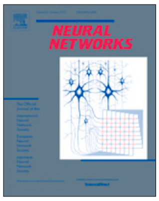
Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet



# Context meta-reinforcement learning via neuromodulation

Eseoghene Ben-Iwhiwhu [a,*], Jeffery Dick [a], Nicholas A. Ketz [b], Praveen K. Pilly [b], Andrea Soltoggio [a]

[a] *Department of Computer Science, Loughborough University, UK*
[b] *HRL Laboratories, Malibu, CA, United States of America*

ARTICLE INFO

*Article history:*
Received 10 November 2021
Received in revised form 11 March 2022
Accepted 5 April 2022
Available online 12 April 2022

*Keywords:*
Meta-learning
Lifelong-learning
Deep reinforcement learning
Neuromodulation

ABSTRACT

Meta-reinforcement learning (meta-RL) algorithms enable agents to adapt quickly to tasks from few samples in dynamic environments. Such a feat is achieved through dynamic representations in an agent's policy network (obtained via reasoning about task context, model parameter updates, or both). However, obtaining rich dynamic representations for fast adaptation beyond simple benchmark problems is challenging due to the burden placed on the policy network to accommodate different policies. This paper addresses the challenge by introducing neuromodulation as a modular component to augment a standard policy network that regulates neuronal activities in order to produce efficient dynamic representations for task adaptation. The proposed extension to the policy network is evaluated across multiple discrete and continuous control environments of increasing complexity. To prove the generality and benefits of the extension in meta-RL, the neuromodulated network was applied to two state-of-the-art meta-RL algorithms (CAVIA and PEARL). The result demonstrates that meta-RL augmented with neuromodulation produces significantly better result and richer dynamic representations in comparison to the baselines.

## 1. Introduction

Human intelligence, though specialized in some sense, is able to generally adapt to new tasks and solve problems from limited experience or few interactions. The field of meta-reinforcement learning (meta-RL) seeks to replicate such a flexible intelligence by designing agents that are capable of rapidly adapting to tasks from few interactions in an environment. The recent progress in the field such as Duan, Schulman, et al. (2016), Finn, Abbeel, and Levine (2017), Gupta, Mendonca, Liu, Abbeel, and Levine (2018), Rakelly, Zhou, Finn, Levine, and Quillen (2019), Wang et al. (2016) and Zintgraf, Shiarli, Kurin, Hofmann, and Whiteson (2019) has showcased start-of-the-art results. Studies with agents endowed with such adaptation capabilities are a promising venue for developing much desired and needed artificial intelligence systems and robots with lifelong learning dynamics.

When an agent's policy for a meta-RL problem is encoded by a neural network, neural representations are adjusted from a base pre-trained point to a configuration that is optimal to solve a specific task. Such dynamic representations are a key feature to enable an agent to rapidly adapt to different tasks. These representations can be derived from gradient-based approaches (Finn et al., 2017), context-based approaches such as memory (Duan, Schulman, et al., 2016; Mishra, Rohaninejad, Chen, & Abbeel, 2018; Wang et al., 2016) and probabilistic (Rakelly et al., 2019), or hybrid approaches (i.e., combination of gradient and context methods) (Zintgraf et al., 2019). The hybrid approach obtains a task context via gradient updates and thus dynamically alters the representations of the network. Context approaches such as CAVIA (Zintgraf et al., 2019) and PEARL (Rakelly et al., 2019) are more interpretable as they disentangle task context from the policy network, thus the task context is used to achieve optimal policies for different tasks.

One limitation of such approaches is that they do not scale well as the problem complexity increases because of the demand to store many diverse policies to be reached within a single network. In particular, it is possible that, as tasks grow in complexity, the tasks similarities reduce and thus the network's representations required to solve each task optimally become dissimilar. We hypothesize that standard policy networks are not likely to produce diverse policies from a trained base representation because all neurons have a homogeneous role or function: thus, significant changes in the policy require widespread changes across the network. From this observation, we speculate that a network endowed with modulatory neurons (neuromodulators) has a significantly higher ability to modify its policy.

Our approach to overcome this limiting design factor in current meta-RL neural approaches is to introduce a neuromodulated

[*] Corresponding author.
*E-mail addresses:* e.ben-iwhiwhu@lboro.ac.uk (E. Ben-Iwhiwhu), j.dick@lboro.ac.uk (J. Dick), naketz@hrl.com (N.A. Ketz), pkpilly@hrl.com (P.K. Pilly), a.soltoggio@lboro.ac.uk (A. Soltoggio).

https://doi.org/10.1016/j.neunet.2022.04.003

policy network to increase its ability to encode rich and flexible dynamic representations. The rich representations are measured based on the dissimilarity of the representations across various tasks, and are useful when the optimal policy of an agent (input-to-action mapping) is less similar across tasks. When combined with the CAVIA and PEARL meta-learning frameworks, the proposed approach produced better dynamic representations for fast adaptation as the neuromodulators in each layer serve as a means of directly altering the representations of the layer in addition to the task context.

Several designs exist for neuromodulation (Doya, 2002), either to gate plasticity (Miconi, Rawal, Clune, & Stanley, 2020; Soltoggio, Bullinaria, Mattiussi, Dürr, & Floreano, 2008), gate neural activations (Beaulieu et al., 2020) or alter high level behaviour (Xing, Zou, & Krichmar, 2020). The proposed mechanism in this work focuses on just one simple principle: modulatory signals alter the representations in each layer by gating the weighted sum of input of the standard neural component.

The primary contribution of this work is a neuromodulated policy network for meta-reinforcement learning for solving increasingly difficult problems. The modular approach of the design allows for the proposed layer to be used with other existing layers (such as standard fully connected layer, convolutional layer and so on) when stacking them to form a deep network. The experimental evidence in this work demonstrates that neuromodulation is beneficial to adapt network representations with more flexibility in comparison to standard networks. Experimental evaluations were conducted across high dimensional discrete and continuous control environments of increasing complexity using CAVIA and PEARL meta-RL algorithms. The results indicate that the neuromodulated networks show an increasing advantage as the problem complexity increases, while they perform comparably on simpler problems. The increased diversity of the representations from the neuromodulated policy network is examined and discussed. The open source implementation of the code can be found at: https://github.com/dlpbc/nm-metarl

## 2. Related work

**Meta-reinforcement learning.** This work builds on the existing meta learning frameworks (Bengio, Bengio, Cloutier, & Gecsei, 1992; Schmidhuber, Zhao, & Wiering, 1996; Schweighofer & Doya, 2003; Thrun & Pratt, 1998) in the domain of reinforcement learning. Recent studies in meta-reinforcement learning (meta-RL) can be largely classified into optimization and context-based methods. Optimization methods (Finn et al., 2017; Li, Zhou, Chen, & Li, 2017; Rothfuss, Lee, Clavera, Asfour, & Abbeel, 2019; Stadie et al., 2018) seek to learn good initial parameters of a model that can be adapted with a few gradient steps to a specific task. In contrast, context-based methods seek to adapt a model to a specific task based on few-shot experiences aggregated into context variables. The context can be derived via probabilistic methods (Liu, Raghunathan, Liang, & Finn, 2021; Rakelly et al., 2019), recurrent memory (Duan, Schulman, et al., 2016; Wang et al., 2016), recursive networks (Mishra et al., 2018) or the combination of probabilistic and memory (Humplik et al., 2019; Zintgraf et al., 2020). Hybrid methods (Gupta et al., 2018; Zintgraf et al., 2019) combine optimization and context-based methods whereby task specific context parameters are obtained via gradient updates.

**Neuromodulation.** Neuromodulation in biological brains is a process whereby a neuron alters or regulates the properties of other neurons in the brain (Marder, 2012). The altered properties can either be in the cellular activities or synaptic weights of the neurons. Well known biological neuromodulators include dopamine (DA), serotonin (5-HT), acetycholine (ACh), and noradrenaline (NA) (Avery & Krichmar, 2017; Bear, Connors, & Paradiso, 2020). Such neuromodulators were described in Doya (2002) within the reinforcement learning computation framework, with dopamine loosely mapped to the reward signal error (like TD error), serotonin representing discount factor, acetycholine representing learning rate and noradrenaline representing randomness in a policy's action distribution. Several studies have drawn inspiration from neuromodulation and applied it to gradient-based RL (Miconi et al., 2020; Xing et al., 2020) and neuroevolutionary RL (Soltoggio et al., 2008; Soltoggio, Durr, Mattiussi, & Floreano, 2007; Velez & Clune, 2017) for dynamic task settings. In broader machine learning, neuromodulation has been applied to goal-driven perception (Zou, Kolouri, Pilly, & Krichmar, 2020), and also in continual learning setting (Beaulieu et al., 2020) where it was combined with meta-learning to sequentially learn a number of classification tasks without catastrophic forgetting. The neuromodulators used in these studies have different designs or functions: plasticity gating (Miconi et al., 2020; Soltoggio et al., 2008), activation gating (Beaulieu et al., 2020), direct action modification in a policy (Xing et al., 2020).

## 3. Background

### 3.1. Problem formulation

In a meta-RL setting, tasks are sampled from a task distribution $p(\mathcal{T})$. Each task $\mathcal{T}_i$ is a Markov Decision Process (MDP), which is a tuple $M_i$=$\{\mathcal{S}, \mathcal{A}, q, r, q_0\}$ consisting of a state space $\mathcal{S}$, an action space $\mathcal{A}$, a state transition distribution $q(s_{t+1}|s_t, a_t)$, a reward function $r(s_t, a_t, s_{t+1})$, and an initial state distribution $q_0(s_0)$. When presented with a task $\mathcal{T}_i$, an agent (with a policy $\pi$) is required to quickly adapt to the task from few interactions. Therefore, the goal of the agent for each task is to maximize the expected reward in the shortest time possible:

$$\mathcal{J}(\pi) = \mathbf{E}_{q_0,q,\pi}\left[\sum_{t=0}^{H-1} \gamma^t r(s_t, a_t, s_{t+1})\right], \tag{1}$$

where $H$ is a finite horizon and $\gamma \in [0, 1]$ is the discount factor.

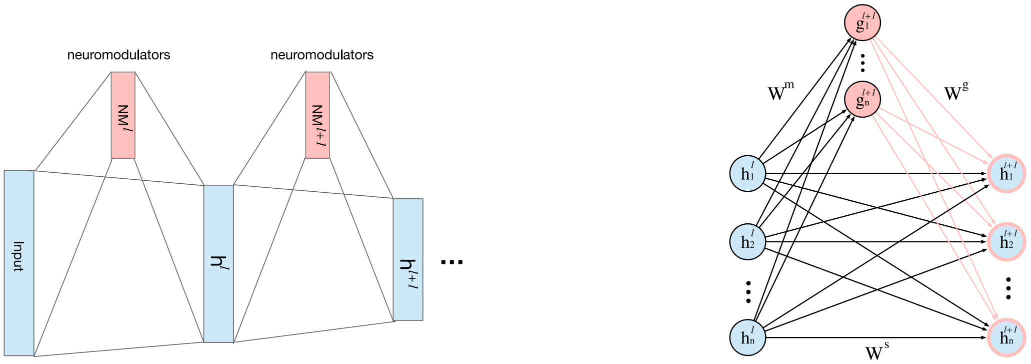
### 3.2. Context Adaptation via Meta-Learning (CAVIA)

The CAVIA meta-learning framework (Zintgraf et al., 2019) is an extension of the *model-agnostic meta-learning algorithm (MAML)* (Finn et al., 2017) that is interpretable and less prone to meta-overfitting. The key idea in CAVIA is the introduction of context parameters in a policy network. Therefore, the policy $\pi_{\theta,\phi}$ contains the standard network parameters $\theta$ and the context parameters $\phi$. During the adaptation phase for each task (the gradient updates in the inner loop), only the context parameters are updated, while the network parameters are updated during the outer loop gradient updates. There are different ways to provide the policy network with the context parameters. In Zintgraf et al. (2019), the parameters were concatenated to the input.

In the meta-RL framework, an agent is trained for a number of iterations. For each iteration, $N$ tasks represented as $\mathbf{T}$ are sampled from the task distribution $\mathcal{T}$. For each task $i$, a batch of trajectories $\tau_i^{train}$ is obtained using the policy $\pi_{\theta,\phi}$ with the context parameters set to an initial condition $\phi_0$. The obtained trajectories for task $i$ are used to perform a one step inner loop gradient update of the context parameters to new values $\phi_i$, shown in the equation below:

$$\phi_i = \phi_0 - \alpha \nabla_\phi \mathcal{J}_{\mathcal{T}_i}(\tau_i^{train}, \pi_{\theta,\phi_0}), \tag{2}$$

where $\mathcal{J}_{\mathcal{T}_i}(\tau_i, \pi_{\theta,\phi})$ is the objective function for task $i$. After the one step gradient update of the policy, another batch of trajectories $\tau_i^{test}$ is collected using the updated task specific policy $\pi_{\theta,\phi_i}$.



(a) Modular format of the proposed architecture

(b) A single neuromodulated fully connected layer

**Fig. 1.** Overview of the proposed computational framework. (a) Light blue boxers indicate layers of standard neurons and pink boxes are layers of modulatory neurons. (b) Illustration of a single layer of the proposed architecture. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

After completing the above procedure for all tasks sampled from $\mathcal{T}$, a meta gradient step (also referred to as the outer loop update) is performed, updating $\theta$ to maximize the average performance of the policy across the task batch.

$$\theta = \theta - \beta\nabla_\theta \frac{1}{N}\sum_{\tau_i\in\mathbf{T}} \mathcal{J}_{\mathcal{T}_i}(\tau_i^{test}, \pi_{\theta,\phi_i}). \tag{3}$$

### 3.3. Probabilistic Embeddings for Actor–Critic Meta-RL (PEARL)

PEARL (Rakelly et al., 2019) is an off-policy meta-RL algorithm that is based on the soft actor–critic architecture (Haarnoja, Zhou, Abbeel, & Levine, 2018). The algorithm derives the context of the task to which an agent is exposed through probabilistic sampling. Given a task, the agent maintains a prior belief of the task, and as the agent interacts with the environment, it updates the posterior distribution with the goal of identifying the specific task context. The context variables $\mathbf{z}$ are concatenated to the input of the actor and critic neural components of the setup. To estimate this posterior $p(\mathbf{z}|\mathbf{c})$, an additional neural component called an inference network $q_\phi(\mathbf{z}|\mathbf{c})$ is trained using the trajectories $\mathbf{c}$ collected for tasks sampled from the task distribution $\mathcal{T}$. The objective function for the actor, critic and inference neural components are described below,

$$\mathcal{L}_{actor} = \mathbb{E}_{\substack{\mathbf{s}\sim\mathcal{B},\mathbf{a}\sim\pi_\theta\\ \mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{c})}} \left[ D_{\mathrm{KL}}\left(\pi_\theta(\mathbf{a}|\mathbf{s},\bar{\mathbf{z}}) \middle\| \frac{\exp(Q_\theta(\mathbf{s},\mathbf{a},\bar{\mathbf{z}}))}{\mathcal{Z}_\theta(\mathbf{s})}\right)\right] \tag{4}$$

$$\mathcal{L}_{critic} = \mathbb{E}_{\substack{(\mathbf{s},\mathbf{a},r,\mathbf{s}')\sim\mathcal{B}\\ \mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{c})}} [Q_\theta(\mathbf{s},\mathbf{a},\mathbf{z}) - (r+\bar{V}(\mathbf{s}',\bar{\mathbf{z}}))]^2 \tag{5}$$

$$\mathcal{L}_{inference} = \mathbb{E}_{\mathcal{T}}[\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{c}^{\mathcal{T}})}[\mathcal{L}_{critic} + \beta D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{c}^{\mathcal{T}}) \parallel p(\mathbf{z}))]]xx \tag{6}$$

where $\bar{V}$ is a target network and $\bar{\mathbf{z}}$ means that gradients are not being computed through it, $p(\mathbf{z})$ is a unit Gaussian prior over $Z$, $\mathcal{B}$ is the replay buffer and $\beta$ is a weighting hyper-parameter.

## 4. Neuromodulated network

This section introduces the extension of the policy network with neuromodulation. A graphical representation of the network is shown in Fig. 1(a). The neuromodulated policy network is a stack of neuromodulated fully connected layers.

### 4.1. Computational framework

A neuromodulated fully connected layer contains two neural components: standard neurons and neuromodulators (see Fig. 1(b)). The standard neurons serve as the output of the layer (i.e., the layer's representations) and they are connected to the preceding layer via standard fully connected weights $W^s$. The neuromodulators serve as a means to alter the output of the standard neurons. They receive input via standard fully connected weights $W^g$ from the preceding layer in order to generate their neural activity, which is then projected to the standard neurons via another set of fully connected weights $W^m$. The function of the projected neuromodulatory activity defines the representation altering mechanism. For example, it could gate the plasticity of $W^s$, gate neural activation of $\mathbf{h}$ or do something else based on the designer's specification. While different types of neuromodulators can be used (Doya, 2002), in this particular work, we employ an activity-gating neuromodulator. Such neuromodulator multiplies the activity of the target (standard) neurons before a non-linearity is applied to the layer. Formally, the structure can be described with three parameter matrices: $W^s$ defines weights connecting the input to the standard neurons, $W^g$ defines weights connecting the input to the neuromodulators and $W^m$ defines weights connecting the neuromodulators to the standard neurons. The step-wise computation of a forward pass through the neuromodulatory structure is given below:

$$\mathbf{h}^s = W^s \cdot \mathbf{x} \tag{7}$$

$$\mathbf{g} = ReLU(W^g \cdot \mathbf{x}) \tag{8}$$

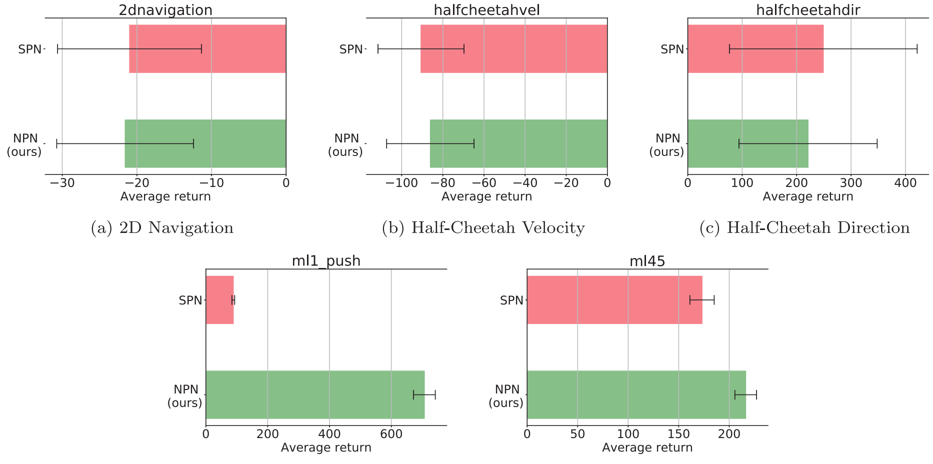
$$\mathbf{h}^m = tanh(W^m \cdot \mathbf{g}) \tag{9}$$

$$\mathbf{h} = ReLU(\mathbf{h}^s \otimes \mathbf{h}^m) \tag{10}$$

where $\mathbf{x}$ is the layer's input, $\mathbf{h}^s$ is the weighted sum of input of the standard neurons, $\mathbf{g}$ is activity of the neuromodulators derived from the weighted sum of input, $\mathbf{h}^m$ is the neuromodulatory activity projected onto the standard neurons, and $\mathbf{h}$ is the output of the layer. The key modulating process takes place in the element-wise multiplication of the $\mathbf{h}^s$ and $\mathbf{h}^m$.

The *tanh* non-linearity is employed to enable positive and negative neuromodulatory signals, and thus gives the network the ability to affect both the magnitude and the sign of target activation values. When *ReLU* is used as the non-linearity for the layer's output $h$, $\mathbf{h}^m$ has the intrinsic ability to dynamically turn on or off certain output in $\mathbf{h}$.

A simpler version of the proposed model can be achieved by only considering the sign, and not the magnitude, of the



(a) 2D Navigation (b) Half-Cheetah Velocity (c) Half-Cheetah Direction

(d) Meta-World ML1 (e) Meta-World ML45

**Fig. 2.** Adaptation performance across tasks of the standard policy network (SPN) and the neuromodulated policy network (NPN) in continuous control environment using CAVIA meta-RL framework. Across three seed runs, the performance was measured based on average return from the rewards acquired during evaluation.

neuromodulatory signal, using the following variation of Eq. (10):

$$\mathbf{h} = ReLU(\mathbf{h}^s \otimes \mathbf{sign}(\mathbf{h}^m)) \tag{11}$$

This variation is shown to be suited for discrete control problems.

## 5. Results and analysis

In this section, the results of the neuromodulated policy network evaluations across high dimensional discrete and continuous control environments with varying levels of complexity are presented. The continuous control environments are the simple 2D navigation, the half-cheetah direction (Finn et al., 2017) and velocity (Finn et al., 2017) Mujoco (Todorov, Erez, & Tassa, 2012) based environments and the meta-world ML1 and ML45 environments (Yu et al., 2020). The discrete action environment is a graph navigation environment that supports configurable levels of complexity called the CT-graph (Ben-Iwhiwhu et al., 2020; Ladosz et al., 2021; Soltoggio, Ladosz, Ben-Iwhiwhu, & Dick, 2021). The experimental setup focused on investigating the beneficial effect of the proposed neuromodulatory mechanism when augmenting existing meta-RL frameworks (i.e., neuromodulation as complementary tool to meta-RL rather than competing). To this end, using CAVIA meta-RL method (Zintgraf et al., 2019), a standard policy network (SPN) is compared against the neuromodulated policy network (NPN) across the aforementioned environments. Similarly, SPN is compared against NPN using PEARL (Rakelly et al., 2019) method only in the continuous control environments because the soft actor–critic architecture employed by PEARL is designed for continuous control. We present the analysis of the learned dynamic representations from a standard and a neuromodulated network in Section 5.2. Finally, the policy networks were evaluated in a RGB autonomous vehicle navigation domain in the CARLA driving simulator using CAVIA and the results and discussions are presented in Appendix D.

### 5.1. Performance

The experimental setup for CAVIA and PEARL as in Rakelly et al. (2019) and Zintgraf et al. (2019) were followed. For PEARL, neuromodulation was applied only to the actor neural component. The details of the experimental setup and hyper-parameters are presented in Appendix A. The performance reported is the meta-testing results of the agents in the evaluation environments after meta-training has been completed (Figs. 2, 3, 4 and 5). During meta-testing in CAVIA, the policy networks were fine-tuned for 4 inner loop gradient steps. Lastly, depending on the evaluation environment, the metric used to judge evaluation performance was either return[1] or success rate.[2]

#### 5.1.1. 2D navigation environment

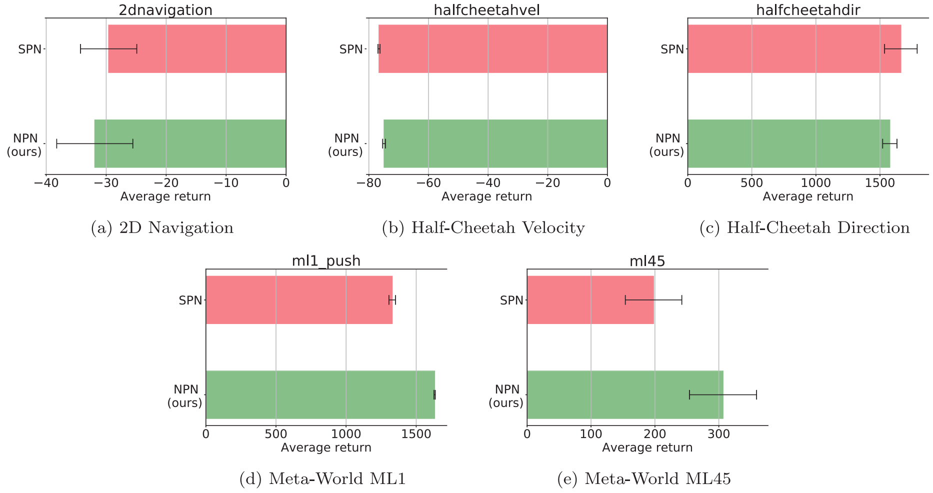
The first simulations are in the 2D point navigation experiment introduced in Finn et al. (2017). An agent is tasked with navigating to a randomly sampled goal position from a start position. A goal position is sampled from the interval $[-0.5, 0.5]$. The reward function is the negative squared distance between the current agent position and the goal. An observation is the agent's current 2D position while the actions are velocity commands clipped at $[-0.1, 0.1]$. The result of the meta-testing performance evaluation comparing both the standard policy network and neuromodulated policy network is presented in Fig. 2(a) for CAVIA and Fig. 3(a) for PEARL. The result shows that both policy networks had a relative good performance. Such optimal performance is expected from both policies as the environment is simple and the dynamic representations required for each task are not very distinct.

#### 5.1.2. Half-cheetah

The half-cheetah is an environment based on the MuJoCo simulator (Todorov et al., 2012) that requires an agent to learn continuous control locomotion. We employ two standard meta-RL benchmarks using the environment as proposed in Finn et al. (2017); (i) the direction task that requires the cheetah agent to run either forward or backward and (ii) the velocity task that requires the agent to run at a certain velocity sampled from a distribution of velocities. Although challenging (due to their high dimensional nature) in comparison to the 2D navigation task, these benchmark are still simplistic as the direction benchmark contains only two unique tasks and the velocity benchmark samples small range of velocities ($[0, 2.0)$ or $[0, 3.0)$). Therefore, the optimal policies across tasks in these benchmarks possess similar representations. The results of the experiments for both benchmarks are presented in Figs. 2(c) and 2(b) for CAVIA, and

[1] Return is a standard metric in RL that is computed as the sum of cumulative reward acquired by the agent.

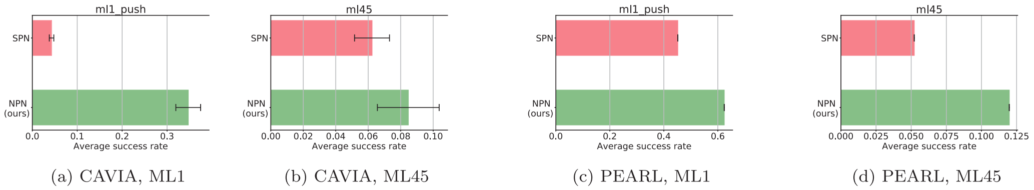
[2] Success rate is a metric introduced in Meta-World, having a value of 1 if the agent has solved or is close to solving the task (i.e., if the distance between the current position of the task relevant object and goal position is smaller than some $\epsilon$ value, otherwise, it is set to 0.



(a) 2D Navigation (b) Half-Cheetah Velocity (c) Half-Cheetah Direction

(d) Meta-World ML1 (e) Meta-World ML45

**Fig. 3.** Adaptation performance across tasks of the standard policy network (SPN) and the neuromodulated policy network (NPN) in continuous control environment using PEARL meta-RL framework. Across three seed runs, the performance was measured based on average return from the rewards acquired during evaluation.



(a) CAVIA, ML1 (b) CAVIA, ML45 (c) PEARL, ML1 (d) PEARL, ML45

**Fig. 4.** Adaptation performance (across tasks, based on success rate metric) of the standard policy network (SPN) and the neuromodulated policy network (NPN) in CAVIA and PEARL. Across three seed runs, the performance was measured based on the success rate metric from the evaluation.

Figs. 3(c) and 3(b) for PEARL. Unsurprisingly, the results show comparable level of performance between the standard policy network and the neuromodulated policy network across CAVIA and PEARL. These benchmarks are of medium complexity and the optimal policy for each task is similar to others.

### 5.1.3. Meta-world

The neuromodulated policy network was evaluated in a complex high-dimensional continuous control environment called meta-world (Yu et al., 2020). In meta-world, an agent is required to manipulate a robotic arm to solve a wide range of tasks (e.g. pushing an object, pick and place objects, opening a door and more). Two instances of the benchmark ML1 and ML45 were employed. In ML1 instance, the robot is required to solve a single task that contains several parametric variations (e.g. push an object to different goal locations). The parametric variations of the selected task are used as the meta-train and meta-test tasks. ML45 is a more complex instance that contains a wide variety of tasks (each task with parametric variations). It consists of 45 distinct meta-train tasks and 5 distinct meta-test tasks. The standard policy network and neuromodulated policy network were evaluated in ML1 and ML45 instances using CAVIA and PEARL. The results[3] are presented in Figs. 2(d) and 2(e) for CAVIA, and Figs. 3(d) and 3(e) for PEARL. In these complex benchmarks, the results show that the neuromodulated policy network outperforms the standard policy network in both CAVIA and PEARL, highlighting the advantage neuromodulation offers in complex problem setting. In addition to judging the performance based on reward, results are also presented using the success rate metric (introduced in Yu et al. (2020) as a metric judge whether or not an agent is able to solve a task) in Fig. 4. The results again show that the neuromodulated policy network achieved significantly higher average success rate both in CAVIA and PEARL in comparison to the standard policy network.

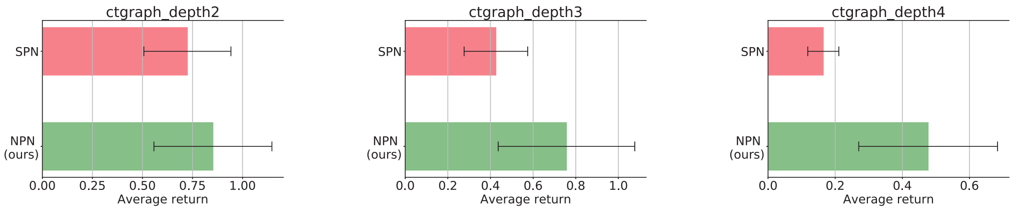
### 5.1.4. Configurable Tree graph (CT-graph) environment

The CT-graph is a sparse reward discrete control graph environment with increasing complexity that is specified via parameters such as branch $b$ and depth $d$. An environment instance consists of a set of states including a start state and a number of end states. An agent is tasked with navigating to a randomly sampled end state from the start state. See Appendix B for more details about the CT-graph. The three CT-graph instances used in this work were setup with varying depth parameter: with increasing depth, the sequence of actions grows linearly, but the search space for the policy network grows exponentially. The simplest instance has $d$ set to 2 (CT-graph depth2), and the next has $d$ set to 3 (CT-graph depth3) and the most complex instance has $d$ set to 4 (CT-graph depth4). The meta-testing results are presented in Fig. 5. The results show a significant difference in performance between standard and neuromodulated policy network. The optimal adaption performance from the neuromodulated policy network stems from the rich dynamic representations needed for adaptation as discussed in Section 5.2.

## 5.2. Analysis

In this section, we conduct analysis on the learnt representations of the standard and neuromodulated policy networks for tasks in the 2D Navigation and CT-graph environments. The policy networks trained using CAVIA was chosen for the analysis as the single neural component in CAVIA (i.e. the policy network)

[3] The experiments were conducted using the updated Meta-World (i.e., v2) environment containing the updated reward function.

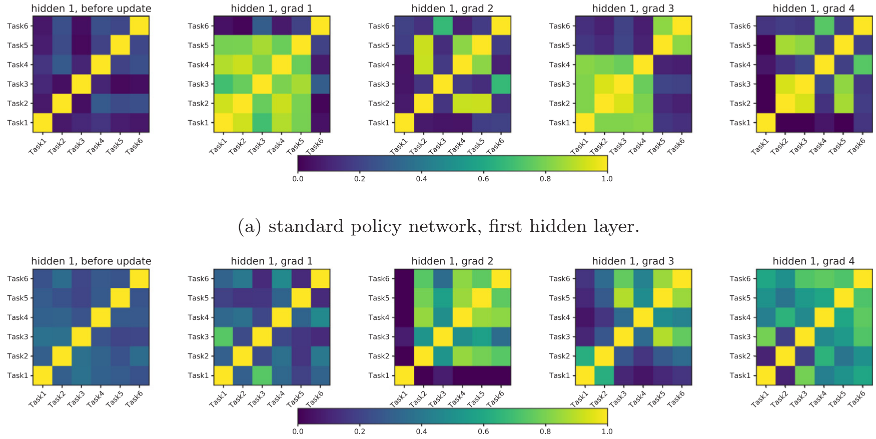


(a) CT-graph ($b = 2, d = 2$)

(b) CT-graph ($b = 2, d = 3$)

(c) CT-graph ($b = 2, d = 4$)

**Fig. 5.** Adaptation performance across tasks of the standard policy network (SPN) and the neuromodulated policy network (NPN) in three discrete control environments using CAVIA meta-RL framework. Across three seed runs, the performance was measured based on the success rate metric from the evaluation.



(a) standard policy network, first hidden layer.

(b) neuromodulated policy network, first hidden layer.

**Fig. 6.** Representation similarities between tasks in the 2D Navigation environment.

makes it easier to analyse in comparison to PEARL which contain multiple neural components. Furthermore, PEARL experiments were conducted only in continuous control environments (similar to the original paper), whereas CAVIA experiments covered both discrete and continuous control environments. Hence, analysis in CAVIA allowed for more coverage across benchmarks.

To measure representation similarity across task, we employ the use of the centred kernel alignment (CKA) (Kornblith, Norouzi, Lee, & Hinton, 2019) similarity index, comparing per layer representations of both standard and neuromodulated policy networks across different tasks. There exist several similarity index measures such as canonical correlation analysis (CCA) (Morcos, Raghu, & Bengio, 2018), representation similarity analysis (RSA) (Kriegeskorte, Mur, & Bandettini, 2008), Hilbert–Schmidt Independence Criterion (HSIC) (Gretton, Bousquet, Smola, & Schölkopf, 2005) and more.

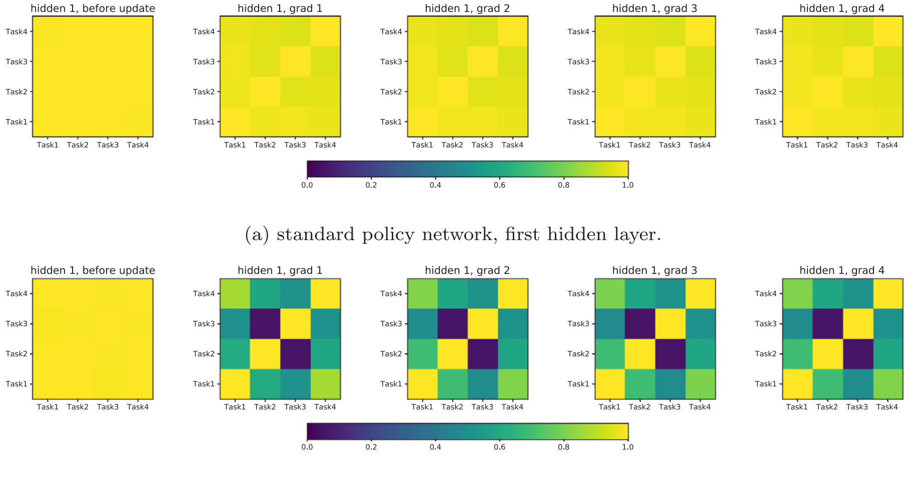
The principle behind CKA is the generation of a similarity measure between two representations by comparing the similarity structure of both representations. Each similarity structure is produced from the measure of similarity between pairwise examples or data points in a representation. Furthermore, CKA is a generalized extension of HSIC, with the inclusion of normalization which introduces the property of isotropic scaling invariance. Describing the formulation of the CKA is outside the scope of this work and we refer readers to the original paper for detailed theoretical formulations. While the RSA similarity index measure employed in Goerttler and Obermayer (2021) is a valid alternative, we chose the CKA due to its robustness to random initialization and enabling comparison within layers of the same network and comparison across networks. Furthermore, CKA has been employed previously in meta-RL setting, e.g., demonstrated in Raghu, Raghu, Bengio, and Vinyals (2020).

### *5.2.1. Analysis: Representation similarities for standard and neuromodulated policy networks across tasks*

The per layer output representation similarities between tasks were plotted as heat maps in Figs. 6 and 7. Each heatmap in a row (for example 6(a)) depicts the similarity before or after few steps of gradient updates to the layer. Before any gradient updates, the representations are similar between tasks in the figure. After gradient updates, some dissimilarities between tasks begin to emerge. Additional analysis plots are presented in Appendix F.

**2D Navigation**. For the simple 2D Navigation environment, the plots for the first hidden layer of the standard policy network shown in Fig. 6(a) depicts good dissimilarity between tasks, thus highlighting the fact that the learnt representations are sufficient to produce distinct task behaviours. The same is true as well for the first hidden layer of the neuromodulated policy network (see Fig. 6(b)). This further justifies why both policies obtained roughly comparable performance in this environment. The simplicity of the problem enables task distinct representations to be obtained easily. Appendix F.1 contains the plots of the representation similarity for the second hidden layer of both policy networks.

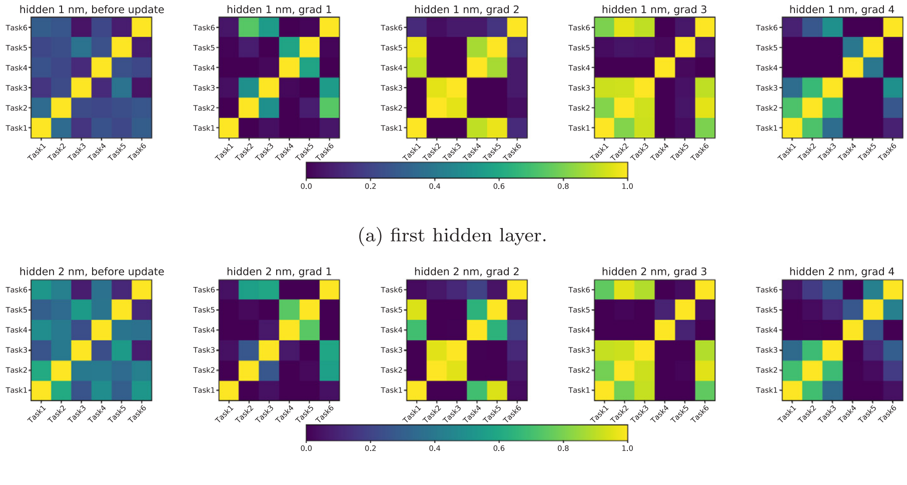
**CT-graph**. In Figs. 7(a) and 7(b), we compare the representation similarity of the first hidden layer of the standard and neuromodulated policy networks in the CT-graph depth2 environment. We see that representations of the neuromodulated policy are more dissimilar between the tasks than those of the standard policy. Due to the complexity of the environment, the task specific representations required to solve each task are distinct from one another. Therefore, adaptation by fine-tuning the representations of a base network via few gradient steps of parameters update would require a significant jump in the solution space. Standard policy network struggles to enable such jump in the solution space. However, by incorporating neuromodulators



(a) standard policy network, first hidden layer.

(b) neuromodulated policy network, first hidden layer.

**Fig. 7.** Representation similarities between tasks in the CT-graph depth2 environment.



(a) first hidden layer.

(b) second hidden layer.

**Fig. 8.** Representation similarities of neuromodulatory activities $h^m$ between tasks in the 2D Navigation environment across the first hidden layer (a), and the second hidden layer (b) of the network. Non-uniformity across heatmap plots shows dissimilar representations emerge from neuromodulatory activity which helped the neuromodulated policy network to solve tasks in complex problem benchmarks.

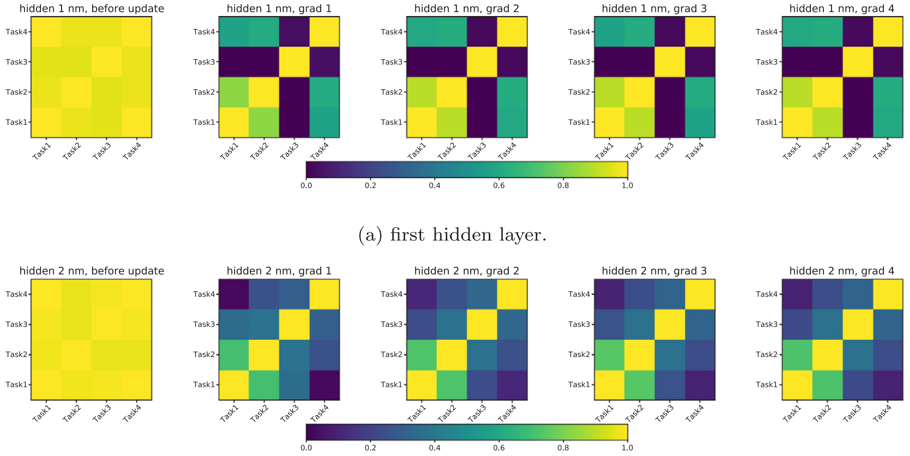
that dynamically alters the representations, such jump becomes possible. Appendix F.1 contains the plots of the representation similarity for the second hidden layer of both policy networks.

#### 5.2.2. Analysis: Representation similarities of the neuromodulatory units across tasks

Now we ask ourselves where the representational diversity (dissimilarity in representations across tasks) comes from. Is the neuromodulatory layer effectively contributing to rich representations as Fig. 7(b) appeared to suggest? The analysis we present here shows task representation similarities measured more specifically across the neuromodulatory layers of the proposed architecture. From Fig. 7(b), it appears that such dissimilarity is enhanced by the neuromodulatory activities in the NPN. Again the centred kernel alignment (CKA) was employed and we compare the neuromodulatory activities per layer across different tasks. Figs. 8–10 present the heat map plots for the 2D navigation, CT-graph depth 2 and ML45 environments (additional plots for other environment are presented in Appendix F.4)). The non-uniformity in the heatmap plots, in contrast to those of Fig. 7(a), indicates that those layers encode diverse or dissimilar representations across task. We can therefore conclude that the neuromodulatory activities, when projected onto a layer's standard neurons, produce the desired dissimilar representations across tasks.

### 5.3. Control experiments: larger SPN, equalling the number of parameters of the NPN

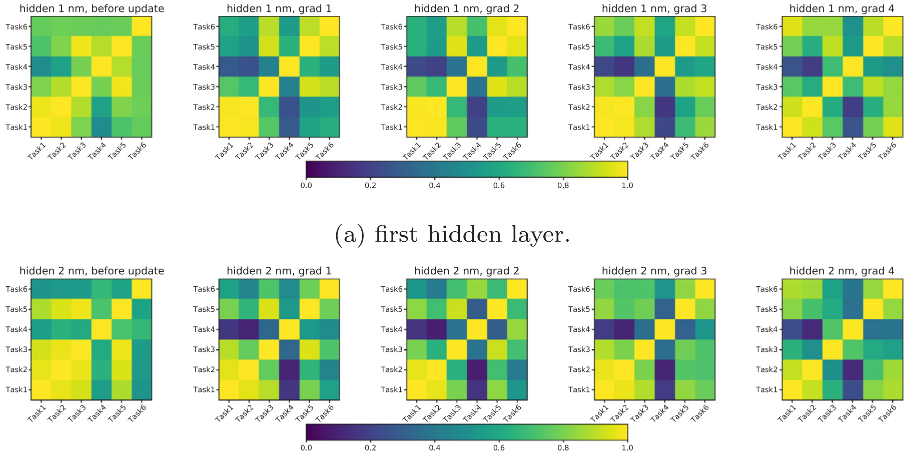
Since the inclusion of neuromodulators increases the number of parameters in a neuromodulated policy network, a set of control experiments were conducted in which the number of parameters in a standard policy network was configured to approximately match that of a neuromodulated policy network. This was achieved by increasing the size of each hidden layer of the standard policy network (called SPN larger width) in one experiment, and increasing the depth or number of hidden layers by 1, i.e., an additional layer (SPN larger depth) in another experiment. Using CAVIA, experiments were conducted in the CT-graph depth4 and the ML45 meta-world environments, comparing the standard policy network (i.e., the original size), its larger variants and a neuromodulated policy network. The results are presented in Fig. 11. We observe from the results that the increase in the size of the policy network does not lead to match of the performance of the neuromodulated policy network.



(a) first hidden layer.

(b) second hidden layer.

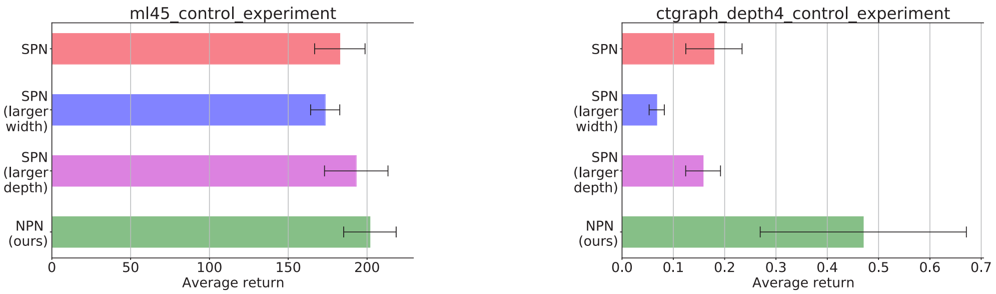
**Fig. 9.** Representation similarities of neuromodulatory activities $h^m$ between tasks in the CT-graph depth2 environment across the first hidden layer (a), and the second hidden layer (b) of the network. Non-uniformity across heatmap plots shows dissimilar representations emerge from neuromodulatory activity which helped the neuromodulated policy network to solve tasks in complex problem benchmarks.



(a) first hidden layer.

(b) second hidden layer.

**Fig. 10.** Representation similarities of neuromodulatory activities $h^m$ between tasks in the Meta-World ML45 environment across the first hidden layer (a), and the second hidden layer (b) of the network. Non-uniformity across heatmap plots show dissimilar representations emerge from neuromodulatory activity which helped the neuromodulated policy network to solve tasks in complex problem benchmarks.



(a) ML45

(b) CT-graph depth4

**Fig. 11.** Control experiments. Adaptation performance of standard policy network (SPN), a larger SPN variant and neuromodulated policy network (NPN) using CAVIA. Note, the number of parameters in *SPN (larger)* approximately matches that of the NPN in each environment.

## 6. Discussions

**Neuromodulation and gated recurrent networks:** The neuromodulatory gating mechanism introduced in this work is reminiscent of the gating in recurrent/memory networks (LSTMs (Hochreiter & Schmidhuber, 1997) and GRUs (Cho, Van Merriënboer, Bahdanau, & Bengio, 2014)). In this respect (with the observation of improved performance as a consequence of neuromodulatory gating in this work), the noteworthy performance demonstrated by meta-RL memory approaches (Duan, Schulman, et al., 2016; Wang et al., 2016) could also be a consequence of such gating mechanisms.[4] Nonetheless, the present study aims to highlight the advantage of a simpler form of gating (i.e., neuromodulatory gating) on a MLP feedforward network, and thus could help to pinpoint the advantage of such dynamics in isolation. Furthermore, the advantage of our approach over gated recurrent variants is somewhat similar to the advantages derived from decoupling attention mechanism from recurrent models (where it was originally introduced) and applying it to MLP networks (i.e., Transformer models) (Vaswani et al., 2017). By decoupling neuromodulatory (gating) mechanism from recurrent models and applying it to MLP models (as in our work), the advantages of faster training and better parallelization were achieved while maintaining the benefit of neuromodulatory gating. Therefore, our proposed approach is faster to train and more parallelizable in comparison to memory variants, while maintaining the advantages that neuromodulatory gating offers. Memory based approaches will still be required for problems where memory is advantageous such as sequential data processing and POMDPs.

**Task similarity measure and robust benchmarks:** Increasing task complexity was presented in this work by moving from simple 2D point navigation environment to half-cheetah locomotion and then to the complex robotic arm setup of the Meta-World environment. Furthermore, exploiting the benefits of configurable parameters in the CT-graph environment, we were able to control the complexity in the environment. Overall, task complexity was viewed through the perspective of task similarity (i.e., environments with dissimilar task were viewed as more complex and vice versa). Despite these efforts, a precise measure of task complexity and similarity was not clearly outlined in this work and this is widely the case in meta-RL literatures. There is a need for the development of precise metrics for measuring task similarity and complexity in the field. The CT-graph with its configurable parameters allow for tasks to be mathematically defined, which is a first step towards alleviating this issue. However, a separate future research investigation would be necessary to develop explicit metrics that can be incorporated into meta-RL benchmarks.

We hypothesize that such a task similarity metric should be able to capture the precise change points in a task relative to other tasks. For example, a useful metric could be one that capture task change either as a function of change in reward, or state space, or transition function, or a combination of these factors. Most benchmarks in meta-RL have been focused on task change as reward function change. However, a more robust benchmark could include the aforementioned change points in order to further control the complexity. The CT-graph, Meta-world, and the recently developed Alchemy (Wang et al., 2021) environment are examples of benchmarks with early stage work in this direction, albeit implicitly. Therefore, the development of a precise measure of task similarity and complexity, as well as robust benchmarks with configurable change points (i.e., reward, state/input, and transition) would be highly beneficial to the meta-RL field.

## 7. Conclusion and future work

This paper introduced an architectural extension of the standard meta-RL policy networks to include a neuromodulatory mechanism, investigating the beneficial effect of neuromodulation when augmenting existing meta-RL frameworks (i.e., neuromodulation as complementary tool to meta-RL rather than competing). The aim is to implement richer dynamic representations and facilitate rapid task adaptation in increasingly complex problems. The effectiveness of the proposed approach was evaluated in meta-RL setting using CAVIA and PEARL algorithms. In the experimental setup across environments of increasing complexity, the neuromodulated policy network significantly outperformed the standard policy network in complex problems while showcasing comparable performance in simpler problems. The results highlight the usefulness of neuromodulators to enable fast adaptation via rich dynamic representations in meta-RL problems. The architectural extension, although simple, presents a general framework for extending meta-RL policy networks with neuromodulators that expand their ability to encode different policies. The projected neuromodulatory activity can be designed to perform other functions apart from the one introduced in this work e.g., gating plasticity of weights, or including different neuromodulators in the same layer. The neuromodulatory extension could also be tested with a recurrent meta-RL policy, with the goal of enhancing the memory dynamics of the policy. Our analysis indicates that this framework is most suited to problems that require rapid change in optimal representations across tasks, while its advantage is reduced when tasks can be solved using similar representations.

## Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Praveen K. Pilly and Nicholas A. Ketz are affiliated with HRL Laboratories LLC, both having pending patent applications in lifelong learning. Other authors declare no conflict of interest. **Funding.** This material is based upon work supported by the United States Air Force Research Laboratory (AFRL) and Defense Advanced Research Projects Agency (DARPA) under Contract No. FA8750-18-C0103. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force Research Laboratory (AFRL) and Defense Advanced Research Projects Agency (DARPA).

## Acknowledgements

This material is based upon work supported by the United States Air Force Research Laboratory (AFRL), United States of America and Defense Advanced Research Projects Agency (DARPA), United States of America under Contract No. FA8750-18-C0103. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force Research Laboratory (AFRL) and Defense Advanced Research Projects Agency (DARPA). The authors would like to thank Jeffrey Krichmar for useful discussion. Lastly, the authors thank the anonymous reviewers for their valuable feedback which helped to improve the paper.

[4] Although not the focus of this work, we ran an experiment using $RL^2$ (a memory based meta-RL method) in the ML45 environment and achieved an average meta-test success rate performance of 10%, which is comparable to the results obtained using neuromodulatory gating mechanism. See Appendix E for discussions about performance in meta-world, in relation to the performance reported in the original meta-world paper.

## Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.neunet.2022.04.003.

## References

Avery, M. C., & Krichmar, J. L. (2017). Neuromodulatory systems and their interactions: a review of models, theories, and experiments. *Frontiers in Neural Circuits*, *11*, 108.

Bear, M., Connors, B., & Paradiso, M. A. (2020). *Neuroscience: exploring the brain*. Jones & Bartlett Learning, LLC.

Beaulieu, S., Frati, L., Miconi, T., Lehman, J., Stanley, K. O., Clune, J., et al. (2020). Learning to continually learn. arXiv preprint arXiv:2002.09571.

Ben-Iwhiwhu, E., Ladosz, P., Dick, J., Chen, W.-H., Pilly, P., & Soltoggio, A. (2020). Evolving inborn knowledge for fast adaptation in dynamic POMDP problems. In *Proceedings of the 2020 genetic and evolutionary computation conference* (pp. 280–288).

Bengio, S., Bengio, Y., Cloutier, J., & Gecsei, J. (1992). On the optimization of a synaptic learning rule. In *Preprints conf. optimality in artificial and biological neural networks* (pp. 6–8). Univ. of Texas.

Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259.

Doya, K. (2002). Metalearning and neuromodulation. *Neural Networks*, *15*(4–6), 495–506.

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

Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th international conference on machine learning-vol. 70* (pp. 1126–1135). JMLR. org.

Goerttler, T., & Obermayer, K. (2021). Exploring the similarity of representations in model-agnostic meta-learning. In *Learning to learn-workshop at ICLR 2021*.

Gretton, A., Bousquet, O., Smola, A., & Schölkopf, B. (2005). Measuring statistical dependence with Hilbert-Schmidt norms. In *International conference on algorithmic learning theory* (pp. 63–77). Springer.

Gupta, A., Mendonca, R., Liu, Y., Abbeel, P., & Levine, S. (2018). Meta-reinforcement learning of structured exploration strategies. In *Proceedings of the 32nd international conference on neural information processing systems* (pp. 5307–5316).

Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning* (pp. 1861–1870). PMLR.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Humplik, J., Galashov, A., Hasenclever, L., Ortega, P. A., Teh, Y. W., & Heess, N. (2019). Meta reinforcement learning as task inference. arXiv preprint arXiv:1905.06424.

Kornblith, S., Norouzi, M., Lee, H., & Hinton, G. (2019). Similarity of neural network representations revisited. In *International conference on machine learning* (pp. 3519–3529). PMLR.

Kriegeskorte, N., Mur, M., & Bandettini, P. A. (2008). Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, *2*, 4.

Ladosz, P., Ben-Iwhiwhu, E., Dick, J., Ketz, N., Kolouri, S., Krichmar, J. L., et al. (2021). Deep reinforcement learning with modulated hebbian plus Q-network architecture. *IEEE Transactions on Neural Networks and Learning Systems*.

Li, Z., Zhou, F., Chen, F., & Li, H. (2017). Meta-sgd: Learning to learn quickly for few-shot learning. arXiv preprint arXiv:1707.09835.

Liu, E. Z., Raghunathan, A., Liang, P., & Finn, C. (2021). Decoupling exploration and exploitation for meta-reinforcement learning without sacrifices. In *International conference on machine learning* (pp. 6925–6935). PMLR.

Marder, E. (2012). Neuromodulation of neuronal circuits: back to the future. *Neuron*, *76*(1), 1–11.

Miconi, T., Rawal, A., Clune, J., & Stanley, K. O. (2020). Backpropamine: training self-modifying neural networks with differentiable neuromodulated plasticity. arXiv preprint arXiv:2002.10585.

Mishra, N., Rohaninejad, M., Chen, X., & Abbeel, P. (2018). A simple neural attentive meta-learner. In *International conference on learning representations*. URL https://openreview.net/forum?id=B1DmUzWAW.

Morcos, A., Raghu, M., & Bengio, S. (2018). Insights on representational similarity in neural networks with canonical correlation. *Advances in Neural Information Processing Systems*, *31*.

Raghu, A., Raghu, M., Bengio, S., & Vinyals, O. (2020). Rapid learning or feature reuse? Towards understanding the effectiveness of MAML. In *International conference on learning representations*. URL https://openreview.net/forum?id=rkgMkCEtPB.

Rakelly, K., Zhou, A., Finn, C., Levine, S., & Quillen, D. (2019). Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International conference on machine learning* (pp. 5331–5340).

Rothfuss, J., Lee, D., Clavera, I., Asfour, T., & Abbeel, P. (2019). ProMP: Proximal meta-policy search. In *International conference on learning representations*. URL https://openreview.net/forum?id=SkxXCi0qFX.

Schmidhuber, J., Zhao, J., & Wiering, M. (1996). Simple principles of metalearning. *Technical Report IDSIA*, *69*, 1–23.

Schweighofer, N., & Doya, K. (2003). Meta-learning in reinforcement learning. *Neural Networks*, *16*(1), 5–9.

Soltoggio, A., Bullinaria, J. A., Mattiussi, C., Dürr, P., & Floreano, D. (2008). Evolutionary advantages of neuromodulated plasticity in dynamic, reward-based scenarios. In *Proceedings of the 11th international conference on artificial life* (CONF), (pp. 569–576). MIT Press.

Soltoggio, A., Durr, P., Mattiussi, C., & Floreano, D. (2007). Evolving neuromodulatory topologies for reinforcement learning-like problems. In *2007 IEEE Congress on evolutionary computation* (pp. 2471–2478). IEEE.

Soltoggio, A., Ladosz, P., Ben-Iwhiwhu, E., & Dick, J. (2021). CT-graph environments - lifelong learning machines (L2M). GitHub Repository https://github.com/soltoggio/CT-graph.

Stadie, B. C., Yang, G., Houthooft, R., Chen, X., Duan, Y., Wu, Y., et al. (2018). Some considerations on learning to explore via meta-reinforcement learning. arXiv preprint arXiv:1803.01118.

Thrun, S., & Pratt, L. (1998). Learning to learn: Introduction and overview. In *Learning to learn* (pp. 3–17). Springer.

Todorov, E., Erez, T., & Tassa, Y. (2012). Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International conference on intelligent robots and systems* (pp. 5026–5033). IEEE.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Velez, R., & Clune, J. (2017). Diffusion-based neuromodulation can eliminate catastrophic forgetting in simple neural networks. *PLoS One*, *12*(11), Article e0187736.

Wang, J. X., King, M., Porcel, N., Kurth-Nelson, Z., Zhu, T., Deck, C., et al. (2021). Alchemy: A structured task distribution for meta-reinforcement learning. arXiv preprint arXiv:2102.02926.

Wang, J. X., Kurth-Nelson, Z., Tirumala, D., Soyer, H., Leibo, J. Z., Munos, R., et al. (2016). Learning to reinforcement learn, 2016. arXiv preprint arXiv:1611.05763.

Xing, J., Zou, X., & Krichmar, J. L. (2020). Neuromodulated patience for robot and self-driving vehicle navigation. In *2020 International joint conference on neural networks* (pp. 1–8). IEEE.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., et al. (2020). Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning* (pp. 1094–1100). PMLR.

Zintgraf, L., Shiarli, K., Kurin, V., Hofmann, K., & Whiteson, S. (2019). Fast context adaptation via meta-learning. In *International conference on machine learning* (pp. 7693–7702).

Zintgraf, L., Shiarlis, K., Igl, M., Schulze, S., Gal, Y., Hofmann, K., et al. (2020). VariBAD: A very good method for Bayes-adaptive deep RL via meta-learning. In *International conference on learning representations*.

Zou, X., Kolouri, S., Pilly, P. K., & Krichmar, J. L. (2020). Neuromodulated attention and goal-driven perception in uncertain domains. *Neural Networks*, *125*, 56–69.