	0.00	0.00	0.00	0.00	0.01	0.07	0.96	0.08	0.47	0.05	0.02	0.05	0.19
	A1	0.19	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
	A2	0.81	1.00	1.00	1.00	1.00	0.99	0.93	0.03	0.92	0.53	0.95	0.98	0.95	0.81
MASKBLC	A0	0.07	0.08	0.16	0.17	0.17	0.30	0.22	0.53	0.36	0.55	0.32	0.29	0.21	0.18
	A1	0.36	0.04	0.09	0.14	0.07	0.10	0.08	0.18	0.10	0.16	0.22	0.10	0.09	0.07
	A2	0.58	0.88	0.76	0.70	0.76	0.61	0.70	0.30	0.54	0.29	0.46	0.61	0.69	0.76
Optimal Policy	A0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
	A1	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	A2	0.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00

Figure 8: Knowledge reuse when learning a new tasks. The softmax output probabilities are shown during an episode with an unforeseen task (task 4 in *CT8*, task 12 in *CT12*, and task 7 in *MG10*) as the agent is guided through an optimal policy. From the top row down, $\mathrm{EWC}_{\mathrm{MH}}$ shows balanced probabilities for all actions. $\mathrm{Mask}_{\mathrm{LC}}$ displays biased probabilities representing an average behavior across previous tasks. $\mathrm{Mask}_{\mathrm{BLC}}$ shows biased probabilities but contains more randomness in comparison to $\mathrm{Mask}_{\mathrm{LC}}$. A visual comparison with the optimal policy (bottom row) suggests that both $\mathrm{Mask}_{\mathrm{LC}}$ and $\mathrm{Mask}_{\mathrm{BLC}}$ start learning unforeseen tasks with useful knowledge.

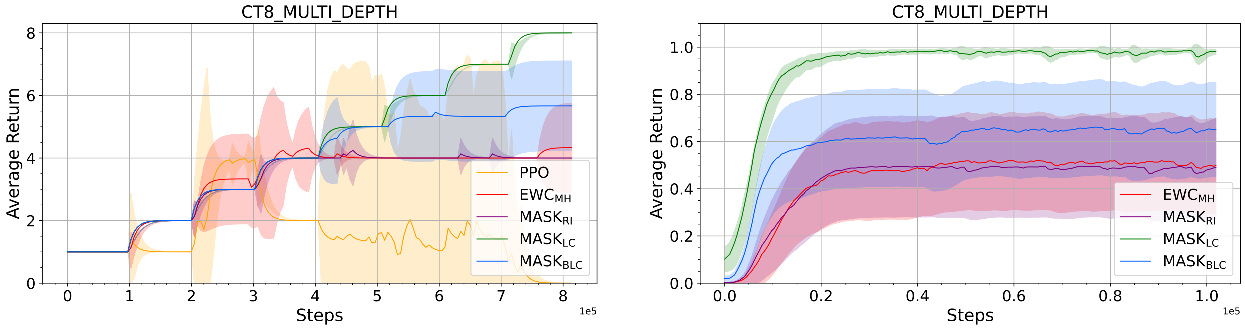


Figure 9: Performance in the *CT8 multi depth* curriculum. (Left) Lifelong evaluation performance on all tasks (mean and 95% confidence interval on 3 seeds/samples). (Right) Training performance on each task, measured as the average return across all tasks and seeds runs (mean and 95% confidence interval on 8 tasks and 3 seeds, i.e., 24 samples). Excluding $\mathrm{MASK}_{\mathrm{LC}}$, the training performance for other methods are sub-optimal, due to the failure to solve later tasks in the curriculum.

environment allows for increasing the complexity by increasing the depth of the graph. In particular, with a depth of 5, the benchmark results in a highly sparse reward environment with a probability of getting a reward with a random policy of only one in $3^{11} = 177,147$ episodes. We therefore designed a curriculum, the *CT8 multi depth*, composed of a set of related but increasingly complex tasks with depth 2, 3, 4 and 5 (two tasks each depth with a different reward function).

Figure 9 shows the performance in the *CT8 multi depth* curriculum. $\mathrm{EWC}_{\mathrm{MH}}$ was able to learn the first 4 tasks with depth 2 and 3, but failed to learn the last 4 more complex tasks. Interestingly, $\mathrm{Mask}_{\mathrm{BLC}}$ managed to learn task 5 and 6 only partially. $\mathrm{Mask}_{\mathrm{LC}}$ was able to learn all tasks, demonstrating that it could reuse previous knowledge to solve increasingly difficult tasks. Figure 10 presents the training performance for each individual task, highlighting were most methods fail in the curriculum.

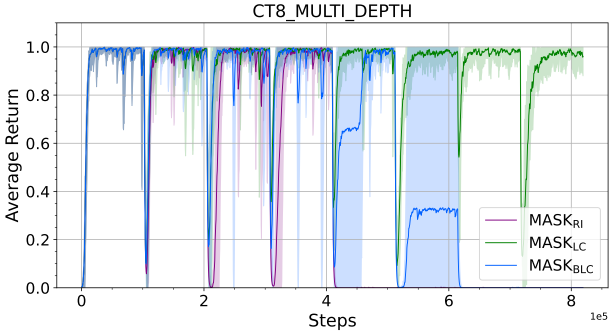


Figure 10: Training performance on each task, measured as the average return across all seeds runs (mean and 95% confidence interval on 3 seeds/samples), expanded from Figure 9(Right).

Method	Total Eval	Fwd Trnsf.
PPO	238.67 ± 283.86	0.10 ± 0.14
EWC$_{\mathrm{MH}}$	530.33 ± 43.48	0.11 ± 0.12
Mask$_{\mathrm{RI}}$	520.00 ± 2.48	–
Mask$_{\mathrm{LC}}$	719.33 ± 1.43	0.77 ± 0.08
Mask$_{\mathrm{BLC}}$	624.67 ± 63.83	0.44 ± 0.15

Table 6: Total evaluation return (AUC of the lifelong evaluation plot in Figure 9(Left)) and forward transfer during lifelong training in the *CT8 multi depth*. Mean ± 95% confidence interval reported.

7 Discussion

The proposed approach employs modulatory masking methods to deliver lifelong learning performance in a range of reinforcement learning environments. The evaluation plots (the left panels of Figures 2, 3, 4 and 5) show how the performance increases as the agent learns through a given curriculum. The monotonic increase, indicating minimal forgetting, is most clear for the CT-graph and Minigrid, while the Continual World appears to have a more noisy performance. The EWC$_{\mathrm{MH}}$ baseline algorithm shows to be a highly performing baseline in the *CT8* and *CT12* curricula, it performs less well in the MG10 curriculum, and poorly in the *CW10* curriculum. The masking methods, instead, perform consistently in all benchmarks, with the linear combination methods, Mask$_{\mathrm{LC}}$ and Mask$_{\mathrm{BLC}}$, showing some advantage over the random initialization Mask$_{\mathrm{RI}}$. Evaluations on the ProcGen environments, while noisy to interpret from Figure 6, reveal that the masking methods outperform IMPALA, Online EWC, P&C and CLEAR by significant margins (Table 5).

While the learning dynamics of the core algorithm Mask$_{\mathrm{RI}}$ indicate superior performance to the baselines, we focused in particular on two extensions of the algorithm, Mask$_{\mathrm{LC}}$ and Mask$_{\mathrm{BLC}}$. These use a linear combination of previously learned masks to search for the optimal policy in a new unforeseen task. These two variations combine previously learned masks with a new random mask to search for the optimal policy. One catch with this approach is that the performance on a new task will depend on which and how many tasks were previously learned. However, this is a property of all lifelong learning algorithms that leverage on previous knowledge. The balanced approach Mask$_{\mathrm{BLC}}$ starts with a 0.5 weight on the new random mask and can be seen as a blend between the random initialization Mask$_{\mathrm{RI}}$ and the linear combination Mask$_{\mathrm{LC}}$. On average, it appears to achieve slightly better performance than either Mask$_{\mathrm{RI}}$ or Mask$_{\mathrm{LC}}$. The standard linear combination Mask$_{\mathrm{LC}}$ was the only algorithm able to learn the most difficult task on the CT-graph. This suggests that the algorithm is capable of exploiting previous knowledge to solve challenging RL problems. The fact that both Mask$_{\mathrm{LC}}$ and Mask$_{\mathrm{BLC}}$ have superior performance to the core random initialization Mask$_{\mathrm{RI}}$ validates the hypothesis that previous knowledge stored in masks can be reused.

The analysis of the coefficients of the linear combination (Section 6.1) reveals that the optimization can tune them to adapt to the nature of the curriculum. In the CT-graph and Minigrid curricula, previous masks are used in balanced proportions. On the Continual World environment, instead, particular masks, and layers within those masks, had significantly larger weights than others. From this observation, we conclude that the new proposed approach may be flexible enough to adapt to a variety of different curricula with different degrees of task similarity.

One concern with modulating masks is that memory requirements increase linearly with the number of tasks. This makes the approach not particularly scalable to large numbers of tasks. However, the promising performance of the linear combination approaches suggests that an upper limit could be imposed on the number of masks to be stored. After such a limit has been reached, new tasks can be learned solely as linear combinations of known masks, significantly reducing memory requirements. While this paper tested vector parameters with a scalar for each network layer, the analysis of the tuned parameters suggests that in some cases a single scalar for each mask could be used, further reducing memory requirements.

In the current study, the binarization process, key to reduce memory use, was successful in the discrete benchmarks and only after performing the linear combination in Mask_{LC} and Mask_{BLC}. Binarized masks resulted in poor performance in the continuous value Continual World benchmark, and if binarization was performed before the linear combination. Further studies could investigate this issue in more detail. It is possible that binarized masks could result in good performance if only the head layer was made continuous, thus ensuring smoothness in the output.

A different approach to reduce memory consumption is to take advantage of the apparent equal representations of known masks (as shown in Figure 7). If the advantage of previous knowledge can be represented as an average of previous masks, it is possible to modify the algorithm to maintain only a moving average of all previous masks. In such a case, the algorithm will combine a new mask with the average of all previous masks. This extreme version of the algorithm is memory efficient, but may under-perform in curricula where coefficients are tuned to be diverse as in the CW10 benchmark (Figure 7 right-most column). Building on this idea, a limited number of *template* masks can be used instead of a single average mask. Each template can be a running average of a cluster of tasks, simply determined by L2 distances of masks, that will ensure good forward transfer while maintaining scalability.

One further approach to reduce memory is to experiment with high level of sparsity in the masks (Equations 1 and 3). Increasing the threshold (currently set to 0), or applying top k-winners as in (Wortsman et al., 2020) for supervised learning, may lead to a meta optimization process where significantly smaller masks maintain acceptable levels of performance. In summary, while more work is required to improve the memory efficiency of the proposed approaches, the success of the linear combination methods suggests venues of research to reduce memory consumption, while the performance advantages justify further research of masking methods in LRL.

8 Conclusion

This work introduces the use of modulating masks for lifelong reinforcement learning problems. Variations of the algorithm are devised to ignore or exploit previous knowledge. All versions demonstrate the ability to learn on sequential curricula and show no catastrophic forgetting thanks to separate mask training. The analysis revealed that using previous masks to learn new tasks is beneficial as the linear combination of masks introduces knowledge in new policies. This finding promises potential new developments for compositional knowledge RL algorithms. Exploiting previous knowledge, one version of the algorithm was able to solve extremely difficult problems with reward probabilities as low as $5.6 \cdot 10^{-6}$ per episode simply using random exploration. These results suggest that modulating masks are a promising tool to expand the capabilities of lifelong reinforcement learning, in particular with the ability to exploit and compose previous knowledge to learn new tasks.

Broader Impact Statement

The advances introduced in this work contribute to the development of intelligent systems that can learn multiple tasks over a lifetime. Such a system has the potential for real-world deployment, especially in robotics and automation of manufacturing processes. As real-world automation increases, it may lead to reduce the demand for human labor in some industries, thereby impacting the economic security of people.

Also, when such systems are deployed, careful considerations are necessary to ensure a smooth human machine collaboration in the workforce, ethical considerations and mitigation of human injuries.

Acknowledgments

Anonymized.

References

LF Abbott and Wade G Regehr. Synaptic computation. *Nature*, 431(7010):796–803, 2004.

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 139–154, 2018.

Michael C Avery and Jeffrey L Krichmar. Neuromodulatory systems and their interactions: a review of models, theories, and experiments. *Frontiers in neural circuits*, pp. 108, 2017.

Mohammad Babaeizadeh, Iuri Frosio, Stephen Tyree, Jason Clemons, and Jan Kautz. Reinforcement learning through asynchronous advantage actor-critic on a gpu. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=r1VGvBcxl.

Megan M. Baker, Alexander New, Mario Aguilar-Simon, Ziad Al-Halah, Sébastien M.R. Arnold, Ese Ben-Iwhiwhu, Andrew P. Brna, Ethan Brooks, Ryan C. Brown, Zachary Daniels, Anurag Daram, Fabien Delattre, Ryan Dellana, Eric Eaton, Haotian Fu, Kristen Grauman, Jesse Hostetler, Shariq Iqbal, David Kent, Nicholas Ketz, Soheil Kolouri, George Konidaris, Dhireesha Kudithipudi, Erik Learned-Miller, Seungwon Lee, Michael L. Littman, Sandeep Madireddy, Jorge A. Mendez, Eric Q. Nguyen, Christine Piatko, Praveen K. Pilly, Aswin Raghavan, Abrar Rahman, Santhosh Kumar Ramakrishnan, Neale Ratzlaff, Andrea Soltoggio, Peter Stone, Indranil Sur, Zhipeng Tang, Saket Tiwari, Kyle Vedder, Felix Wang, Zifan Xu, Angel Yanguas-Gil, Harel Yedidsion, Shangqun Yu, and Gautam K. Vallabha. A domain-agnostic approach for characterization of lifelong learning systems. *Neural Networks*, 2023. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2023.01.007. URL https://www.sciencedirect.com/science/article/pii/S0893608023000072.

Mark Bear, Barry Connors, and Michael A Paradiso. *Neuroscience: Exploring the Brain, Enhanced Edition: Exploring the Brain*. Jones & Bartlett Learning, 2020.

Eseoghene Ben-Iwhiwhu, Jeffery Dick, Nicholas A Ketz, Praveen K Pilly, and Andrea Soltoggio. Context meta-reinforcement learning via neuromodulation. *Neural Networks*, 152:70–79, 2022.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 532–547, 2018.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-GEM. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Hkf2_sC5FX.

Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic gridworld environment for openai gym. https://github.com/maximecb/gym-minigrid, 2018.

Karl Cobbe, Chris Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. In *International conference on machine learning*, pp. 2048–2056. PMLR, 2020.

Cédric Colas, Olivier Sigaud, and Pierre-Yves Oudeyer. How many random seeds? statistical power analysis in deep reinforcement learning experiments. *arXiv preprint arXiv:1806.08295*, 2018.

Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. *Advances in neural information processing systems*, 28, 2015.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021.

Kenji Doya. Metalearning and neuromodulation. *Neural networks*, 15(4-6):495–506, 2002.

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International conference on machine learning*, pp. 1407–1416. PMLR, 2018.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 3762–3773. PMLR, 2020.

Jean-Marc Fellous and Christiane Linster. Computational models of neuromodulation. *Neural computation*, 10(4):771–805, 1998.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Yunhui Guo, Mingrui Liu, Tianbao Yang, and Tajana Rosing. Improved schemes for episodic memory-based lifelong learning. *Advances in Neural Information Processing Systems*, 33:1023–1035, 2020.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Raia Hadsell, Dushyant Rao, Andrei A Rusu, and Razvan Pascanu. Embracing change: Continual learning in deep neural networks. *Trends in cognitive sciences*, 24(12):1028–1040, 2020.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Michał Kempka, Marek Wydmuch, Grzegorz Runc, Jakub Toczek, and Wojciech Jaśkowski. Vizdoom: A doom-based ai research platform for visual reinforcement learning. In *2016 IEEE conference on computational intelligence and games (CIG)*, pp. 1–8. IEEE, 2016.

Samuel Kessler, Jack Parker-Holder, Philip Ball, Stefan Zohren, and Stephen J Roberts. Unclear: A straightforward method for continual reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Samuel Kessler, Jack Parker-Holder, Philip Ball, Stefan Zohren, and Stephen J Roberts. Same state, different task: Continual reinforcement learning without interference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 7143–7151, 2022.

Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. Towards continual reinforcement learning: A review and perspectives. *arXiv preprint arXiv:2012.13490*, 2020.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Soheil Kolouri, Nicholas A Ketz, Andrea Soltoggio, and Praveen K Pilly. Sliced cramer synaptic consolidation for preserving deeply learned representations. In *International Conference on Learning Representations*, 2019.

Nils Koster, Oliver Grothe, and Achim Rettinger. Signing the supermask: Keep, hide, invert. *arXiv preprint arXiv:2201.13361*, 2022.

Dhireesha Kudithipudi, Mario Aguilar-Simon, Jonathan Babb, Maxim Bazhenov, Douglas Blackiston, Josh Bongard, Andrew P Brna, Suraj Chakravarthi Raja, Nick Cheney, Jeff Clune, Anurag Daram, Stefano Fusi, Peter Helfer, Leslie Kay, Nicholas Ketz, Zsolt Kira, Soheil Kolouri, Jeffrey L. Krichmar, Sam Kriegman, Michael Levin, Sandeep Madireddy, Santosh Manicka, Ali Marjaninejad, Bruce McNaughton, Risto Miikkulainen, Zaneta Navratilova, Tej Pandit, Alice Parker, Praveen K. Pilly, Sebastian Risi, Terrence J. Sejnowski, Andrea Soltoggio, Nicholas Soures, Andreas S. Tolias, Darío Urbina-Meléndez, Francisco J. Valero-Cuevas, Gido M. Van de Ven, Joshua T. Vogelstein, Felix Wang, Ron Weiss, Angel Yanguas-Gil, Xinyun Zou, and Hava Siegelmann. Biological underpinnings for lifelong learning machines. *Nature Machine Intelligence*, 4(3):196–210, 2022.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations*, 2016.

Sen Lin, Li Yang, Deliang Fan, and Junshan Zhang. TRGP: Trust region gradient projection for continual learning. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=iEvAf8i6JjO`.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7765–7773, 2018.

Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 67–82, 2018.

Jorge A Mendez, Harm van Seijen, and Eric Eaton. Modular lifelong reinforcement learning via neural composition. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=5XmLzdslFNN`.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937. PMLR, 2016.

Alexander New, Megan Baker, Eric Nguyen, and Gautam Vallabha. Lifelong learning metrics. *arXiv preprint arXiv:2201.08278*, 2022.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

Sam Powers, Eliot Xing, Eric Kolve, Roozbeh Mottaghi, and Abhinav Gupta. Cora: Benchmarks, baselines, and metrics as a platform for continual reinforcement learning agents. In *Conference on Lifelong Learning Agents (CoLLAs)*, 2022.

Vivek Ramanujan, Mitchell Wortsman, Aniruddha Kembhavi, Ali Farhadi, and Mohammad Rastegari. What's hidden in a randomly weighted neural network? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11893–11902, 2020.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in Neural Information Processing Systems*, 32, 2019.

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

Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=3AOj0RCNC2.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

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

Wolfram Schultz, Peter Dayan, and P Read Montague. A neural substrate of prediction and reward. *Science*, 275(5306):1593–1599, 1997.

Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning*, pp. 4528–4537. PMLR, 2018.

Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International Conference on Machine Learning*, pp. 4548–4557. PMLR, 2018.

Kun Shao, Dongbin Zhao, Nannan Li, and Yuanheng Zhu. Learning battles in vizdoom via deep reinforcement learning. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, pp. 1–4. IEEE, 2018.

Andrea Soltoggio, John A Bullinaria, Claudio Mattiussi, Peter Dürr, and Dario Floreano. Evolutionary advantages of neuromodulated plasticity in dynamic, reward-based scenarios. In *Proceedings of the 11th international conference on artificial life (Alife XI)*, number CONF, pp. 569–576. MIT Press, 2008.

Andrea Soltoggio, Kenneth O Stanley, and Sebastian Risi. Born to learn: the inspiration, progress, and future of evolved plastic artificial neural networks. *Neural Networks*, 108:48–67, 2018.

Andrea Soltoggio, Pawel Ladosz, Eseoghene Ben-Iwhiwhu, and Jeff Dick. The CT-graph environments, 2019. URL https://github.com/soltoggio/ct-graph.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Johannes von Oswald, Christian Henning, Benjamin F. Grewe, and João Sacramento. Continual learning with hypernetworks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJgwNerKvB.

Maciej Wołczyk, Michał Zając, Razvan Pascanu, Łukasz Kuciński, and Piotr Miłoś. Continual world: A robotic benchmark for continual reinforcement learning. *Advances in Neural Information Processing Systems*, 34:28496–28510, 2021.

Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. Supermasks in superposition. *Advances in Neural Information Processing Systems*, 33:15173–15184, 2020.

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=Sk7KsfW0-`.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020.

Guanxiong Zeng, Yang Chen, Bo Cui, and Shan Yu. Continual learning of context-dependent processing in neural networks. *Nature Machine Intelligence*, 1(8):364–372, 2019.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pp. 3987–3995. PMLR, 2017.

Hattie Zhou, Janice Lan, Rosanne Liu, and Jason Yosinski. Deconstructing lottery tickets: Zeros, signs, and the supermask. *Advances in neural information processing systems*, 32, 2019.