Meta-Reinforcement Learning with Self-Modifying Networks

Anonymous Author(s)
Affiliation
Address
email

Abstract

Deep Reinforcement Learning has demonstrated the potential of neural networks tuned with gradient descent for solving complex tasks in well-delimited environments. However, these neural systems are slow learners producing specialized agents with no mechanism to continue learning beyond their training curriculum. On the contrary, biological synaptic plasticity is persistent and manifold, and has been hypothesized to play a key role in executive functions such as working memory and cognitive flexibility, potentially supporting more efficient and generic learning abilities. Inspired by this, we propose to build networks with dynamic weights, able to continually perform self-reflexive modification as a function of their current synaptic state and action-reward feedback, rather than a fixed network configuration. The resulting model, MetODS (for Meta-Optimized Dynamical Synapses) is a broadly applicable meta-reinforcement learning system able to learn efficient and powerful control rules in the agent policy space. A single layer with dynamic synapses can perform one-shot learning, generalize navigation principles to unseen environments and demonstrate a strong ability to learn adaptive motor policies, comparing favorably with previous meta-reinforcement learning approaches.

1 Introduction

The algorithmic shift from hand-designed to learned features characterizing modern deep learning has been transformative for Reinforcement Learning (RL), allowing to solve complex problems ranging from video games [1, 2] to multiplayer contests [3] or motor control [4, 5]. Yet, "deep" RL has mostly produced specialized agents unable to cope with rapid contextual changes or tasks with novel or compositional structure [6–8]. The vast majority of models have relied on gradient-based optimization to learn static network parameters adjusted during a predefined curriculum, arguably preventing the emergence of online adaptivity. A potential solution to this challenge is to *meta-learn* [9–11] computational mechanisms able to rapidly capture a task structure and automatically operate complex feedback control: Meta-

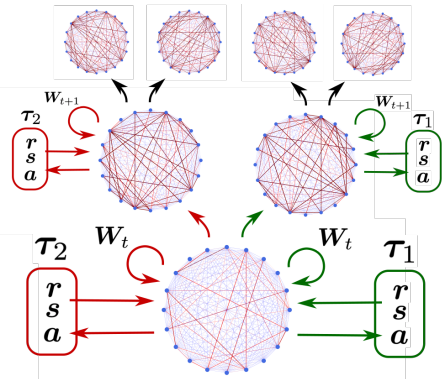

Figure 1: Schema of MetODS. Our model meta-learns to continually specialize its weight state $\boldsymbol{W}_t$ as a function of itself and interactions with different tasks (here, $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$).

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

Reinforcement Learning constitutes a promising direction to build more adaptive artificial systems [12] and identify key neuroscience mechanisms that endow humans with their versatile learning abilities [13].

In this work, we draw inspiration from biological fast synaptic plasticity, hypothesized to orchestrate flexible cognitive functions according to context-dependent rules [14–17]. By tuning neuronal selectivity at fast time scales – from fast neural signaling (milliseconds) to experience-based learning (seconds and beyond) – fast plasticity can support in principle many cognitive faculties including motor and executive control. From a dynamical system perspective, fast plasticity can serve as an efficient mechanism for information storage and manipulation and has led to modern theories of working memory [18–22]. Despite the fact that the magnitude of the synaptic gain variations may be small, such modifications are capable of profoundly altering the network transfer function [23] and constitute a plausible mechanism for rapidly converting reward and choice history into tuned neural functions [24].

From a machine learning perspective, despite having a long history [25–29], fast weights have most often been investigated in conjonction with recurrent neural activations [30, 31] and rarely as a function of an external reward signal or of the current synaptic state itself. In this work, we explore an original self-referential update rule that allows the model to form synaptic updates conditionally on information present in its own synaptic memory. Additionally, environmental reward is injected continually in the model as a rich feedback signal to drive the weight dynamics. These features endow our model with a unique recursive control scheme, that support the emergence of a self-contained reinforcement learning program.

Contribution: We demonstrate that a neural network trained to continually self-modify its weights as a function of sensory information and its own synaptic state can produce a powerful reinforcement learning program. The resulting general-purpose meta-RL agent called 'MetODS' (for Meta-Optimized Dynamical Synapses) is theoretically presented as a model-free approach performing stochastic feedback control in the policy space. In our experimental evaluation, we investigate the reinforcement learning strategies implemented by the model and demonstrate that a single layer with lightweight parametrization can implement a wide spectrum of cognitive functions, from one-shot learning to continuous motor-control, producing better agents than previous meta-RL approaches. We hope that MetODS inspires more works around self-optimizing neural networks.

The remainder of the paper is organised as follows: In Section 2 we introduce our mathematical formulation of the meta-RL problem, which motivates MetODS computational principles presented in Section 3. In Section 4 we review previous approaches of meta-reinforcement learning and we discuss other models of artificial fast plasticity and their relation to associative memory. In Section 5 we report experimental results in 3 different tasks. Finally, in Section 6 we summarise the main advantages of MetODS and outline future work directions.

2 Background

Throughout, we refer to "tasks" as Markov decision processes (MDP) defined by the following tuple $\tau = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \rho_0)$, where $\mathcal{S}$ and $\mathcal{A}$ are respectively the state and action sets, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto [0, 1]$ refers to the state transition distribution measure associating a probability to each tuple (state, action, new state), $r : \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}$ is a bounded reward function and ρ_0 is the initial state distribution. For simplicity, we consider finite-horizon MDP with T time-steps although our discussion can be extended to the infinite horizon case as well as partially observed MDP. We further specify notation when needed by subscripting symbols with the corresponding task τ or time-step t.

Meta-Reinforcement learning considers the problem of generating policies π in a policy space Π that perform well on a set $\mathbb{T}$ of tasks with distribution $\mu_{\mathbb{T}}$, using a reinforcement signal r coming from a sequence of interactions with the task environments. Provided the existence of an optimal policy π^* for any task $\tau \in \mathbb{T}$, we can define the distribution measure μ_{π^*} of these policies over Π. Arguably, an ideal system aims at associating to any task τ its optimal policy π^*, i.e, finding the transport plan $\boldsymbol{\gamma}$ in the space $\Gamma(\mu_{\mathbb{T}}, \mu_{\pi^*})$ of couplings with marginals $\mu_{\mathbb{T}}$ and μ_{π^*} that maximizes the

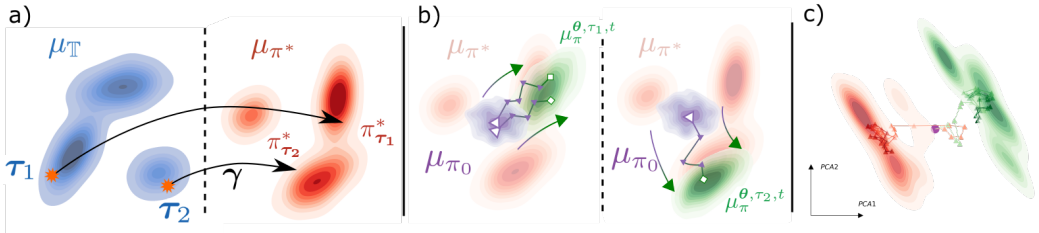

Figure 2: Meta-Reinforcement Learning as a transport problem and MetODS synaptic adaptation: **a)** Associating any task $\boldsymbol{\tau}$ in $\mathbb{T}$ to its optimal policy $\boldsymbol{\pi}^*_{\boldsymbol{\tau}}$ in $\boldsymbol{\Pi}$ can be regarded as finding an optimal transport plan from $\boldsymbol{\mu}_{\mathbb{T}}$ to $\boldsymbol{\mu}_{\boldsymbol{\pi}^*}$ with respect to the cost $-\boldsymbol{\mathcal{R}}$. Finding this transport plan is generally an intractable problem. **b)** Meta-RL approximates a solution by defining a stochastic flow in the policy space $\boldsymbol{\Pi}$ conditioned by the current task $\boldsymbol{\tau}$ and driving a prior distribution $\boldsymbol{\mu}_{\boldsymbol{\pi}_0}$ of policies $\boldsymbol{\pi}_0$ towards a distribution $\mu_{\pi}^{\boldsymbol{\theta},\tau,t}$ of policies with high score $\boldsymbol{\mathcal{R}}$. **c)** Density and mean trajectories of principal components of our model dynamic weights over several episodes of the Harlow task (see section 5.1) reveal this policy specialization. Two modes colored with respect to whether the agent initial guess was good or bad emerge, corresponding to two different policies to solve the task.

expected cumulative reward $\mathcal{R}$:

$$\max_{\gamma \in \Gamma(\mu_{\mathbb{T}}, \mu_{\pi^*})} \mathbb{E}_{(\tau,\pi)\sim\gamma}\Big[\mathcal{R}(\tau,\pi)\Big] \quad \text{where} \quad \mathcal{R}(\tau,\pi) = \mathbb{E}_{\pi,\mathcal{P}_\tau}\Big[\sum_{t=0}^{T} r_\tau(\boldsymbol{a}_t,\boldsymbol{s}_t)\Big] \tag{1}$$

Here $\mathcal{R}$ corresponds to the expected cumulative reward of action-state trajectories$(\boldsymbol{s}_0, \boldsymbol{a}_0, \ldots, \boldsymbol{s}_T, \boldsymbol{a}_T)$ for task τ and policy π, i.e such that state transitions are governed by $\boldsymbol{s}_{t+1} \sim \mathcal{P}_\tau(.|\boldsymbol{s}_t, \boldsymbol{a}_t)$, actions are sampled according to the policy π: $\boldsymbol{a}_t \sim \pi$ and initial state $\boldsymbol{s}_0$ follows the distribution $\rho_{0,\tau}$. Most generally, problem (1) is intractable, since μ_{π^*} is unknow or has no explicit form. Instead, previous approaches aim to optimize a surrogate problem, by defining an iterative "specialization" procedure which builds for any task τ, a sequence (π_t) of improving policies (see Fig. 2). Defining $\boldsymbol{\theta}$ the meta-parameters governing the evolution of the sequences (π_t) and $\mu_{\pi}^{\boldsymbol{\theta},\tau,t}$ the distribution measure of the policy π_t after learning task τ during some period t, the optimization problem amounts to finding the meta-parameters $\boldsymbol{\theta}$ that best adapt $\pi_t \sim \mu_{\pi}^{\boldsymbol{\theta},\tau,t}$ over the task distribution.

$$\max_{\boldsymbol{\theta}} \quad \mathbb{E}_{\tau\sim\mu_{\mathbb{T}}}\Big[\mathbb{E}_{\pi\sim\mu_{\pi}^{\boldsymbol{\theta},\tau,t}}\big[\mathcal{R}(\tau,\pi)\big]\Big] \tag{2}$$

Formulation (2) allows to appreciate the different meta-RL approaches previously proposed under the unifying paradigm of flows in the policy space Π. How such flows are constructed to "guide" policies towards high-reward regions of the policy space determines the quality of a meta-learning system. For instance, imitation learning leverages an oracle teacher that represents a good approximation of the target distribution μ_{π^*} of optimal policies that inform the direction of the policy update [**?**]. When no information or demonstration is provided, τ can only be known by sampling state-action trajectories $(\boldsymbol{s}_t, \boldsymbol{a}_t)_{t\leq T}$ of τ. Model-based meta-RL approaches with posterior approximation of the task such as PEARL [32] and VariBAD [33] accelerate policy adaptation by learning efficient task identification, but might fail when tasks are too different from the training task distribution. On the other hand, MAML [34] is more generic as it corresponds to an explicit gradient flow with respect to $\mathcal{R}$ and benefits theoretically from convergence properties towards a distribution over fixed points in Π even for unseen tasks. However, gradient-based update might be inefficient as it requires a lot of interaction with the environment to yield an accurate estimate of the update direction, and it is unclear at which frequency or which event should trigger an update of the synaptic configuration of the network. Altogether, problem (2) emphasizes three desirable properties of a reinforcement learning program that we further discuss below and for which we show that meta-learnt synaptic updates have a strong potential.

Efficiency: How "fast" the distribution $\mu_{\pi}^{\boldsymbol{\theta},\tau,t}$ is transported towards a distribution of high-performing policies when accumulating experience on a task τ. Arguably, an efficient learning mechanism should

require few interaction steps t with the environment to identify the task rule and adapt its policy accordingly. We show that learnt synaptic update rules are able to change the agent transfer function drastically in a few updates, thus supporting one-shot learning of a task-contingent association rule in the Harlow task, adapting a motor policy in a few steps or exploring original environments quickly in the Maze experiment.

Capacity: That property defines how sensitive the learner is to specific task features and determines its achievable level of performance for a distribution of tasks μ_τ. It is linked to the sensitivity of learning system shaping $\mu_\pi^{\boldsymbol{\theta},\tau,t}$ to task particularities, i.e how the agent captures and retains task sufficient statistics, as well as their conversion into a precise state in the policy space. Because our mechanism is continual, it allows for constant tracking of the environment information and policy update. We test this property in the maze experiment in Section 5, showing that tuned online synaptic updates obtain the best capacity under systematic variation of the environment.

Generality: We refer here to the overall ability of the meta-learnt policy flow to drive $\mu_\pi^{\boldsymbol{\theta},\tau,t}$ towards high-performing policy regions for a diverse set of tasks (*generic trainability*) but also to how general is the resulting reinforcement learning program and how well it transfers to tasks unseen during training? (*transferability*). We show in the former case that our synaptic mechanism being model-free, it allows for tackling diverse types of policy learning, from navigation to motor control. Arguably, to build reinforcing agents that learn in open situations, we should strive for generic and efficient computational mechanisms rather than learnt heuristics. For transferability, this corresponds to the ability of the policy flow to generally yield improving updates even in unseen policy regions of the space $\mathbf{\Pi}$ or conditioned by unseen task properties: new states, actions and transitions, new reward profile. etc. We show in a motor-control experiment using the Meta-World benchmark that meta-tuned synaptic updates are a potential candidate to produce a more systematic learner agnostic to environment setting and reward profile. The generality property remains the hardest for current meta-RL approaches, demonstrating the importance of building more stable and invariant control learning rules.

3 MetODS: Meta-Optimized Dynamic synapses

Our model learns to train itself by updating its weights through interaction with the environment and its own current weight state. This mechanism enables MetODS to rapidly compress experience of a task τ into a particular synaptic configuration, building the following policy sequence:

$$\forall t \leq T, \qquad \pi(\boldsymbol{a}|\boldsymbol{s}, \boldsymbol{W}_t) \sim \mu_\pi^{\boldsymbol{\theta},\tau,t} \tag{3}$$

where $\boldsymbol{W}_t$ are instance-particular dynamic weights governed by locally parameterized update rules driving their evolution over time and with respect to the state trajectory $(\boldsymbol{s}_{i\leq t}, \boldsymbol{a}_{i<t}, \boldsymbol{r}_{i<t})$. Specifically, at every time-step t, network computation and learning rules consists in recursive application of read-write operations such that the model learns to update its weights given both external stimuli $\boldsymbol{v}_t$ and relevant information stored in the network synaptic configuration $\boldsymbol{W}_t$.

Update operations: The core mechanism consists in two simple operations that respectively linearly project neurons activation $\boldsymbol{v}$ through the dynamic weights $\boldsymbol{W}$ followed by an element-wise non-linearity, and build an update through a hebbian update rule with element-wise weighting $\boldsymbol{\alpha}$:

$$\begin{cases} \phi(\boldsymbol{W}, \boldsymbol{v}) = \sigma(\boldsymbol{W}.\boldsymbol{v}) & \textit{read} \\ \psi(\boldsymbol{v}) = \boldsymbol{\alpha} \odot \boldsymbol{v} \otimes \boldsymbol{v} & \textit{write} \end{cases} \tag{4}$$

where, $\boldsymbol{\alpha}$ is a matrix of $\mathbb{R}^{N\times N}$, $\otimes$ denotes the outer-product, $\odot$ is the element-wise multiplication, σ is a non-linear activation function. The element-wise weighting $\boldsymbol{\alpha}$ allows for different plasticity amplitudes at every connection consistent with biology and locally tuned synaptic plasticity [35, 36], and generating a matrix update with potentially more than rank one as in the classic hebbian rule.

Multi-step scheme We further introduce a multi-step scheme that recursively applies the previous rules S times. This scheme allows to learn relations between stored patterns and incoming information

by mixing information between current neural activation and previous iterates. Starting from an initial activation pattern $\boldsymbol{v}^{(0)}$ and previous weight state $\boldsymbol{W}^{(0)} = \boldsymbol{W}_{t-1}$, the model recursively applies equations in (4) on $\boldsymbol{v}^{(s)}$ and $\boldsymbol{W}^{(s)}$ such that:

$$\text{for } s \in [1, S] \quad : \quad \begin{cases} \boldsymbol{v}^{(s)} = \sum_{l=0}^{s-1} \boldsymbol{\kappa}_s^{(l)} \boldsymbol{v}^{(l)} + \boldsymbol{\kappa}_s^{(s)} \phi(\boldsymbol{W}^{(s-1)}, \boldsymbol{v}^{(s-1)}) \\ \boldsymbol{W}^{(s)} = \sum_{l=0}^{s} \boldsymbol{\beta}_s^{(l)} \boldsymbol{W}^{(l)} + \boldsymbol{\beta}_s^{(s)} \psi(\boldsymbol{v}^{(s-1)}) \end{cases} \tag{5}$$

Parameters $\boldsymbol{\kappa}_s^{(l)}$ and $\boldsymbol{\beta}_s^{(l)}$ are learnt along with plasticity parameters $\boldsymbol{\alpha}$, and correspond to delayed contributions of previous patterns and synaptic states to the current operation. This is motivated by biological evidence of different time-scales in synaptic neuromodulators concentration change and of their mutual retroactive influence over synaptic efficacy. Finally, $(\boldsymbol{v}^{(S)}, \boldsymbol{W}^{(S)})$ are respectively used as activation for the next layer, and as the new synaptic state $\boldsymbol{W}_t$. In this work, we test a single dynamic layer and leave the extension of the synaptic plasticity to the full network for future work. In order for the model to learn a credit assignment strategy, state transition and previous reward information $[\boldsymbol{s}_t, \boldsymbol{a}_{t-1}, \boldsymbol{r}_{t-1}]$ are embedded into a vector $\boldsymbol{v}_t$ by a feedforward map $\boldsymbol{f}$ as in previous meta-RL approaches [37, 38]. Action and advantage estimate are read-out by a feedforward policy map $\boldsymbol{g}$. Altogether, the synaptic control as well as the state-value and action policy estimate of MetODS consists in the following update summed in Algorithm 1. We meta-learn the plasticity and update coefficients, as well as the embedding and read-out function altogether, hence meta-parameters are $\boldsymbol{\theta} = [\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{\alpha}, \boldsymbol{\kappa}, \boldsymbol{\beta}]$. Additionally, the initial synaptic configuration $\boldsymbol{W}_0$ can be learnt, fixed a priori or sampled from an specified distribution.

Algorithm 1 MetODS synaptic learning

Require: $\boldsymbol{\theta} = [\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{\alpha}, \boldsymbol{\kappa}, \boldsymbol{\beta}]$ and $\boldsymbol{W}_0$
for $1 \leq t \leq T$ **do**
  $\boldsymbol{v}^{(0)} \leftarrow f(\boldsymbol{s}_t, \boldsymbol{a}_{t-1}, \boldsymbol{r}_{t-1})$
  $\boldsymbol{W}^{(0)} \leftarrow \boldsymbol{W}_{t-1}$
  for $1 \leq s \leq S$ **do**
    $\boldsymbol{v}^{(s)} \leftarrow \sum_{l=0}^{s-1} \boldsymbol{\kappa}_s^{(l)} \boldsymbol{v}^{(l)} + \boldsymbol{\kappa}_s^{(s)} \sigma(\boldsymbol{W}^{(s-1)}.\boldsymbol{v}^{(s-1)})$
    $\boldsymbol{W}^{(s)} \leftarrow \sum_{l=0}^{s-1} \boldsymbol{\beta}_s^{(l)} \boldsymbol{W}^{(l)} + \boldsymbol{\beta}_s^{(s)} \big(\boldsymbol{\alpha} \odot \boldsymbol{v}^{(s-1)} \otimes \boldsymbol{v}^{(s-1)}\big)$
  end for
  $\boldsymbol{a}_t, \boldsymbol{v}_t \leftarrow \boldsymbol{g}(\boldsymbol{v}^{(s)})$
  $\boldsymbol{W}_t \leftarrow \boldsymbol{W}^{(s)}$
end for=0

Computational interpretation: We note that if S=1 in equation (5), the operation boils down to a simple hebbian update with a synapse-specific weighting $\boldsymbol{\alpha}^{i,j}$. This perspective makes MetODS an original form of modern Hopfield network [39] with hetero-associative memory that can dynamically access and edit stored representations driven by observations, rewards and actions. While pattern retrieval from Hopfield networks has a dense litterature, our model recursive scheme is an original proposal to learn automatic synaptic updates able to bind representations across timesteps. The promising results shown in our experimental section suggest that such learnt updates can generate useful self-modifications to sequentially adapt to incoming information at runtime.

3.1 Optimization

Defining the weight parameters $\boldsymbol{W}$ of MetODS as dynamic variables lifts the optimization problem (2) into a functional space of control functions parameterized by $\boldsymbol{\theta}$. Hence, meta-optimizing the control necessitates the estimation of gradients with respect to $\boldsymbol{\theta}$ over the space $\mathbb{T}$ and for any possible trajectory π_t in Π. Interestingly, previous meta RL approaches have performed policy gradient

optimization by sampling a single policy trajectory $\pi_t \sim \mathcal{M}_{\boldsymbol{\theta}}(\tau)$ over M multiple tasks, showing that it is sufficient to obtain correct gradient estimates on $\boldsymbol{\theta}$. We proceed in the same way, by estimating the gradient policy update integrated over the space of tasks as mini-batches over tasks.

$$\frac{\partial}{\partial \boldsymbol{\theta}} \mathbb{E}_{\tau \sim \mu_{\mathbb{T}}} \left[\mathbb{E}_{\pi \sim \mu_{\pi}^{\boldsymbol{\theta}, \tau, t}} \left[\mathcal{R}(\tau, \pi) \right] \right] \quad \approx \sum_{\tau_1, \ldots, \tau_n} \sum_{t=0}^{T} \frac{\partial \log \pi_t(\boldsymbol{a}_t | \boldsymbol{W}_t, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} r_{\tau_i}(\boldsymbol{a}_t, \boldsymbol{s}_t) \tag{6}$$

Additionally, the memory cost of storing synaptic weights trajectories instead of hidden activity in a network of N neurons is $\mathcal{O}(N^2)$ instead of $\mathcal{O}(N)$. This might lead to prohibitively large memory requirements for training with BPTT [40] over long episodes. We present in S.I an alternative solution to train the model through the discrete adjoint sensitivity method, leveraging the work of [41] yielding a memory cost of $\mathcal{O}(1)$. The agent's log-policy total derivative with respect to $\boldsymbol{\theta}$ can be computed as the solution of an augmented adjoint problem [42].

4 Related work

Meta-Reinforcement learning has recently flourished into several different approaches aiming at learning high-level strategies for capturing task rules and structures. A direct line of work consists in automatically meta-learning components or parameters of the RL arsenal to improve over heuristic settings [43–45]. Orthogonally, work building on the Turing-completeness of recurrent neural networks has shown that simple recurrent neural networks can be trained to store past information in their persistent activity state to inform current decision, in such a way that the network implements a form of reinforcement learning over each episode [46, 37, 47]. It is believed that vanilla recurrent networks alone are not sufficient to meta-learn the efficient forms of episodic control found in biological agents [48, 13]. Hence additional work has tried to enhance the system with a better episodic memory model [49–51] or by modeling a policy as an attention module over an explicitely stored set of past events [38]. Optimization based approaches have tried to cast episodic adaptation as an explicit optimization procedure either by treating the optimizer as a black-box system [52, 53] or by learning a synaptic configuration such that one or a few gradient steps are sufficient to adapt the input/output mapping to a specific task [34].

Artificial fast plasticity: Networks with dynamic weights that can adapt as a function of neural activation have shown promising results over regular recurrent neural networks to handle sequential data [54, 29, 55, 56, 31, 57]. However, contrary to our work, these models postulate a persistent neural activity orchestrating weights evolution. On the contrary, we show that synaptic states are the sole persistent components needed to perform fast adaptation. Additionally, the possibility of optimizing synaptic dynamics with evolutionary strategies in randomly initialized networks [58] or through gradient descent [59] has been demonstrated, as well as in a time-continuous setting [60]. Recent results have shown that plasticity rules differentially tuned at the synapse level allow to dynamically edit and query networks memory [55, 42]. However another specificity of this work is that our model synaptic rule is a function of reward and synaptic state, allowing to drive weight dynamics conditionally on both an external feedback signal and the current model belief.

Associative memory: As discussed above, efficient memory storage and manipulation is a crucial feature for building rapidly learning agents. To improve over vanilla recurrent neural network policies [37], some models have augmented recurrent agents with content-addressable dictionaries able to reinstate previously encoded patterns given the current state [61–63, 13]. However these slot-based memory systems are subject to interference with incoming inputs and their memory cost grows linearly with experience. Contrastingly, attractor networks can be learnt to produce fast compression of sensory information into a fixed size tensorial representation [64, 65]. One class of such network are Hopfield networks [66–69] which benefit from a large storage capacity [69], can possibly perform hetero-associative concept binding [70, 57] and produce fast and flexible information retrieval [39].

5 Experiments

In this section, we demonstrate the potential of meta-learnt synaptic updates with respect to the three properties of the meta-RL problem exposed in section 2. Namely, 1) *efficiency* of the learning program. 2) *capacity* of the meta-learner with respect to task particularity, and finally 3) *generality* of the produced algorithm. We compare when possible our model with three state-of-the-art meta-RL algorithms previously discussed: RL^2 [37], MAML [34] and PEARL [32]. We show in those settings that our meta-optimized synaptic learner compares favorably to these approaches with no particular tuning. All of the experiments were performed using PyTorch [71] which allows us to compute the gradient of all the plasticity parameters using automatic differentiation. Experimental details are further discussed in S.I. The code will be made accessible upon acceptance of the present work.

5.1 Efficiency: One-shot reinforcement learning and rapid motor control

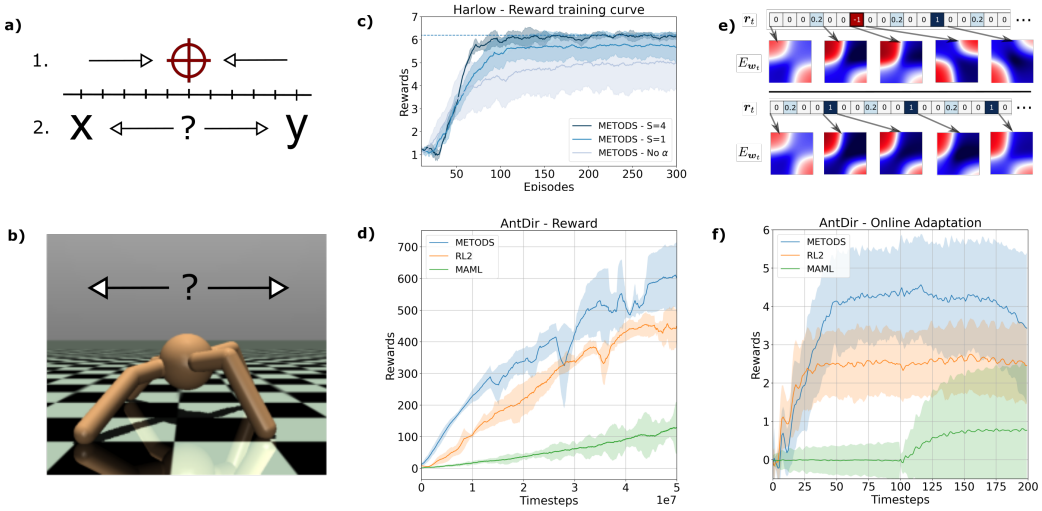

Figure 3: a-b) Schemas of the Harlow and Mujoco *Ant-directional* locomotion task. An episode of the Harlow task consists of five sequential presentations of two random variables placed on a one-dimensional line with random permutation of their positions that an agent must select by reaching the corresponding position. One value is associated with a positive reward and the other with a negative reward. The five trials are presented in alternance with periods of fixation where the agent should return to a neutral position between items. In *Ant-dir*, the agent must learn to select a rewarded direction of locomotion over a single episode of 200 steps. **c-d)** Evolution of accumulated reward over training. In the Harlow task, we conduct an ablation study by either reducing the number of recursive iterations (S=1) or removing the trainable plasticity weights $\boldsymbol{\alpha}$ resulting in sub-optimal policy. In *Ant-dir* we compare our agent training profile against MAML and RL^2. **e)** We can interpret the learned policy in terms of a Hopfield energy adapting with experience. We show horizontally two reward profiles of different episodes and the energy $E_{\boldsymbol{W}_t}(v_1, v_2) = v_1^T \boldsymbol{W}_t v_2$ along two principal components of the vector trajectory $\boldsymbol{v_t}$. In the first episode, the error in the first presentation (red square) transforms the energy landscape which changes the agent policy, while on the other episode, the model belief does not change over time. Note the two modes for every energy map, which allows the model to handle the potential position permutation of the presented values. **f)** Average rewards per timestep during a single episode of the *Ant-dir* task.

To first illustrate that learnt synaptic dynamics can support fast behavioral adaptation, we use a classic experiment from the neuroscience literature originally presented by Harlow [72] and recently reintroduced in artificial meta-RL in [47] as well as a heavily-benchmarked Mujoco directional locomotion task (see Fig. 3 for description). To behave optimally in both settings, the agent must quickly identify the task structure: In the Harlow task, since the location values are randomly permuted across presentations, the agent cannot develop a mechanistic strategy to reach high rewards based on initial position. Instead, to reach the maximal expected reward over the episode, the agent needs to perform one-shot learning of the task-contingent association rule during the first presentation. We found that even a very small network of N=20 neurons proved to be sufficient to solve the task perfectly. We investigated the synaptic mechanism encoding the agent policy. A principal component

analysis reveals a differentiation of the synaptic configuration with respect to the initial value choice outcome (see Figure 2.c) that we can interpret as a change in the Hopfield energy of the dynamic weights (Figure 3.e). Moreover, the largest synaptic variations measured by the sum of absolute synaptic variations occur for states that carry a non-null reward signal (see S.I). Theses results suggest that the recursive hebbian update combined with reward feedback is sufficient to support one-shot reinforcement learning of the task association rule. Similarly in *Ant-dir*, we found that MetODS can adapt within a single episode in a few time-steps thanks to its continual adaptation mechanism (similar to RL^2). By design, MAML and PEARL do not present such a property, and they need multiple episodes before being able to perform adaptation correctly. We still report MAML performance after running its gradient adaptation at timestep t=100.

5.2 Capacity : Maze exploration task

We further test the systematicity of our model learnt reinforcement program on a more challenging partially observable Markov decision process (POMDP): An agent must locate a target in a randomly generated maze while starting from random locations and only being able to observe a small portion of its environment. While visual navigation has been previously explored in meta-RL [37, 38, 55], we here focus on the mnemonic component of navigation by complexifying the task in two ways depicted in Figure 4, we reduce the agent's visual field to a small size of 3x3 pix. and randomize the agent's position after every reward encounter. The agent can take discrete actions in the set {up,down,left,right} which moves it accordingly by one coordinate. The agent's reward signal is solely received by hitting the target location, thus receiving a reward of 10. Note that the reward is invisible to the agent, and thus the agent only knows it has hit the reward location because of the activation of the reward input. The reduced observability of the environment and the sparsity of the reward signal (most of the state transitions yield no reward) requires the agent to perform logical binding between distant temporal events to navigate the maze. Again, this setting rules out PEARL since its latent context encoding mechanism erases temporal dependencies between state transitions which are here crucial for efficient exploration. We note that our system can also be combined with approaches that perform posterior task inference such as VariBAD [33] which we leave for future work. Despite having no particular inductive bias for efficient spatial exploration or path memorization, a strong policy emerges spontaneously from training. We additionally test the capability of the learnt navigation skills to generalize to to a larger maze size of 10×10 pix. unseen during training. We show that MetODS is able to retain its advantage (see table 5 for results and S.I for full experimental details).

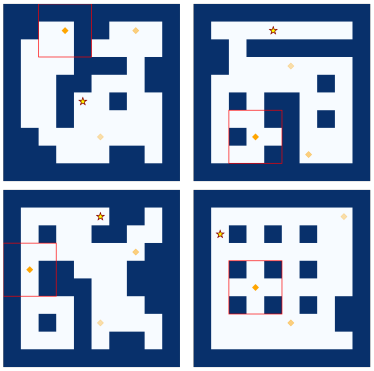

Figure 4: Examples of generated maze configurations. The mazes consist in 8 × 8 pix. areas, with walls and obstacles randomly placed according to a variation of Prim's algorithm [73] while the target location is selected randomly for the entire duration of the episode (star). Each time the agent hits the target, its position is randomly reassigned on the map (orange) and the exploration resumes until 100 steps are accumulated during which the agent must collect as much reward as possible. The agent receptive field is highlighted in red.

Agent	1st rew.*(↓)	Success (↑)	Cum. Rew. (↑)	Cum. Rew (Larger) (↑)
Random	96.8 ± 0.5	5%	3.8 ± 8.9	3.7 ± 6.4
MAML	64.3 ± 39.3	45.2%	14.95 ± 4.5	5.8 ± 10.3
RL^2	16.2 ± 1.1	96.2%	77.7 ± 46.5	28.1 ± 29.7
MetODS	**14.7 ± 1.4**	**96.6%**	**86.5 ± 46.8**	**34.9 ± 34.9**

Figure 5: MetODS better explores the maze as measured by the average number of steps before 1st reward and the success rate in finding the reward at least once. It then better exploits the maze as per the accumulated reward. (* We assign 100 to episodes with no reward encounter.)

5.3 Generality : Motor control

Finally, we test the generality of the reinforcement learning program learnt by our model for different continuous control tasks.

MetaWorld: First, we use the dexterous manipulation benchmark proposed in [**?**] in which a Sawyer robot is tasked with diverse operations. A full adaptation episode consists in N=10 rollouts of 500 timesteps of the same task across which dynamic weights are carried over. Observation consists in the robot's joint angles and velocities, and the actions are its joint torques. We compare MetODS with baseline methods in terms of meta-training and meta-testing success rate for 3 settings, *push, reach* and *ML-10*. We show in Fig. 6 the meta-training results for all the methods in the MetaWorld environments. All subplots show the task success rate over the training timesteps. Due to computational resource constraints, we restrict our experiment to a budget of 10M steps per run. We note that all tested approaches performed modestly on ML10 for test tasks, which highlights the limitation of current methods. We conjecture that this might be due to the absence of inductive bias for sharing knowledge between tasks or fostering systematic exploration of the environment of the tested meta-learning algorithms.

Robot impairment: We also tested the robustness of MetODS learnt reinforcement programs by evaluating the agent ability to perform in a setting not seen during training: specifically, when partially impairing the agent motor capabilities. We adopt the same experimental setting as section 5.1 for the *Ant* and *Cheetah* robots and evaluate the performance when freezing one of the robots torque. We show that our model policy retains a better proportion of its performance compared to other approaches. These results suggest that learning fast synaptic dynamics is not only better suited to support fast adaptation of a motor policy in the continuous domain, but they also implement a more robust reinforcement learning program when impairing the agent motor capabilities.

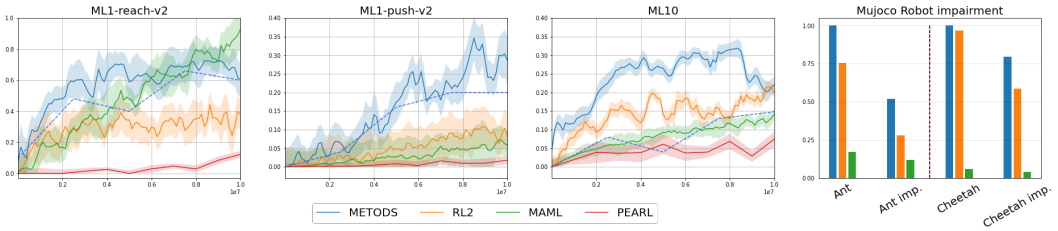

Figure 6: Left Meta-training results for MetaWorld benchmarks. Average meta-test results for MetODS is shown in dotted line. **Right** Cumulative reward of the *Ant* and *Cheetah* directional locomotion task. For each condition, results are normalized against the best performing policy

6 Discussion

In this work, we introduce a novel meta-RL system, MetODS, which leverages a self-referential weight update mechanism for rapid specialization at the episodic level. Our approach is generic and supports discrete and continuous domains, giving rise to a promising repertoire of skills such as one-shot adaptation, spatial navigation or motor coordination. MetODS compares favorably with prior meta-RL algorithms and we conjecture that further tuning the hyperparameters as well as combining MetODS with more sophisticated reinforcement learning techniques can boost its performance. Generally, the success of the approach provides evidence for the benefits of fast plasticity in artificial neural networks, and the exploration of self-referential networks.

References

[1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing atari with deep reinforcement learning. *ArXiv*, abs/1312.5602, 2013.

[2] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[3] Max Jaderberg, Wojciech M. Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio Garcia Castañeda, Charles Beattie, Neil C. Rabinowitz, Ari S. Morcos, Avraham Ruderman, Nicolas Sonnerat, Tim Green, Louise Deason, Joel Z. Leibo, David Silver, Demis Hassabis, Koray Kavukcuoglu, and Thore Graepel. Human-level performance in 3d multiplayer games with population-based reinforcement learning. *Science*, 364(6443):859–865, 2019.

[4] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

[5] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *ICLR (Poster)*, 2016.

[6] Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40:e253, 2017.

[7] Brenden Lake and Marco Baroni. Still not systematic after all these years: On the compositional skills of sequence-to-sequence recurrent networks, 2018.

[8] Karl Cobbe, Christopher Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning, 2020.

[9] Jürgen Schmidhuber, Jieyu Zhao, and Marco Wiering. Shifting inductive bias with success-story algorithm, adaptive levin search, and incremental self-improvement. *Machine Learning*, 28, 01 1997.

[10] Sebastian Thrun. *Lifelong Learning Algorithms*, pages 181–209. Springer US, Boston, MA, 1998.

[11] Ricardo Vilalta and Youssef Drissi. A perspective view and survey of meta-learning. *Artif. Intell. Rev.*, 18(2):77–95, oct 2002.

[12] Jeff Clune. Ai-gas: Ai-generating algorithms, an alternate paradigm for producing general artificial intelligence. *CoRR*, abs/1905.10985, 2019.

[13] Matthew Botvinick, Sam Ritter, Jane X. Wang, Zeb Kurth-Nelson, Charles Blundell, and Demis Hassabis. Reinforcement learning, fast and slow. *Trends in Cognitive Sciences*, 23(5):408–422, 2019.

[14] S. J. Martin, P. D. Grimwood, and R. G. M. Morris. Synaptic plasticity and memory: An evaluation of the hypothesis. *Annual Review of Neuroscience*, 23(1):649–711, 2000.

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

[16] Wade G Regehr. Short-term presynaptic plasticity. *Cold Spring Harbor perspectives in biology*, 4(7):a005702, 2012.

[17] Natalia Caporale and Yang Dan. Spike timing–dependent plasticity: A hebbian learning rule. *Annual Review of Neuroscience*, 31(1):25–46, 2008. PMID: 18275283.

[18] Gianluigi Mongillo, Omri Barak, and Misha Tsodyks. Synaptic theory of working memory. *Science*, 319(5869):1543–1546, 2008.

[19] Nicolas Masse, Guangyu Yang, H. Song, Xiao-Jing Wang, and David Freedman. Circuit mechanisms for the maintenance and manipulation of information in working memory, 04 2018.

[20] Omri Barak and Misha Tsodyks. Working models of working memory. *Current Opinion in Neurobiology*, 25:20–24, 2014. Theoretical and computational neuroscience.

[21] Mark G. Stokes. 'activity-silent' working memory in prefrontal cortex: a dynamic coding framework. *Trends in Cognitive Sciences*, 19(7):394–405, 2015.

[22] Sanjay G. Manohar, Nahid Zokaei, Sean J. Fallon, Tim P. Vogels, and Masud Husain. Neural mechanisms of attending to items in working memory. *Neuroscience and Biobehavioral Reviews*, 101:1–12, 2019.

[23] Pierre Yger, Marcel Stimberg, and Romain Brette. Fast learning with weak synaptic plasticity. *Journal of Neuroscience*, 35(39):13351–13362, 2015.

[24] Răzvan V Florian. Reinforcement learning through modulation of spike-timing-dependent synaptic plasticity. *Neural computation*, 19(6):1468–1502, 2007.

[25] Geoffrey E Hinton and David C Plaut. Using fast weights to deblur old memories. In *Proceedings of the 9th Annual Conference of the Cognitive Science Society*, pages 177–186, 1987.

[26] Jürgen Schmidhuber. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139, 1992.

[27] Jürgen Schmidhuber. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4:131–139, 1992.

[28] Christoph von der Malsburg. *The Correlation Theory of Brain Function*, pages 95–119. Springer New York, New York, NY, 1994.

[29] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

[30] Jimmy Ba, Geoffrey E Hinton, Volodymyr Mnih, Joel Z Leibo, and Catalin Ionescu. Using fast weights to attend to the recent past. *Advances in Neural Information Processing Systems*, 29:4331–4339, 2016.

[31] Imanol Schlag and Jürgen Schmidhuber. Learning to reason with third-order tensor products. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 10003–10014, 2018.

[32] Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International conference on machine learning*, pages 5331–5340. PMLR, 2019.

[33] Luisa Zintgraf, Kyriacos Shiarlis, Maximilian Igl, Sebastian Schulze, Yarin Gal, Katja Hofmann, and Shimon Whiteson. Varibad: A very good method for bayes-adaptive deep rl via meta-learning. *arXiv preprint arXiv:1910.08348*, 2019.

[34] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR, 06–11 Aug 2017.

[35] Larry F Abbott and Sacha B Nelson. Synaptic plasticity: taming the beast. *Nature neuroscience*, 3(11):1178–1183, 2000.

[36] Tuning into diversity of homeostatic synaptic plasticity. *Neuropharmacology*, 78:31–37, 2014. Homeostatic Synaptic Plasticity.

[37] Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. Rl2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

[38] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018.

[39] Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Lukas Gruber, Markus Holzleitner, Thomas Adler, David Kreil, Michael K Kopp, et al. Hopfield networks is all you need. In *International Conference on Learning Representations*, 2020.

[40] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[41] Michael Betancourt, Charles C Margossian, and Vianey Leos-Barajas. The discrete adjoint method: Efficient derivatives for functions of discrete sequences. *arXiv preprint arXiv:2002.00326*, 2020.

[42] Mathieu Chalvidal, Matthew Ricci, Rufin VanRullen, and Thomas Serre. Go with the flow: Adaptive control for neural ODEs. In *International Conference on Learning Representations*, 2021.

[43] Rein Houthooft, Richard Y Chen, Phillip Isola, Bradly C Stadie, Filip Wolski, Jonathan Ho, and Pieter Abbeel. Evolved policy gradients. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 5405–5414, 2018.

[44] Zhongwen Xu, Hado van Hasselt, and David Silver. Meta-gradient reinforcement learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 2402–2413, 2018.

[45] Abhishek Gupta, Russell Mendonca, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Meta-reinforcement learning of structured exploration strategies. *Advances in Neural Information Processing Systems*, 31:5302–5311, 2018.

[46] Sepp Hochreiter, A Steven Younger, and Peter R Conwell. Learning to learn using gradient descent. In *International Conference on Artificial Neural Networks*, pages 87–94. Springer, 2001.

[47] Jane X. Wang, Zeb Kurth-Nelson, Dharshan Kumaran, Dhruva Tirumala, Hubert Soyer, Joel Z. Leibo, Demis Hassabis, and Matthew Botvinick. Prefrontal cortex as a meta-reinforcement learning system. 2018.

[48] Máté Lengyel and Peter Dayan. Hippocampal contributions to control: The third way. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2008.

[49] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850. PMLR, 2016.

[50] Alexander Pritzel, Benigno Uria, Sriram Srinivasan, Adria Puigdomenech Badia, Oriol Vinyals, Demis Hassabis, Daan Wierstra, and Charles Blundell. Neural episodic control. In *International Conference on Machine Learning*, pages 2827–2836. PMLR, 2017.

[51] Samuel Ritter, Jane X Wang, Zeb Kurth-Nelson, and M Botvinick. Episodic control as meta-reinforcement learning. *bioRxiv*, page 360537, 2018.

[52] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 1842–1850. JMLR.org, 2016.

[53] Sachin Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017.

[54] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *International Conference on Machine Learning*, pages 2554–2563. PMLR, 2017.

[55] Thomas Miconi, Kenneth Stanley, and Jeff Clune. Differentiable plasticity: training plastic neural networks with backpropagation. In *International Conference on Machine Learning*, pages 3559–3568. PMLR, 2018.

[56] Thomas Miconi, Aditya Rawal, Jeff Clune, and Kenneth O Stanley. Backpropamine: training self-modifying neural networks with differentiable neuromodulated plasticity. In *International Conference on Learning Representations*, 2018.

[57] Imanol Schlag, Tsendsuren Munkhdalai, and Jürgen Schmidhuber. Learning associative inference using fast weight memory. In *International Conference on Learning Representations*, 2020.

[58] Elias Najarro and Sebastian Risi. Meta-learning through hebbian plasticity in random networks. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20719–20731. Curran Associates, Inc., 2020.

[59] Thomas Miconi. Learning to learn with backpropagation of hebbian plasticity. *arXiv: Neural and Evolutionary Computing*, 2016.

[60] Krzysztof Choromanski, Jared Davis, Valerii Likhosherstov, Xingyou Song, Jean-Jacques E. Slotine, Jacob Varley, Honglak Lee, Adrian Weller, and Vikas Sindhwani. An ode to an ode. *ArXiv*, abs/2006.11421, 2020.

[61] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014.

[62] Wojciech Zaremba and Ilya Sutskever. Reinforcement learning neural turing machines - revised, 2016.

[63] Alexander Pritzel, Benigno Uria, Sriram Srinivasan, Adrià Puigdomènech Badia, Oriol Vinyals, Demis Hassabis, Daan Wierstra, and Charles Blundell. Neural episodic control. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2827–2836. PMLR, 06–11 Aug 2017.

[64] Sergey Bartunov, Jack Rae, Simon Osindero, and Timothy Lillicrap. Meta-learning deep energy-based memory models. In *International Conference on Learning Representations*, 2019.

[65] Wei Zhang and Bowen Zhou. Learning to update auto-associative memory in recurrent neural networks for improving sequence memorization, 2017.

[66] J J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.

[67] Pascal Koiran. Dynamics of discrete time, continuous state hopfield networks. *Neural Computation*, 6(3):459–468, 1994.

[68] Mete Demircigil, Judith Heusel, Matthias Löwe, Sven Upgang, and Franck Vermet. On a model of associative memory with huge storage capacity. *Journal of Statistical Physics*, 02 2017.

[69] Dmitry Krotov and John J. Hopfield. Dense associative memory for pattern recognition. In *NIPS*, 2016.

[70] Imanol Schlag and Jürgen Schmidhuber. Learning to reason with third order tensor products. In *NeurIPS*, pages 10003–10014, 2018.

[71] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[72] Harry Frederick Harlow. The formation of learning sets. *Psychological review*, 56 1:51–65, 1949.

[73] Robert Clay Prim. Shortest connection networks and some generalizations. *The Bell System Technical Journal*, 36(6):1389–1401, 1957.

7 Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [No] N.A
   (b) Did you include complete proofs of all theoretical results? [No] N.A
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The experimental details can be found in the supplementary information section. Code will be released upon acceptation of the present submission.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See S.I
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] All experiments are reported fro three different seeds
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See S.I
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Software and baselines used are correctly cited in the main text
   (b) Did you mention the license of the assets? [No] N.A
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] N.A
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] N.A
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [No] N.A
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [No] N.A
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [No] N.A