Appendix

Lifelong Reinforcement Learning with Modulating Masks

A Significance Testing

Results obtained from experiments in RL usually produce high variance across seeds (Henderson et al., 2018). This issue further leads to challenge of reproducible results. To address this concern, a difference test (Colas et al., 2018), using the Welch t-test and bootstrap confidence interval (BCI) were performed on the main results, evaluation performance and forward transfer. The tests were carried out at a significance level of 0.05. The BCI tests were run with 10,000 bootstrap iterations.

The outcome of the evaluation performance tests are reported in Tables 7, 8, 9, and 10, while the forward transfer tests are reported in Tables 11, 12, and 13. The MASK_{LC} was chosen as the method to compare against for the difference testing. The table cells colored green signify that the test reported enough evidence to establish an order relationship between compared methods, and vice versa for cells colored red. Also, when a positive only interval is reported in the BCI test, it signifies that MASK_{LC} has a higher value than the method compared, and vice versa for negative only interval. The number of samples for the evaluation performance test is 3, the number of seed runs per method, while the number of samples for the forward transfer test is 3 multiplied by the number of tasks in each curriculum.

	CT8		CT12		CT8 MD	
Method	p-value	BCI	p-value	BCI	p-value	BCI
PPO	3.48e-10	[544.80, 563.00]	4.87e-09	[1152.20, 1180.80]	3.19e-04	[414.20, 559.20]
EWC_{MH}	1.52e-01	[-6.20, 67.20]	5.73e-01	[-62.00, 47.00]	2.35e-04	[157.80, 207.60]
MASK_{RI}	8.00e-01	[-4.60, 6.00]	5.74e-02	[2.80, 16.20]	2.28e-04	[169.60, 216.60]
MASK_{BLC}	1.16e-03	[-18.60, -10.80]	1.30e-04	[-17.00, -10.60]	7.17e-03	[50.40, 100.60]

Table 7: Total evaluation performance significance testing for the CT-graph curricula. Welch t-test (p-value) and bootstrap confidence interval significance difference testing at 5%, for $\mu_1 - \mu_2$, where μ_1 is the average total evaluation performance achieved by MASK_{LC} and μ_2 that of the comparisons (rows) in the table. Green colored cells are statistically significant result where there is enough evidence to establish an order (difference) between μ_1 and μ_2, and vice versa cells are colored red.

	MG10	
Method	p-value	BCI
PPO	4.16e-07	[1010.67, 1109.96]
EWC_{MH}	4.21e-04	[299.03, 499.05]
MASK_{RI}	5.32e-01	[-79.28, 35.90]
MASK_{BLC}	2.18e-01	[-109.42, 14.67]

Table 8: Total evaluation performance significance testing for the Minigrid curriculum. Welch t-test (p-value) and bootstrap confidence interval (BCI) significance difference testing at 5%, for $\mu_1 - \mu_2$, where μ_1 is the average total evaluation performance achieved by MASK_{LC} and μ_2 that of the comparisons (rows) in the table. Green colored cells are statistically significant result where there is enough evidence to establish an order (difference) between μ_1 and μ_2, and vice versa cells are colored red.

	CW10	
Method	p-value	BCI
PPO	5.98e-03	[161.63, 268.27]
EWC_{MH}	1.12e-02	[206.87, 301.57]
$\text{MASK}_{\text{RI\_D}}$	1.18e-02	[195.02, 290.22]
$\text{MASK}_{\text{RI\_C}}$	6.14e-01	[-46.63, 96.80]
MASK_{BLC}	5.09e-01	[-41.97, 106.07]

Table 9: Total evaluation performance significance testing for the Continual World curriculum. Welch t-test (p-value) and bootstrap confidence interval (BCI) significance difference testing at 5%, for $\mu_1 - \mu_2$, where μ_1 is the average total evaluation performance achieved by MASK_{LC} and μ_2 that of the comparisons (rows) in the table. Green colored cells are statistically significant result where there is enough evidence to establish an order (difference) between μ_1 and μ_2, and vice versa cells are colored red.

Method	Welch test p-value for $\mu_1 - \mu_2$ Train tasks	Test tasks	Confidence Interval for $\mu_1 - \mu_2$ Train tasks	Test tasks
IMPALA	4.09e-03	3.26e-05	[7421.60, 9977.69]	[8912.31, 10226.61]
Online EWC	2.63e-03	4.94e-05	[6276.49, 9157.50]	[7858.06, 9233.60]
P&C	1.28e-03	1.01e-04	[7426.82, 10827.51]	[8170.99, 9802.61]
CLEAR	6.25e-01	5.77e-03	[-1468.47, 3087.61]	[2982.40, 4946.90]
MASK_{RI}	9.78e-01	9.23e-01	[-1474.67, 1156.01]	[-1844.39, 2830.19]
MASK_{BLC}	4.99e-01	6.67e-01	[-871.74, 2481.94]	[-846.61, 2214.05]

Table 10: ProcGen Total evaluation performance: Welch t-test and bootstrap confidence interval significance difference testing at 5%, for $\mu_1 - \mu_2$, where μ_1 is the average total evaluation performance achieved by MASK_{LC} and μ_2 that of the comparisons (rows) in the table. Green colored cells are statistically significant result where there is enough evidence to establish an order (difference) between μ_1 and μ_2, and vice versa cells are colored red.

Method	CT8 p-value	CT8 BCI	CT12 p-value	CT12 BCI	CT8 MD p-value	CT8 MD BCI
PPO	9.73e-03	[0.07, 0.54]	5.76e-09	[0.45, 0.84]	1.18e-16	[0.62, 0.87]
EWC_{MH}	2.50e-08	[0.48, 0.91]	3.36e-11	[0.45, 0.77]	1.03e-21	[0.59, 0.78]
MASK_{BLC}	4.23e-03	[-0.33, -0.06]	3.02e-03	[-0.23, -0.05]	1.10e-05	[0.18, 0.42]

Table 11: Forward transfer significance testing for the CT-graph curricula. Welch t-test (p-value) and bootstrap confidence interval significance difference testing at 5%, for $\mu_1 - \mu_2$, where μ_1 is the average total evaluation performance achieved by MASK_{LC} and μ_2 that of the comparisons (rows) in the table. Green colored cells are statistically significant result where there is enough evidence to establish an order (difference) between μ_1 and μ_2, and vice versa cells are colored red.

Method	MG10 p-value	MG10 BCI
PPO	5.11e-01	[-0.29, 0.58]
EWC_{MH}	5.39e-04	[0.36, 1.21]
MASK_{BLC}	2.03e-03	[-0.84, -0.19]

Table 12: Forward transfer significance testing for the Minigrid curriculum. Welch t-test (p-value) and bootstrap confidence interval (BCI) significance difference testing at 5%, for $\mu_1 - \mu_2$, where μ_1 is the average total evaluation performance achieved by MASK_{LC} and μ_2 that of the comparisons (rows) in the table. Green colored cells are statistically significant result where there is enough evidence to establish an order (difference) between μ_1 and μ_2, and vice versa cells are colored red.

Method	CW10 p-value	CW10 BCI
PPO	6.37e-03	[0.94, 5.93]
EWC_{MH}	6.34e-04	[3.30, 10.29]
MASK_{BLC}	3.66e-01	[-0.34, 0.83]

Table 13: Forward transfer significance testing for the Continual World curriculum. Welch t-test (p-value) and bootstrap confidence interval (BCI) significance difference testing at 5%, for $\mu_1 - \mu_2$, where μ_1 is the average total evaluation performance achieved by MASK_{LC} and μ_2 that of the comparisons (rows) in the table. Green colored cells are statistically significant result where there is enough evidence to establish an order (difference) between μ_1 and μ_2, and vice versa cells are colored red.

B Hyper-parameters

In the experiments across the CT-graph, Minigrid and Continual World, all lifelong RL agents were built on top of the PPO algorithm. The hyper-parameters for the experiments are presented in Table 14. The EWC_{MH} and EWC_{SH} lifelong RL methods contain additional hyper-parameters which defines the weight preservation (consolidation) loss coefficient λ and the weight of the moving average α, for the online estimation of the fisher information matrix parameters following Chaudhry et al. (2018). For Continual World, $\alpha = 0.75$ and $\lambda = 1 \times 10^4$, while for the CT-graph and Minigrid experiments, $\alpha = 0.5$ and $\lambda = 1 \times 10^2$. The hyper-parameters for each method were set based on well-established values and preliminary tests. In

each aforementioned benchmark, the hyper-parameters for the PPO algorithm were kept the same across all methods to enable fair comparison.

For the ProcGen experiments, the setup reported in Powers et al. (2022) was followed, with each life-long RL agent built on top of the IMPALA algorithm. The hyper-parameters for the baselines (IMPALA, Progress & Compress (P&C), ONLINE EWC, and CLEAR) were kept the same as in Powers et al. (2022) for the experiments are presented in Table 15. ONLINE EWC contains additional hyper-parameter such as $\lambda = 175$ and replay_buffer_size $= 1 \times 10^6$. For P&C, $\lambda = 3000$, replay_buffer_size $= 1 \times 10^5$, and num_train_steps_of_progress $= 3906$. For CLEAR, replay_buffer_size $= 5 \times 10^6$.

Hyper-parameter	CT8 / CT12 / CT8 MD	MG10	CW10
Learning rate	0.00015	0.00015	0.0005
Optimizer	RMSprop	RMSprop	Adam
Discount factor	0.99	0.99	0.99
Gradient clip	5	5	5
Entropy	0.1	0.1	0.005
GAE	0.99	0.99	0.97
Rollout length	128	128	5120
Num. of workers	4	4	1
PPO ratio clip	0.1	0.1	0.2
PPO optim. epochs	8	8	16
PPO optim. mini batch	64	64	160
Train steps per task	102,400	256,000	10.24M
Train iterations per task: $\frac{\text{trainsteps}}{\text{rollout}\times\text{workers}}$	200	500	2000
Eval. interval	10	20	200
Eval. episodes	10	10	10

Table 14: Hyper-parameters for curricula in the CT-graph (CT8, CT12, CT8 Multi Depth), Minigrid (MG10) and Continual World (CW10) environments.

Hyper-parameter	Value
Num. of workers	64
Batch size	32
Rollout length	20
Entropy	0.01
Learning rate	4×10^{-4}
Optimizer	RMSprop
Gradient clip	40
Discount factor	0.99
Num. of cycles (repeat curriculum)	5
Num. of train step per task per cycle	5M
Num. of eval episodes	10
Eval. interval	0.25M train steps

Table 15: Hyper-parameters for the curriculum in the ProcGen environment.

C Network Specifications

The policy network specification for the CT-graph (i.e., *CT8, CT12, and CT8 multi depth*) and Minigrid (i.e., *MG10*) curricula is presented in Table 16, with ReLU activation function employed. The output of the actor layer produces logits of a categorical distribution.

Layer	Input units	Output units
Linear 1 (shared)	-	200
Linear 2 (shared)	200	200
Linear 3 (shared)	200	200
Linear (actor output)	200	3
Linear (value output)	200	1

Table 16: Network specification of policy network across all methods for CT-graph and Minigrid curricula. Note, for multi head EWC network, the are multiple Linear (actor output) corresponding to the number of tasks.

For the Continual World (i.e, *CW10*) curriculum, the policy network specification is presented in Table 17, with Tanh activation function employed. The output of the actor layer produces the mean and standard deviation of a gaussian distribution.The output of the standard deviation actor output layer is clipped within the range [-0.6931, 0.4055].

Layer	Input units	Output units
Linear (actor body 1)	-	128
Linear (actor body 2)	128	128
Linear (actor output, mean)	128	3
Linear (actor output, log std)	128	3
Linear (value body 1)	-	128
Linear (value body 2)	128	128
Linear (value output)	128	1

Table 17: Network specification of policy network across all methods for Continual World curriculum. Note, for multi head EWC network, the are multiple Linear (actor output) corresponding to the number of tasks.

For the ProcGen environment, the input observation is an RGB image with shape $3 \times 64 \times 64$ and 15 discrete actions. ReLU activation was employed in the network. The policy specification for the network across all methods is presented in Table 18

Layer	Input channels/units	Output channels/units	Kernel	Stride	Pad
Conv 1 (shared)	3	32	[8, 8]	4	0
Conv 2 (shared)	32	64	[4, 4]	2	0
Conv 3 (shared)	64	64	[3, 3]	1	0
Flatten	$4 \times 4 \times 64$	1024	-	-	-
Linear 1 (shared)	1024	512	-	-	-
Linear (actor output)	528	15	-	-	-
Linear (value output)	528	1	-	-	-

Table 18: Network specification for the ProcGen experiments. Note, the number of input units for the actor and value output heads changes to 528 because the one-hot action vector (i.e., size 15) and reward scalar (i.e., size 1) from the previous time step is concatenated to the output of Linear 1.

Note that across all multi-head EWC experiments, the policy network contains multiple actor output layer corresponding to the number of tasks.

C.1 Backbone Network Initialization for Modulatory Masking Methods

Across all experiments, the weights of the backbone network for the modulatory masking methods were initialized using the signed Kaiming constant method, introduced in Ramanujan et al. (2020). The constant

$\pm c$ is the standard deviation of the Kaiming normal (distribution) initialization method, and could vary from layer to layer in the network. Furthermore, the bias parameters were disabled for the backbone networks in the masking methods, following the setup in Wortsman et al. (2020).

D Environments

D.1 CT-graph

The configurable tree graph (CT-graph) (Soltoggio et al., 2019; 2023) is a sparse reward, discrete action space environment with configurable parameters that define the search space. The environment is represented as a graph, where each node is a state represented as a 12×12 gray scale image. There exist a number of state/node types in the environment, which are start (H), wait (W), decision (D), end/leaf (E), and fail (F) state. Each environment instance contains one home state, one fail states, and a number of wait, decision and end states. The goal of an agent is to navigate from the home state to one of the end states designated as the goal — the agent receives a reward of 1 when it enters the goal state, but 0 at every other time step. If the agent takes an incorrect action in any state, the agent transitions to the fail state, after which an environment reset takes it back to the home state.

The size and complexity (search space) of each environment (graph) instance is determined by a set of configuration parameters — hence the term "configurable in the name". Two majors parameters in the CT-graph are the branch b and depth d that defines the branch (i.e., the width or number of decision actions at a decision state) and depth (i.e., the length) of the instantiated graph. The combination of the b and d determine how many end states exist in a each environment instance. Also, b determines the action space of an instance — defined as $b + 1$. The search space grow exponentially as b and d increase. Thus, the benchmark can be set up to the appropriate complexity to test the limits of RL algorithms.

A task is defined by setting one of the leaf states as a desired goal state that can be reached only via one trajectory.

For the *CT8* curriculum, a graph instance with parameter $b = 2$ and $d = 3$ was employed — 2^3 end states. The 8 tasks comprise of each end state designated as the goal/reward location per task. For the *CT12* curriculum, two graph instances with 4 ($b = 2$ and $d = 2$) and 8 ($b = 2$ and $d = 2$) different end/reward states were combined. Additionally, the 8-task graph has a longer path to the reward that introduces variations in both the transition and reward functions. The *CT12* curriculum was based on an interleave of the tasks from both graph instances (i.e., task 1 in 4-tasks, task 1 in 8-tasks, task 2 in 4-tasks, task 2 in 8-tasks, task 3 in 4-tasks, and so on). See Figure 11 for a graphical representation of the 8-tasks and 4-tasks CT-graph. Lastly, the *CT8 multi depth* curriculum was composed of the first two end/goal states in each of the following graph instance: (i) $b = 2$ and $d = 2$, (ii) $b = 2$ and $d = 3$, (iii) $b = 2, d = 4$, (iv) $b = 2, d = 5$.

With a branching factor (breadth) b of 2 across all CT-graph curricula, the action space was defined as 3 (i.e., $b + 1$).

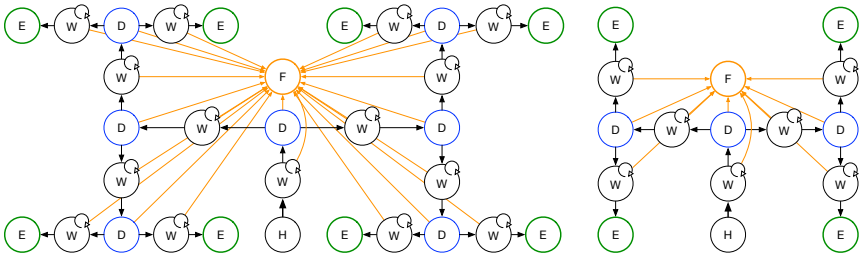

Figure 11: CT-graph environments. States are: home (H), wait (W), decision (D), end (E), fail (F). Three actions at W and D nodes determine the next state. (Left) CT8: a depth-3 graph with three sequential decision states (D). Reward probability $1/3^7 = 1/2187$ reward/episodes. (Right) A depth-2 graph with 4 leaf states (CT4) that combined with CT8 results in the CT12 curriculum.

D.2 Minigrid

Similar to the CT-graph, the Minigrid (Chevalier-Boisvert et al., 2018) is a sparse reward, discrete action navigation environment that consists of a number of predefined partially observable tasks with varying levels of complexity. The environment is setup as a grid world (with fast execution) where an agent is required to navigate to a goal location while avoiding obstacles such as walls, lava, moving balls, etc. It consist of a number of pre-defined grid worlds with several sub-variants defined by changing the random number generator seed. For all Minigrid experiments in this work, the default grid encoding was employed, with each state represented using a tensor of shape $7 \times 7 \times 3$. The agent only get a reward slightly under 1 (depending on the number of steps taken as defined in Equation 4) when it arrives at the goal location, and a reward of 0 at every other state/time step:

$$goal\_reward = 1 - 0.9 \times \frac{es}{ms} \tag{4}$$

where es defines the number of steps taken to navigate to the goal (a green color square), ms is the maximum number of steps the agent is allowed to take in an episode. For *MG10* curriculum, five pre-defined grid-worlds with two seed instances/variants (seed 860 and 861 was employed) per environment (hence 10 tasks) was employed. They are: SimpleCrossingS9N1, SimpleCrossingS9N2, SimpleCrossingS9N3, LavaCrossingS9N1, LavaCrossingS9N2. Figure 12 presents a visual illustration of the 5 grid worlds from which the tasks are derived. Note that when an agent steps a on lava (depicted in orange in the figure), the episode is terminated.

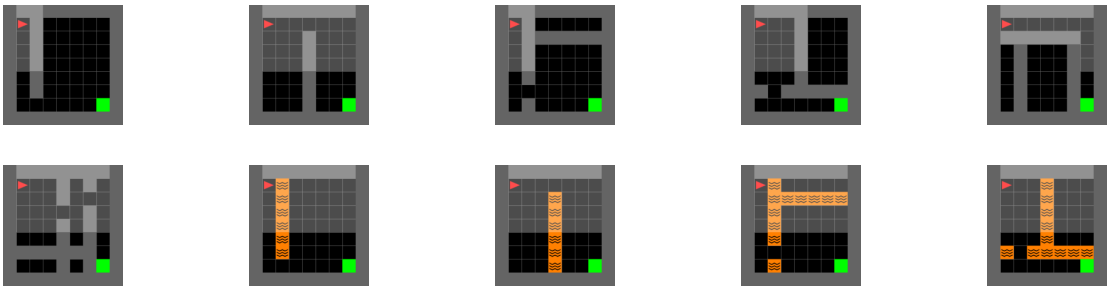

Figure 12: Visual representation of the 10 tasks in the *MG10* curriculum. From left to right, two variants of each class: SimpleCrossingS9N1, SimpleCrossingS9N2, SimpleCrossingS9N3, LavaCrossingS9N1, LavaCrossingS9N2.

Although the default action space in Minigrid is 7, the action space was set to 3 (turn left, turn right, and move forward) in this work as only navigational capabilities were required by the agents across all tasks in the *MG10* curriculum (i.e., other actions such as pick object, drop object, toggle and done actions were not necessary). Furthermore, the reduced action space eased the exploration demands across all methods when learning each task.

D.3 Continual World

The Continual World (Wołczyk et al., 2021) is a benchmark for lifelong/continual RL derived from the Meta-World environment (Yu et al., 2020) — a benchmark consisting of 50 distinct simulated robotic tasks developed using the MuJoCo physics simulator (Todorov et al., 2012). The *CW10* curriculum in the benchmark comes from 10 tasks selected from the Meta World, with the goal of having a high variance in forward transfer across tasks. The 10 tasks (see Figure 13) are: hammer-v2, push-wall-v2, faucet-close-v2, push-back-v2, stick-pull-v2, handle-press-side-v2, push-v2, shelf- place-v2, window-close-v2, peg-unplug-side-v2. The input/state space of each task is a 39 dimension vector representation (consisting of proprioceptor information of the robotic arm as well as position of the objects and goal location in the environment), with an action space of 4 that defines the movement of the robotic arm. The reward function is defined based on a multi-component structure where the agent is reward for achieve sub-goals (i.e., reaching objects, gripping objects, and placing objects or a subset of these) within each task. In addition to the reward, another metric

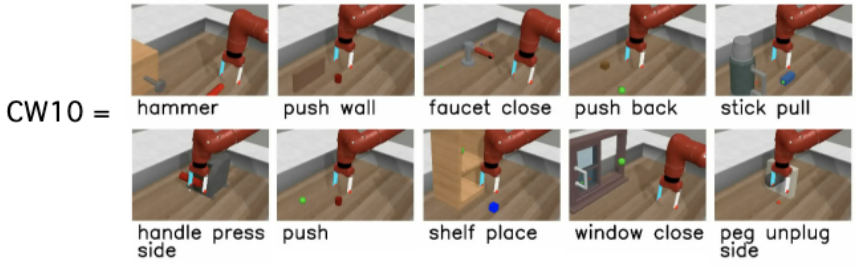

Figure 13: Visual representation of the 10 tasks *CW10* (Wołczyk et al., 2021)

called *success metric* is used to measure performance — where the agent gets a 1 if it solves the overall task or 0 otherwise.

Note that when the Continual World benchmark was released, the authors used what is now termed version 1 (v1) environments in the Meta-World. However, the Meta-World v1 environments contained some issues in the reward function [1] which was fixed in the updated v2 environments. Therefore, the experiments in the paper employed the use of the v2 environment for each task in the Continual World.

D.4 ProcGen

The ProcGen (Cobbe et al., 2020) is a discrete action benchmark that consist of 16 visual diverse video game tasks that are procedurally generated and computationally fast to run, with the aim of evaluating generalization ability of RL agents. It was proposed as a replacement of the Atari games benchmark, while being computationally faster to simulate than Atari. The benchmark was adapted for lifelong RL by (Powers et al., 2022) which introduced a lifelong RL curriculum based on a subset of the ProcGen games. The selected games are Climber, Dodgeball, Ninja, Starpilot, Bigfish, and Fruitbot as shown in Figure 14. The input observation are RGB images of dimension $64 \times 64 \times 3$, along with 15 possible discrete actions. Also, the reward function and scales (range of values) are different for each task in the curriculum.

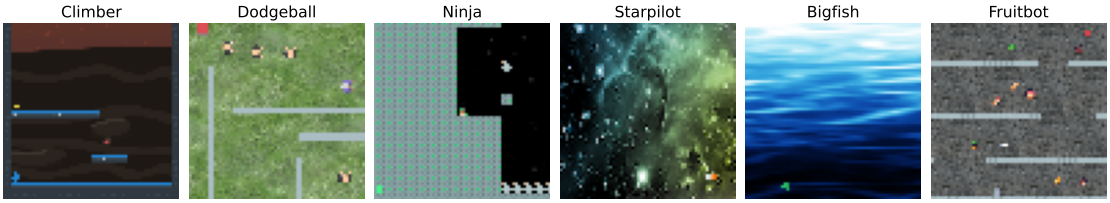

Figure 14: A snapshot of the tasks in the ProcGen curriculum. The texture, objects, RGB color, structure are procedurally generated.

Due to the procedural nature of the environment, each game contains several levels and the properties of each game instance (such as objects, texture maps, layout, enemies etc) can be procedurally generated, thus ensuring high variance within each game. 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.

E Additional Analysis

E.1 Modulatory Mask Similarities

If similarities in tasks allow for our approach to exploit a linear combination of masks, it is reasonable to ask whether masks do reflect such similarity. We consider the two cases of: (1) Mask_{RI} where each mask

[1] as discussed in `https://github.com/rlworkgroup/metaworld/issues/226` and `https://github.com/awarelab/continual_world/issues/2`

is initialized randomly and (2) Mask_{LC} where each mask is a combination of a random mask and known masks. The analysis was conducted on masks learned in the *CT8* curriculum.

Figure 15 shows that, despite task similarities, random initialization of masks results in dissimilar masks. This result is expected as independent gradient optimizations will lead generally to different solutions. However, the linear combination of previously known masks is exploited in the tuning of new masks as we observed that the last two mask are significantly more similar to each other than the first two.

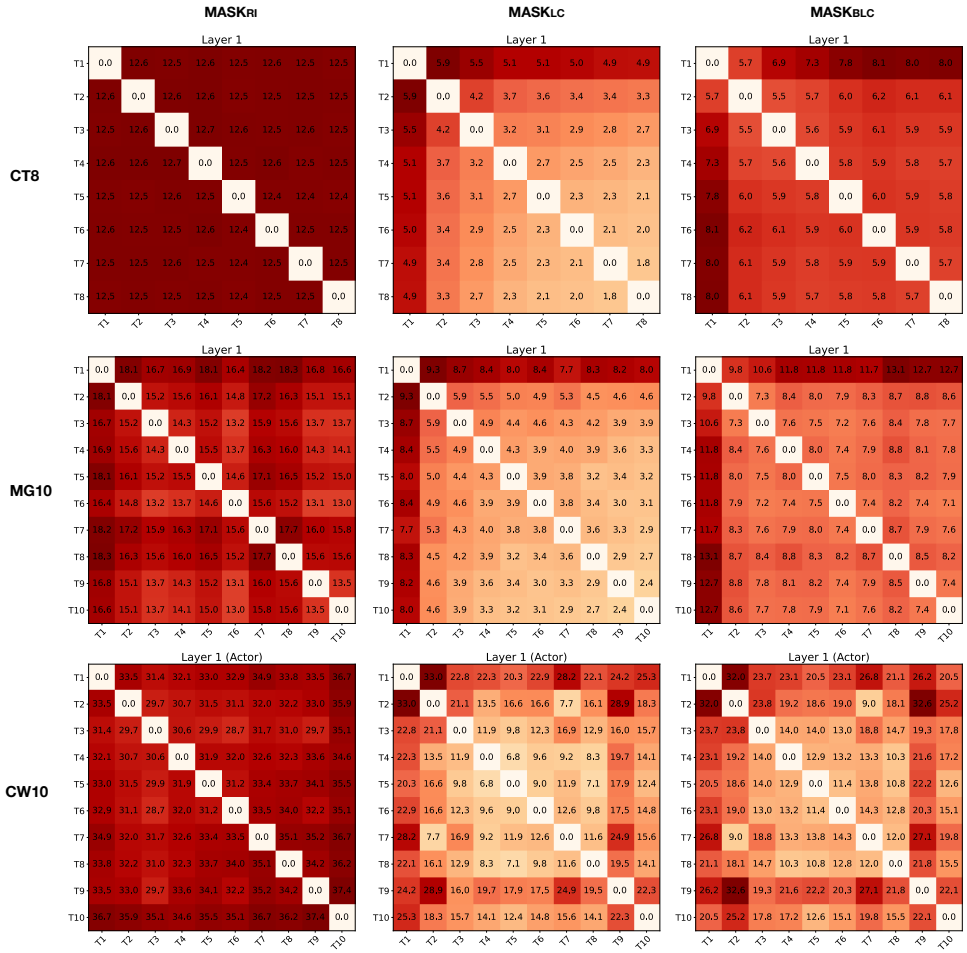

Figure 15: Pairwise mask distances between tasks (i.e., each plot computed as the L2 norm of the difference between task masks) in the first layer of the policy network for each modulatory masking method. Despite tasks having similarities in the CT-graph, Mingrid and Continual World curricula, in MASK_{RI} (Left), the learned masks across tasks show no correlation, MASK_{LC} (Middle) and MASK_{LC} (Right) show mask correlation across tasks (benefiting from knowledge re-use).

E.2 Linear Combination Coefficients

In Section 6.1, Figure 7 showed a summary of the linear combination co-efficients of the input and output layers of the MASK_{LC} network after training. For completeness, this section presents the co-efficients for all layers in the network, across the *CT8*, *CT12*, *CT8 multi depth*, *MG10*, *CW10* curricula. The co-efficients are presented in Figures 16 and 17.

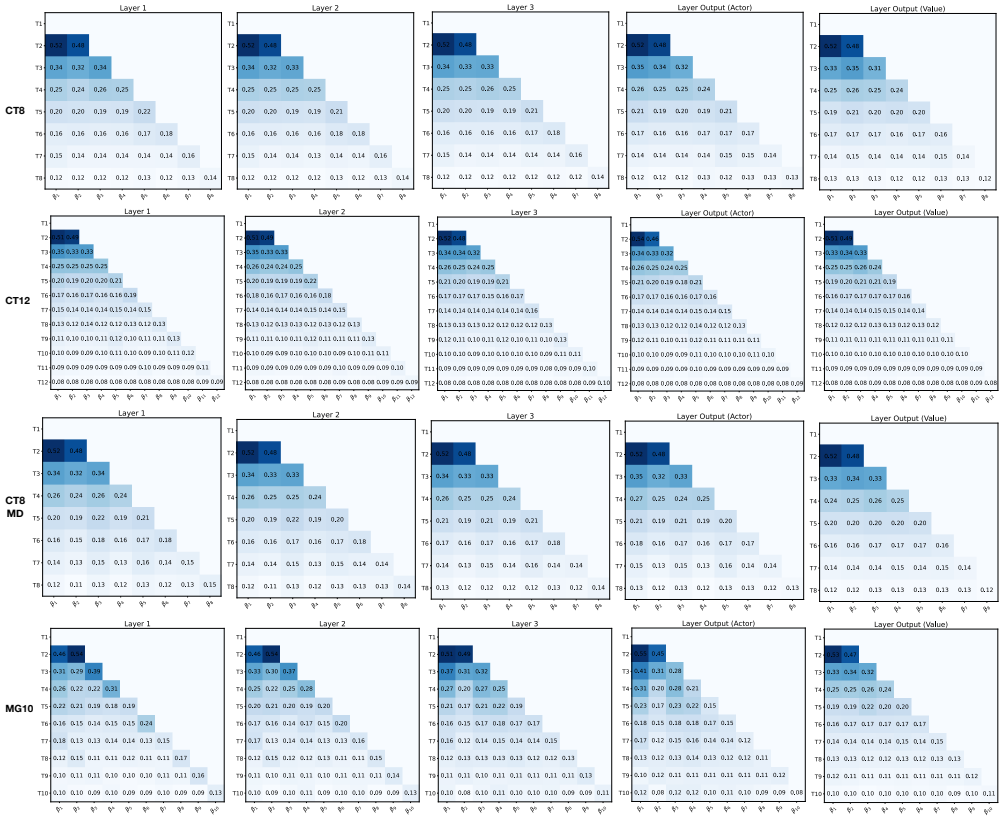

Figure 16: Per layer coefficients β in Mask_{LC} after training on the *CT8*, *CT12*, *CT8 multi depth*, and *MG10* curricula.

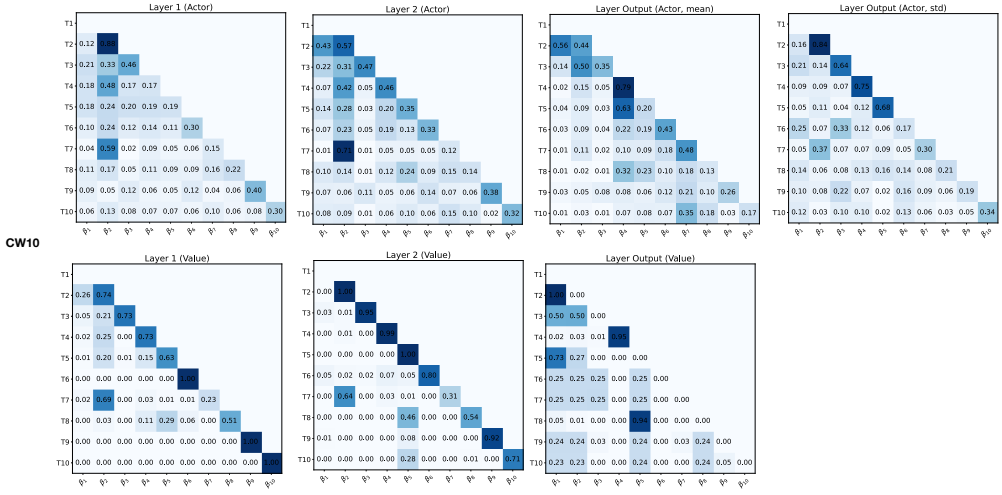

Figure 17: Per layer coefficients β in Mask_{LC} after training on the *CW10* curriculum.

F Additional Results

F.1 EWC Single versus Multi-Head Policy Network

As highlighted in Section 5, the results for the EWC lifelong RL agents presented were based on multi-head (multiple output layers) policy networks, while other methods employed a single head policy network. This is because the EWC single head network EWC_{SH} performed sub-optimally. In the CT-graph *CT8* curriculum, Figure 18 presents the continual evaluation comparison between the EWC_{SH} and EWC_{MH}.

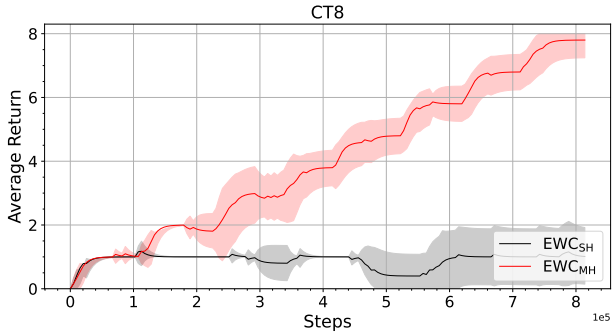

Figure 18: Continual evaluation comparison of EWC single head and multi-head policy networks in the *CT8* curriculum.

F.2 Train plot for all methods

In the lifelong training plots reported in Section 5, only the masking methods were presented for the sake of clarity and readability. The plots in Figure 19 present the lifelong training plots containing all methods across the CT-graph, Minigrid and Continual World curricula.

F.3 Per Task Forward Transfer

In the main text, the forward transfer metric was reported as the averaged across seed runs and tasks in the CT-graph, Minigrid and Continual World curricula. The reported information is expanded in this section to show the forward transfer metric per task (averaged across seed runs only), and reported in Tables 19, 20, 21, 22, and 23. The average across tasks is reported in the last column of each table. As noted in the main text, the tasks are learned independently of other tasks in Mask_{RI}, thus they are omitted in the tables.

	Tasks								
Method	1	2	3	4	5	6	7	8	Avg
PPO	-0.03	0.62	-0.28	-0.19	0.06	0.58	0.17	0.23	0.15
EWC_{MH}	-0.04	0.11	-0.91	-0.67	-0.06	-0.05	-0.22	-0.04	-0.23
MASK_{LC}	0.34	0.81	0.73	0.62	0.13	0.28	0.65	0.15	0.46
MASK_{BLC}	0.34	0.81	0.75	0.73	0.64	0.66	0.72	0.69	0.67

Table 19: Forward transfer per task in the *CT8* curriculum, averaged across seed runs.

F.4 ProcGen: Per Task Forward Transfer Metric

The per task forward transfer metric for all methods except MASK_{RI} in the ProcGen curriculum is presented in Table 24. Note that MASK_{RI} was omitted because the method does not inherently foster forward transfer as each task is learned independently of other tasks (i.e., for each task, a separate modulatory mask is independently initialized and optimized for the task).

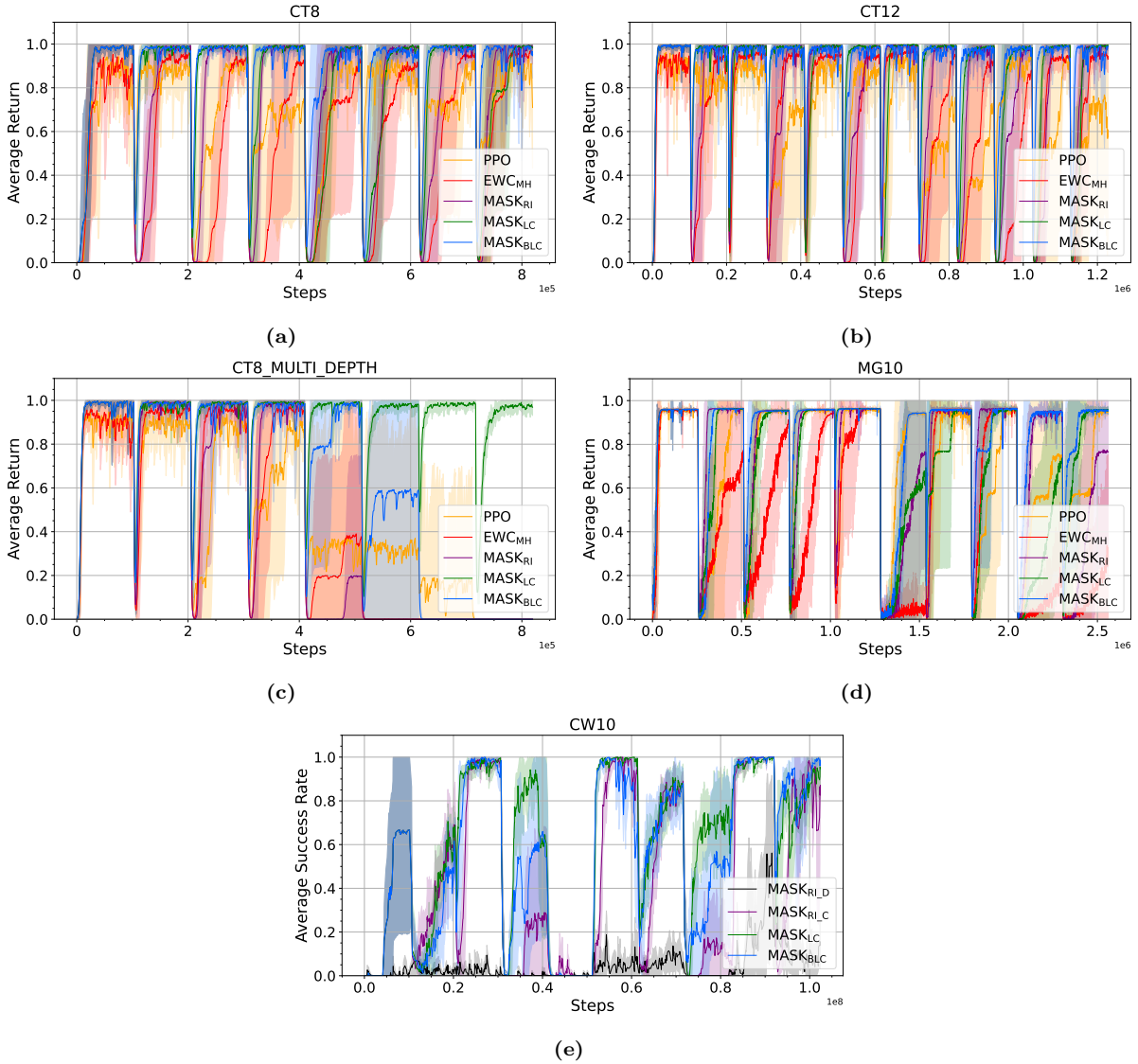

Figure 19: Lifelong training plots for all methods and baselines in CTgraph (CT), Minigrid (MG) and Continual World (CW) curricula: (a) *CT8*, (b) *CT12*, (c) *CT8 multi depth*, (d) *MG10*, and (e) *CW10*.

	Tasks												
Method	1	2	3	4	5	6	7	8	9	10	11	12	Avg
PPO	-0.02	0.30	-0.04	-0.46	-0.33	0.12	-0.46	-0.49	-0.38	0.48	-0.17	-0.36	-0.15
EWC_{MH}	-0.10	-0.22	0.05	0.21	-0.02	-0.49	0.30	-0.44	-0.12	-0.89	0.13	0.24	-0.11
MASK_{LC}	0.50	0.77	0.47	0.83	0.18	0.75	-0.05	0.74	0.66	0.34	0.48	0.36	0.50
MASK_{BLC}	0.50	0.75	0.57	0.79	0.55	0.59	0.45	0.74	0.76	0.58	0.77	0.72	0.65

Table 20: Forward transfer per task in the *CT12* curriculum, averaged across seed runs.

G Learned Modulating Masks and 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

	Tasks								
Method	1	2	3	4	5	6	7	8	Avg
PPO	-0.10	-0.14	-0.23	0.10	0.31	0.31	0.14	0.00	0.05
EWC_{MH}	-0.05	0.06	0.30	0.28	0.23	0.00	0.00	-0.00	0.10
$MASK_{LC}$	0.45	0.54	0.76	0.87	0.94	0.89	0.95	0.87	0.79
$MASK_{BLC}$	0.45	0.55	0.73	0.79	0.83	0.51	0.00	0.00	0.48

Table 21: Forward transfer per task in the *CT8 multi depth* curriculum, averaged across seed runs.

	Tasks										
Method	1	2	3	4	5	6	7	8	9	10	Avg
PPO	0.00	-1.34	-0.17	0.40	-0.20	0.41	-0.55	-1.83	-0.96	0.25	-0.40
EWC_{MH}	-0.01	-2.05	-1.60	-1.47	-0.65	-0.58	-0.10	-0.62	-2.27	-0.99	-1.04
$MASK_{LC}$	0.17	-0.48	-0.17	0.10	0.60	-0.14	-0.71	-1.01	-1.43	0.56	-0.25
$MASK_{BLC}$	0.17	-0.24	0.37	0.47	0.56	0.25	0.48	-0.30	0.31	0.68	0.27

Table 22: Forward transfer per task in the *MG10* curriculum, averaged across seed runs.

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.

H Time taken to reach X% of Optimal (Target) Performance

From the training plots in Figure 19, the time taken or training efficiency (i.e., number of training steps) per task to achieve a certain level of performance (i.e., X% of the optimal performance) can be derived. Note that if an agent fails to achieve the specified performance level during a task training, then the agent is said to have failed that task. In the CT-graph and Minigrid curricula, a target of 75% of the optimal performance was used to conduct the analysis, while 50% was employed in the Continual World curriculum. The results of the analysis are reported in Tables 25, 26, 27, 28, and 29.

Method	Tasks 1	2	3	4	5	6	7	8	9	10	Avg
PPO	-0.87	-1.27	-1.06	0.25	-0.00	-13.79	-2.07	-1.42	-12.70	-7.68	-4.06
EWC_{MH}	-3.01	-1.44	-9.47	0.00	-0.00	-17.60	-2.22	-1.42	-30.07	-8.65	-7.39
$MASK_{LC}$	-1.91	-0.68	0.12	0.63	-0.00	0.54	-0.08	-0.10	-0.44	-1.36	-0.33
$MASK_{BLC}$	-1.91	-0.89	-0.23	0.40	-0.00	-1.84	-0.15	-0.63	-0.34	-0.51	-0.61

Table 23: Forward transfer per task in the *CW10* curriculum, averaged across seed runs.

	0-Climb..	1-Dodge..	2-Ninja	3-Starp..	4-Bigfi..	5-Fruit..	Avg
0-Climb..	–	–	–	–	–	–	–
1-Dodge..	-0.9	–	–	–	–	–	-0.9
2-Ninja	2.6	-3.3	–	–	–	–	-0.3
3-Starp..	-0.7	1.5	0.1	–	–	–	0.3
4-Bigfi..	1.7	-3.6	-1.3	-0.4	–	–	-0.9
5-Fruit..	3.1	-0.9	-1.0	0.5	-0.2	–	0.3
Avg	1.2	-1.6	-0.7	0.0	-0.2	–	-0.2

(a) IMPALA

	0-Climb..	1-Dodge..	2-Ninja	3-Starp..	4-Bigfi..	5-Fruit..	Avg
0-Climb..	–	–	–	–	–	–	–
1-Dodge..	0.0	–	–	–	–	–	0.0
2-Ninja	1.6	-2.2	–	–	–	–	-0.3
3-Starp..	-0.8	2.7	-0.2	–	–	–	0.6
4-Bigfi..	1.7	-3.4	0.1	-0.6	–	–	-0.5
5-Fruit..	0.6	-2.2	0.0	0.5	0.7	–	-0.1
Avg	0.6	-1.3	-0.0	-0.0	0.7	–	-0.1

(b) ONLINE EWC

	0-Climb..	1-Dodge..	2-Ninja	3-Starp..	4-Bigfi..	5-Fruit..	Avg
0-Climb..	–	–	–	–	–	–	–
1-Dodge..	0.0	–	–	–	–	–	0.0
2-Ninja	3.2	0.1	–	–	–	–	1.6
3-Starp..	-4.0	1.1	2.3	–	–	–	-0.2
4-Bigfi..	1.5	0.4	-0.2	0.3	–	–	0.5
5-Fruit..	-0.8	1.2	-1.9	0.3	0.9	–	-0.0
Avg	-0.0	0.7	0.1	0.3	0.9	–	0.3

(c) P&C

	0-Climb..	1-Dodge..	2-Ninja	3-Starp..	4-Bigfi..	5-Fruit..	Avg
0-Climb..	–	–	–	–	–	–	–
1-Dodge..	0.0	–	–	–	–	–	0.0
2-Ninja	0.5	-4.9	–	–	–	–	-2.2
3-Starp..	0.0	1.4	-0.6	–	–	–	0.3
4-Bigfi..	-0.8	-1.5	0.2	-0.2	–	–	-0.6
5-Fruit..	1.0	-1.1	-0.9	0.0	-0.4	–	-0.3
Avg	0.1	-1.5	-0.4	-0.1	-0.4	–	-0.5

(d) CLEAR

	0-Climb..	1-Dodge..	2-Ninja	3-Starp..	4-Bigfi..	5-Fruit..	Avg
0-Climb..	–	–	–	–	–	–	–
1-Dodge..	-0.8	–	–	–	–	–	-0.8
2-Ninja	2.1	-2.0	–	–	–	–	0.1
3-Starp..	-2.2	3.0	-4.8	–	–	–	-1.3
4-Bigfi..	2.3	-2.1	2.0	-2.3	–	–	0.0
5-Fruit..	-1.9	-0.9	2.6	-1.8	-0.8	–	-0.5
Avg	-0.1	-0.5	-0.0	-2.0	-0.8	–	-0.5

(e) MASK LC

	0-Climb..	1-Dodge..	2-Ninja	3-Starp..	4-Bigfi..	5-Fruit..	Avg
0-Climb..	–	–	–	–	–	–	–
1-Dodge..	-0.3	–	–	–	–	–	-0.3
2-Ninja	2.2	-2.1	–	–	–	–	0.0
3-Starp..	-1.4	4.4	-5.0	–	–	–	-0.6
4-Bigfi..	0.1	0.0	2.6	-2.8	–	–	-0.0
5-Fruit..	1.2	-0.0	1.6	-1.9	0.3	–	0.3
Avg	0.4	0.6	-0.2	-2.3	0.3	–	-0.1

(f) MASK BLC

Table 24: ProcGen transfer metrics.

Method	Tasks ($step \times 10^3$) 1	2	3	4	5	6	7	8	Avg
PPO	23	6	30	Fail	35	5	23	16	Fail
EWC_{MH}	23	36	54	54	46	40	44	30	41
$MASK_{RI}$	20	28	21	20	25	30	28	21	24
$MASK_{LC}$	20	7	8	13	39	30	13	28	20
$MASK_{BLC}$	20	8	6	7	14	12	9	8	10

Table 25: Number of training steps taken to achieve 75% optimal (target) performance per task in the *CT8* curriculum. Steps rounded to the nearest thousand.

Method	Tasks ($steps \times 10^3$) 1	2	3	4	5	6	7	8	9	10	11	12	Avg
PPO	8	6	6	Fail	14	18	18	Fail	Fail	11	40	Fail	Fail
EWC_{MH}	9	33	13	25	14	41	9	47	52	69	30	31	31
$MASK_{RI}$	8	23	8	27	9	33	8	24	24	39	35	21	22
$MASK_{LC}$	8	6	8	6	14	8	18	9	15	22	21	25	13
$MASK_{BLC}$	8	6	5	6	6	11	8	7	9	14	8	9	8

Table 26: Number of training steps taken to achieve 75% optimal (target) performance per task in the *CT12* curriculum. Steps rounded to the nearest thousand.

	Tasks ($step \times 10^3$)								
Method	1	2	3	4	5	6	7	8	Avg
PPO	8	8	31	26	Fail	Fail	Fail	Fail	Fail
EWC_{MH}	8	12	18	25	Fail	Fail	Fail	Fail	Fail
$MASK_{RI}$	7	8	22	18	Fail	Fail	Fail	Fail	Fail
$MASK_{LC}$	7	6	7	5	6	10	4	13	7
$MASK_{BLC}$	7	6	7	6	16	Fail	Fail	Fail	Fail

Table 27: Number of training steps taken to achieve 75% optimal (target) performance per task in the *CT8 multi depth* curriculum. Steps rounded to the nearest thousand.

	Tasks ($steps \times 10^3$)										
Method	1	2	3	4	5	6	7	8	9	10	Avg
PPO	29	118	54	19	23	91	36	92	Fail	79	Fail
EWC_{MH}	28	Fail	142	127	46	Fail	31	47	Fail	Fail	Fail
$MASK_{RI}$	22	38	49	37	18	Fail	23	23	63	Fail	Fail
$MASK_{LC}$	22	71	55	37	2	Fail	41	64	Fail	47	Fail
$MASK_{BLC}$	22	60	26	18	3	113	8	38	42	28	36

Table 28: Number of training steps taken to achieve 75% optimal (target) performance per task in the *MG10* curriculum. Steps rounded to the nearest thousand.

	Tasks ($steps \times 10^6$)										
Method	1	2	3	4	5	6	7	8	9	10	Avg
PPO	5.4	0.1	2.5	0.1	Fail	Fail	Fail	Fail	6.5	0.1	Fail
EWC_{MH}	0.7	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail
$MASK_{RI\_D}$	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail	0.1	Fail
$MASK_{RI\_C}$	6.2	0.1	0.1	0.1	Fail	2.8	0.1	0.1	2.8	0.1	Fail
$MASK_{LC}$	6.2	0.1	0.1	0.1	0.1	1.9	0.1	0.1	0.1	0.1	0.9
$MASK_{BLC}$	6.2	0.1	0.1	0.1	0.1	1.9	0.1	0.1	0.3	0.1	0.9

Table 29: Number of training steps taken to achieve 50% optimal (target) performance per task in the *CW10* curriculum. Steps rounded to the hundred thousand.