frontiers
in Computational Neuroscience

ORIGINAL RESEARCH
published: 12 July 2018
doi: 10.3389/fncom.2018.00050

Multi-Timescale Memory Dynamics Extend Task Repertoire in a Reinforcement Learning Network With Attention-Gated Memory

Marco Martinolli, Wulfram Gerstner and Aditya Gilra**

School of Computer and Communication Sciences, School of Life Sciences, Brain-Mind Institute, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland

The interplay of reinforcement learning and memory is at the core of several recent neural network models, such as the Attention-Gated MEmory Tagging (`AuGMEnT`) model. While successful at various animal learning tasks, we find that the `AuGMEnT` network is unable to cope with some hierarchical tasks, where higher-level stimuli have to be maintained over a long time, while lower-level stimuli need to be remembered and forgotten over a shorter timescale. To overcome this limitation, we introduce a hybrid `AuGMEnT`, with leaky (or short-timescale) and non-leaky (or long-timescale) memory units, that allows the exchange of low-level information while maintaining high-level one. We test the performance of the hybrid `AuGMEnT` network on two cognitive reference tasks, sequence prediction and 12AX.

Keywords: reinforcement learning, memory, attention, synaptic plasticity, eligibility trace, synaptic tagging

Edited by:
Paul Miller,
Brandeis University, United States

Reviewed by:
Pieter R. Roelfsema,
Netherlands Institute for Neuroscience (KNAW), Netherlands
Mattia Rigotti,
IBM Research (United States), United States

****Correspondence:***
Marco Martinolli
marco.martinolli@polimi.it
Aditya Gilra
agilra@uni-bonn.de

Received: *24 January 2018*
Accepted: *18 June 2018*
Published: *12 July 2018*

Citation:
Martinolli M, Gerstner W and Gilra A (2018) Multi-Timescale Memory Dynamics Extend Task Repertoire in a Reinforcement Learning Network With Attention-Gated Memory. Front. Comput. Neurosci. 12:50. doi: 10.3389/fncom.2018.00050

1. INTRODUCTION

Memory spans various timescales and plays a crucial role in human and animal learning (Tetzlaff et al., 2012). In cognitive neuroscience, the memory system that enables manipulation and storage of information over a period of a few seconds is called Working Memory (WM), and is correlated with activity in prefrontal cortex (PFC) and basal ganglia (BG) (Mink, 1996; Frank et al., 2001). In computational neuroscience, there are not only several standalone models of WM dynamics (Samsonovich and McNaughton, 1997; Compte et al., 2000; Barak and Tsodyks, 2014), but also supervised and reinforcement learning models augmented by working memory (Graves et al., 2014, 2016; Alexander and Brown, 2015; Rombouts et al., 2015; Santoro et al., 2016).

Memory mechanisms can be implemented by enriching a subset of artificial neurons with slow time constants and gating mechanisms (Hochreiter and Schmidhuber, 1997; Gers et al., 2000; Cho et al., 2014). More recent memory-augmented neural network models like the Neural Turing Machine (Graves et al., 2014) and the Differentiable Neural Computer (Graves et al., 2016), employ an addressable memory matrix that works as a repository of past experiences and a neural controller that is able to store and retrieve information from the external memory to improve its learning performance.

Here, we study and extend the Attention-Gated MEmory Tagging model or `AuGMEnT` (Rombouts et al., 2015). `AuGMEnT` is trained with a Reinforcement Learning (RL) scheme, where learning is based on a reward signal that is received after each action selection. The representation of stimuli is accumulated in the memory states and the memory is reset at the end of each trial (see Methods). The main advantage of the AuGMEnT network for the computational neuroscience community resides in the biological plausibility of its learning algorithm.

Notably, the AuGMEnT network uses a memory-augmented version of a biologically plausible learning rule (Roelfsema and van Ooyen, 2005) mimicking backpropagation (BP). Learning is the result of the joint action of two factors, neuromodulation and attentional feedback, both influencing synaptic plasticity. The former is a global reward-related signal that is released homogeneously across the network to inform each synapse of the reward prediction error after response selection (Schultz et al., 1993, 1997; Waelti et al., 2001). Neuromodulators such as dopamine influence synaptic plasticity (Yagishita et al., 2014; Brzosko et al., 2015, 2017; He et al., 2015; Frémaux and Gerstner, 2016). The novelty of AuGMEnT compared to three-factor rules (Xie and Seung, 2004; Legenstein et al., 2008; Vasilaki et al., 2009; Frémaux and Gerstner, 2016) is to add an attentional feedback system in order to keep track of the synaptic connections that cooperated for the selection of the winning action and overcome the so-called structural credit assignment problem (Roelfsema and van Ooyen, 2005; Rombouts et al., 2015). AuGMEnT includes a memory system, where units accumulate activity across several stimuli in order to solve temporal credit assignment tasks involving delayed reward delivery (Sutton, 1984; Okano et al., 2000). The attentional feedback mechanism in AuGMEnT works with: (a) synaptic eligibility traces that decay slowly over time, and (b) non-decaying neuronal traces that store the history of stimuli presented to the network up to the current time (Rombouts et al., 2015). The AuGMEnT network solves the Saccade-AntiSaccade task (Rombouts et al., 2015), which is equivalent to a temporal XOR task (Abbott et al., 2016) (see Supplementary Material A).

However, in the case of more complex tasks with long trials and multiple stimuli, like 12AX (O'Reilly and Frank, 2006) depicted in **Figure 1A** and explained in detail in section 3.2, we find that the accumulation of information in AuGMEnT leads to a loss in performance. Hence, we ask the question whether a modified AuGMEnT model would lead to a broader applicability of attention-gated reinforcement learning. We propose a variant of the AuGMEnT network, named hybrid AuGMEnT, that introduces a range of timescales of forgetting or leakage in the memory dynamics to overcome this kind of learning limitation. We employ memory units with different decay constants so that they work on different temporal scales, while the network learns to weight their usage based on the requirements of the specific task. In our simulations, we employed just two subgroups of cells in the memory, where one half of the memory is non-leaky and the other is leaky with a uniform decay time constant; however, more generally, the hybrid AuGMEnT architecture may contain several subgroups with distinct leakage behaviors.

The paper is structured as follows. Section 2 presents the architectural and mathematical details of hybrid AuGMEnT. Section 3 describes the simulation results of the hybrid AuGMEnT network, the standard AuGMEnT network and a fully leaky control network, on two cognitive tasks, a non-hierarchical task involving sequence prediction (Cui et al., 2015) and a hierarchical task 12AX (O'Reilly and Frank, 2006). Finally, in section 4 we discuss our main achievements in comparison with state-of-the-art models and present possible future developments of the work.

2. METHODS

2.1. Hybrid AuGMEnT: Network Architecture and Operation

The network controls an agent which, in each time step t, receives a reward in response to the previous action, processes the next stimulus, and takes the next action (**Figure 1B**). In each time step, we distinguish two phases, called the feedforward pass and feedback pass (**Figure 1C**).

2.1.1. Feedforward Pass: Stimulus to Action Selection

In AuGMEnT (Rombouts et al., 2015), information is processed through a network with three layers, as shown in the left panel of **Figure 1C**. Each unit of the output layer corresponds to an action. There are two pathways into the output layer: the regular R branch and the memory M branch.

The regular branch is a standard feedforward network with one hidden layer. The current stimulus $s_i^R(t)$, indexed by unit index $i = 1, \dots, S$ is connected to the hidden units (called regular units) indexed by j, via a set of modifiable synaptic weights v_{ji}^R yielding activity y_j^R:

$$y_j^R(t) = \sigma\left(h_j^R\right), \quad h_j^R = \sum_i v_{ji}^R s_i^R(t), \tag{1}$$

where σ is the sigmoidal function $\sigma(x) = (1 + \exp(-x))^{-1}$. Input units are one-hot binary with values $S_i \in \{0, 1\}$ (equal to 1 if stimulus i is currently presented, 0 otherwise).

The memory branch is driven by *transitions* between stimuli, instead of the stimuli themselves. The sensory input of the memory branch consists of a set of $2S$ transient units, i.e., S ON units $s_l^+ \in \{0, 1\}$, $l = 1, 2, ..., S$, that encode the onset of each stimulus, and S OFF units $s_l^- \in \{0, 1\}$ that encode the offset:

$$\begin{aligned} s_l^+(t) &= [s_l(t) - s_l(t-1)]_+ \\ s_l^-(t) &= [s_l(t-1) - s_l(t)]_+, \end{aligned} \tag{2}$$

where the brackets signify rectification. In the following, we denote the input into the memory branch with a variable s_i^M defined as the concatenation of these ON and OFF units:

$$s_i^M(t) = \begin{cases} s_i^+(t), & \text{if } i \le S \\ s_{i-S}^-(t), & \text{if } i > S, \end{cases} \tag{3}$$

The memory units in the next layer have to maintain task-relevant information through time. The transient input is transmitted via the synaptic connections v_{ji}^M to the memory layer, where it is accumulated in the states:

$$h_j^M(t) = \varphi_j h_j^M(t-1) + \sum_i v_{ji}^M s_i^M(t). \tag{4}$$

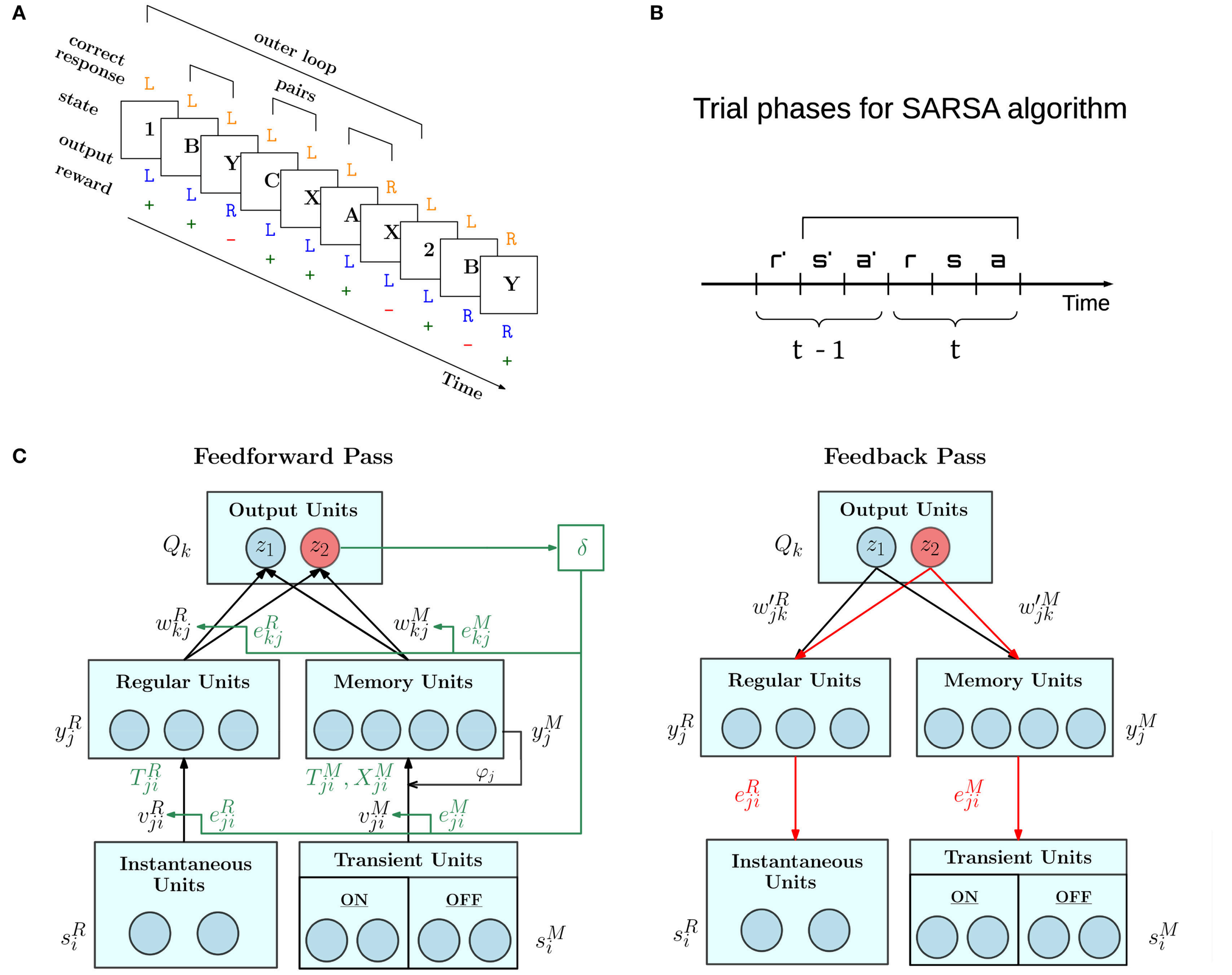

FIGURE 1 | Overview of AuGMEnT network operation. **(A)** Example of trials in the 12AX task. Task symbols appear sequentially on a screen organized in outer loops, which start with a digit, either 1 or 2, followed by a random number of letter pairs (e.g., B-Y, C-X and A-X). On the presentation of each symbol, the agent must choose a Target (R) or Non-Target (L) response. If the chosen and correct responses match, the agent receives a positive reward (+), otherwise it gets a negative reward (–). Figure is adapted from Figure 1 of O'Reilly and Frank (2006). **(B)** AuGMEnT operates in discrete time steps each comprising the reception of reward (r), input of state or stimulus (s) and action taken (a). It implements the State-Action-Reward-State-Action (SARSA, in figure s'a'rsa) reinforcement learning algorithm (Sutton and Barto, 2018). In time step *t*, reward r is obtained for the previous action a' taken in time step *t* – 1. The network weights are updated once the next action a is chosen. **(C)** The AuGMEnT network is structured in three layers with different types of units. Each iteration of the learning process consists of a feedforward pass (left) and a feedback pass (right). In the feedforward pass (black lines and text), sensory information about the current stimulus in the bottom layer, is fed to regular units without memory (left branch) and units with memory (right branch) in the middle layer, whose activities in turn are weighted to compute the *Q*-values in the top layer. Based on the *Q*-values, the current action is selected (e.g., red z_2). The reward obtained for the previous action is used to compute the TD error δ (green), which modifies the connection weights, that contributed to the selection of the previous action, in proportion to their eligibility traces (green lines and text). After this, temporal eligibility traces, synaptic traces and tags (in green) on the connections are updated to reflect the correlations between the current pre and post activities. Then, in the feedback pass, spatial eligibility traces (in red) are updated, attention-gated by the current action (e.g., red z_2), via feedback weights.

We introduce the factor $\varphi_j \in [0, 1]$ here, as an extension to the standard AuGMEnT (Rombouts et al., 2015), to incorporate decay of the memory state h^M_j over time. Setting $\varphi_j \equiv 1$ for all j, we obtain non-leaky memory dynamics as in the original AuGMEnT network (Rombouts et al., 2015) (**Figure 2**, left panel). In our hybrid AuGMEnT network, each memory cell or subgroup of memory cells may be assigned different leak co-efficients φ_j (**Figure 2**, right panel). In this way, the memory is composed of subpopulations of neurons that cooperate in different ways to solve a task, allowing at the same time long-time maintenance and fast decay of information in memory. In contrast to the forget gate of Long Short-Term Memory (Hochreiter and Schmidhuber, 1997) or Gated Recurrent Unit (Cho et al., 2014), our memory leak co-efficient is not trained and gated, but fixed.

The memory state h^M_j leads to the activation of a memory unit:

$$y^M_j(t) = \sigma\left(h^M_j(t)\right). \tag{5}$$

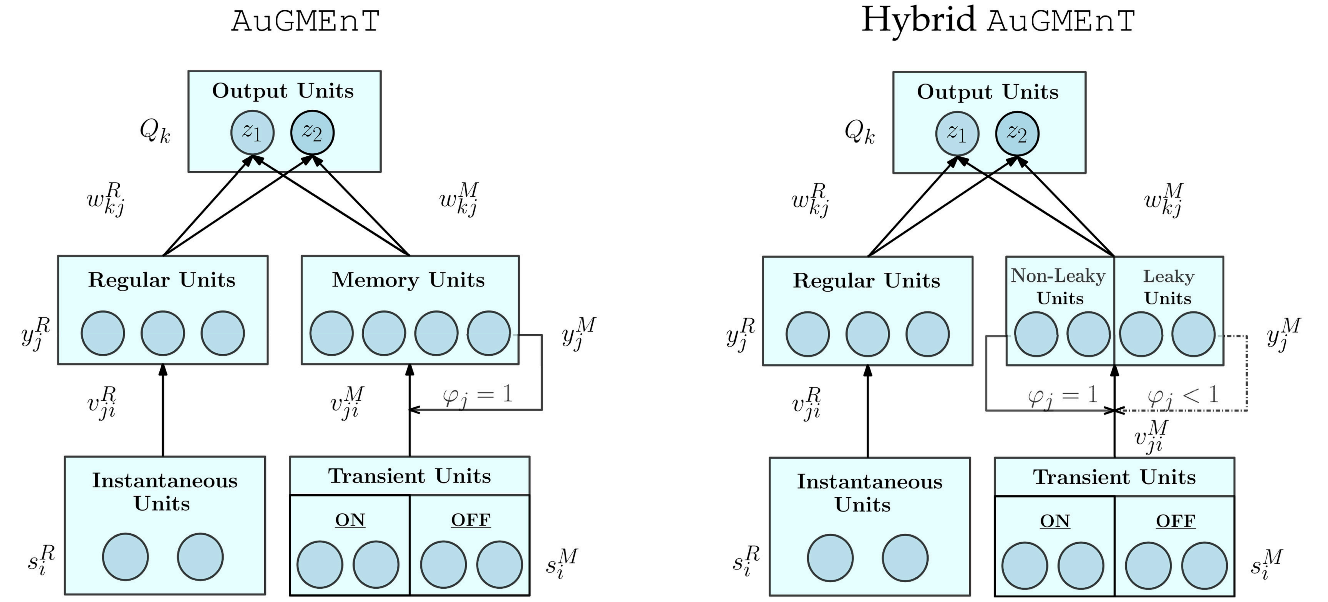

FIGURE 2 | Architectures of standard AuGMEnT and hybrid AuGMEnT networks. The difference between the networks consist in their memory dynamics: the memory layer of standard AuGMEnT (left) has only conservative units with $\varphi_j \equiv 1$, while hybrid AuGMEnT (right) possesses a memory composed of both leaky $\varphi_j < 1$ and non-leaky $\varphi_j = 1$ units.

The states of the memory units are reset to 0 at the end of each trial.

Both branches converge onto the output layer. The activity of an output unit with index k approximates the Q-value of action $a = k$ given the input $\mathbf{s} \equiv [s_i]$, denoted as $Q^{\mathbf{s},a}(t)$. Q-values are formally defined as the future expected discounted reward conditioned on stimulus $\mathbf{s}(t)$ and action $a(t)$:

$$Q^{\mathbf{s},a}(t) = E\left[\sum_{\tau=0}^{\infty} \gamma^{\tau} r_{t+\tau+1} \,\middle|\, \mathbf{s} = \mathbf{s}(t),\ a = a(t)\right], \tag{6}$$

where $\gamma \in [0, 1]$ is a discount factor. Numerically, the vector $\mathbf{Q}$ that approximates the Q-values is obtained by combining linearly the hidden states from the regular and the memory branches, with synaptic weights w^R_{kj} and w^M_{kj}:

$$Q_k(t) = \sum_j w^R_{kj} y^R_j(t) + \sum_j w^M_{kj} y^M_j(t). \tag{7}$$

Finally, the Q-values of the different actions participate in an ϵ-greedy winner-take-all competition (Rombouts et al., 2015) to select the response of the network. With probability $1 - \epsilon$, the next action $a(t)$ is the one with the maximal Q-value:

$$a(t) = \mathrm{argmax}_k Q_k(t). \tag{8}$$

With probability ϵ, a stochastic policy is chosen with probability to take action a given by:

$$p_a = \frac{\exp(g(t)Q_a)}{\sum_k \exp(g(t)Q_k)} \tag{9}$$

where $g(t)$ is a weight function defined as $g(t) = 1 + \frac{m}{\pi}\arctan(\frac{t}{t^*})$, that gradually increases in time with respect to a task-specific, fixed time scale t^* and a scaling factor m. This action selection policy is the same as that used in the original AuGMEnT (Rombouts et al., 2015), except for the weight function $g(t)$ that we introduced, since over time this emphasizes the action with maximal Q-value, improving prediction stability. The time scale parameter t^* and the m factor were manually tuned to optimize convergence time. The choice of the action selection policy and these parameters is further discussed in Supplementary Material B.

2.1.2. After Feedforward Pass: Reward-Based Update of Weights, and Correlation-Based Update of Eligibility Traces, Synaptic Traces, and Tags

AuGMEnT follows the SARSA updating scheme and updates the Q-values for the previous action a' taken at time $t - 1$, once the action a at time t is known (see **Figure 1B**). Q-values depend on the weights via Equation (7). The temporal difference (TD) error is defined as (Wiering and Schmidhuber, 1998; Sutton and Barto, 2018):

$$\delta(t) = \big(r(t) + \gamma Q_a(t)\big) - Q_{a'}(t-1), \tag{10}$$

where a is the action chosen at current time t, and $r(t)$ is the reward obtained for the action a' taken at time $t-1$. The TD error $\delta(t)$ acts as a global reinforcement signal to modify the weights of all connections as

$$\begin{aligned} v^{R,M}_{ji}(t+1) &= v^{R,M}_{ji}(t) + \beta e^{R,M}_{ji}(t)\delta(t), \\ w^{R,M}_{kj}(t+1) &= w^{R,M}_{kj}(t) + \beta e^{R,M}_{kj}(t)\delta(t), \end{aligned} \tag{11}$$

where β is a learning rate and $e_{ji}^{R,M}$ and $e_{kj}^{R,M}$ are synaptic eligibility traces, defined below; see **Figure 1C**. Superscript R or M denotes the regular or memory branch respectively. We use the same symbol $e^{R,M}$ for eligibility traces at the input-to-hidden (*i* to *j*) and hidden-to-output (*j* to *k*) synapses, even though these are different, with the appropriate one clear from context and the convention for indices.

After the update of weights, a synapse from neuron *j* in the hidden layer to neuron *k* in the output layer updates its temporal eligibility trace

$$\begin{aligned} e_{kj}^{R}(t+1) &= y_j^R(t) z_k(t) + (1-\alpha) e_{kj}^{R}(t), \\ e_{kj}^{M}(t+1) &= y_j^M(t) z_k(t) + (1-\alpha) e_{kj}^{M}(t), \end{aligned} \tag{12}$$

where $\alpha \in [0,1]$ is a decay parameter, z_k is a binary one-hot variable that indicates the winning action (equal to 1 if action *k* has been selected, 0 otherwise).

Similarly, a synapse from neuron *i* in the input layer to neuron *j* in the hidden layer sets momentary tags $T_{ji}^{R,M}$ as:

$$\begin{aligned} T_{ji}^{R}(t) &= s_i^R(t)\, \sigma'(h_j^R(t)), \\ T_{ji}^{M}(t) &= X_{ji}^M(t)\, \sigma'(h_j^M(t)), \end{aligned} \tag{13}$$

where $\sigma'(h_j^{R,M})$ is a non-linear function of the input potential, defined as the derivative of the gain function σ, and X_{ji}^M is a synaptic trace (Pfister and Gerstner, 2006; Morrison et al., 2008) defined as follows:

$$X_{ji}^M(t) = \varphi_j\, X_{ji}^M(t-1) + s_i^M(t). \tag{14}$$

Note that the tag $T_{ji}^{R,M}$ has no memory beyond one time step, i.e., it is set anew at each time step. Nevertheless, since X_{ji}^M depends on previous times, the tag T_{ji}^M of memory units can link across time steps. Since activities $y_j^{R,M}$, z_k, $s_i^{R,M}$ and input potentials $h_j^{R,M}$ are quantities available at the synapse, a biological synapse can implement the updates of eligibility traces and tags locally. We emphasize that both eligibility traces and tags can be interpreted as 'Hebbian' correlation detectors.

In the original AuGMEnT model (Rombouts et al., 2015), all eligibility traces and tags were said to be updated in the feedback pass. Here, without changing the algorithm itself, we have conceptually shifted the update of those traces and tags that depend on the correlations of the activities, to the last step of the feedforward pass. Just as in a standard feedforward network with backpropagation of error, we rely on activities during the feedforward pass to calculate the output; therefore the algorithmic update of the weights (Roelfsema and van Ooyen, 2005; Rombouts et al., 2015) has to also rely on these feedforward activities. During the feedback pass activities of the same neurons could in principle change due to attentional gating (Moore and Armstrong, 2003; Roelfsema et al., 2010) or other feedback input. Since feedback input influences the neuronal state (Larkum et al., 1999; Larkum, 2013; Urbanczik and Senn, 2014) the activities in this second phase are different and do not carry the same information as in the feedforward phase. Thus, to increase consistency between biology and algorithm, we evaluate the correlations in the feedforward phase. An alternative could be to use multicompartmental neurons together with the assumption that feedback input arrives at distal dendrites that are only weakly coupled to proximal dendrites where most feedforward inputs arrive (Guerguiev et al., 2017) so that the state of the compartment where feedforward input arrives is only marginally influenced by feedback.

2.1.3. Feedback Pass: Attention-Gated Update of Eligibility Traces

After action selection and the updates of weights, tags, and temporal eligibility traces in the feedforward pass, the synapses that contributed to the currently selected action update their spatial eligibility traces in an attentional feedback step. For the synapses from the input to the hidden layer, the tag $T_{ji}^{R,M}$ from Equation (13) is combined with a spatial eligibility $\sum_k w'^{R,M}_{jk} z_k$ which can be interpreted as an attentional feedback signal (Rombouts et al., 2015).

$$\begin{aligned} e_{ji}^{R}(t+1) &= T_{ji}^{R} \sum_k w'^{R}_{jk} z_k + (1-\alpha) e_{ji}^{R}(t), \\ e_{ji}^{M}(t+1) &= T_{ji}^{M} \sum_k w'^{M}_{jk} z_k + (1-\alpha) e_{ji}^{M}(t), \end{aligned} \tag{15}$$

where feedback weights from the output layer to the hidden layer have been denoted as w'_{jk} and $z_k \in \{0,1\}$ is the value of output unit *k* [one-hot response vector as defined for Equation (12)].

It must be noted that the feedback synapses $w'^{R,M}_{jk}$ follow the same update rule as their feedforward partner $w_{kj}^{R,M}$. Therefore, even if the initializations of the feedforward and feedback weights are different, their strengths become similar during learning.

2.2. Deriving the Learning Rule Via Gradient Descent

For networks with one hidden layer and one-hot coding in the output, attentional feedback is equivalent to backpropagation (Roelfsema and van Ooyen, 2005; Rombouts et al., 2015). Moreover, we now show that the equations for eligibility traces, synaptic traces, and tags, along with the weight update equations reduce a TD-error-based loss function *E*:

$$E = \frac{1}{2}\left(\delta(t)\right)^2, \tag{16}$$

even in the presence of a decay factor $\varphi < 1$. Here, we specifically discuss the case of the tagging Equations (13) and (15) and the update rule (11) associated with the weight v_{ji}^M from sensory input into memory, as these equations contain the memory decay factor φ_j. Analogous update rules for weights v_{ji}^R, w_{kj}^M and w_{kj}^R, in the hybrid AuGMEnT model are identical to existing results (Rombouts et al., 2015), and are omitted here.

Proof: We want to show that

$$\Delta v_{ji}^{M} = \beta\, e_{ji}^{M}\, \delta_t \propto -\frac{\partial E}{\partial v_{ji}^{M}} \tag{17}$$

For simplicity, here we prove (17) for full temporal decay of the eligibility trace e_{ji}^{M} i.e., $\alpha = 1$, corresponding to TD(0) as $\alpha = 1 - \gamma\lambda$, so that

$$e_{ji}^{M} = T_{ji}^{M} \sum_k w'^{M}_{jk} z_k = T_{ji}^{M}\, w'^{M}_{ja'}$$

where a' is the selected action at time $t-1$. The novel aspect of the proof is the presence of a memory decay factor φ_j.

We first observe that the right-hand side of Equation (17) can be rewritten as:

$$-\frac{\partial E}{\partial v_{ji}^{M}} = -\frac{\partial E}{\partial Q_{a'}}\,\frac{\partial Q_{a'}}{\partial v_{ji}^{M}} = \delta_t\,\frac{\partial Q_{a'}}{\partial v_{ji}^{M}}$$

Thus, it remains to show that $\dfrac{\partial Q_{a'}}{\partial v_{ji}^{M}} = e_{ji}^{M}$.

Similarly to the approach used in backpropagation, we now apply the chain rule and we focus on each term separately:

$$\frac{\partial Q_{a'}}{\partial v_{ji}^{M}} = \frac{\partial Q_{a'}}{\partial y_j^{M}}\,\frac{\partial y_j^{M}}{\partial h_j^{M}}\,\frac{\partial h_j^{M}}{\partial v_{ji}^{M}}$$

From Equations (5) and (7), we immediately have that:

$$\frac{\partial y_j^{M}}{\partial h_j^{M}} = \sigma'(h_j^{M}) \qquad \frac{\partial Q_{a'}}{\partial y_j^{M}} = w_{a'j}^{M}$$

We note that, in the feedback step the weight $w_{a'j}^{M}$ is replaced by its feedback counterpart $w'^{M}_{ja'}$. As discussed above, this is a valid approximation because feedforward and feedback weights become similar during learning.

Finally, for the term $\partial h_j^{M}/\partial v_{ji}^{M}$ starting from Equation (4) we can write:

$$\begin{aligned} h_j^{M}(t) &= \sum_i v_{ji}^{M}(t)\, s_i^{M}(t) + \sum_{\tau = t_0}^{t-1} \sum_i \varphi_j^{t-\tau}\, v_{ji}^{M}(\tau)\, s_i^{M}(\tau) \\ &\approx \sum_i v_{ji}^{M}(t) \sum_{\tau = t_0}^{t} \varphi_j^{t-\tau}\, s_i^{M}(\tau) \end{aligned}$$

where t_0 indicates the starting time of the trial and last approximation derives from the assumption of slow learning dynamics, i.e., $v_{ij}^{M}(\tau) = v_{ij}^{M}(t)$ for $t_0 \leq \tau < t$. As a consequence, we have:

$$\frac{\partial h_j^{M}(t-1)}{\partial v_{ji}^{M}(t-1)} \approx \sum_{\tau = t_0}^{t-1} \varphi_j^{t-\tau+1}\, s_i^{M}(\tau) = X_{ji}^{M}(t-1)$$

In conclusion, we combine the different terms and we obtain the desired result:

$$\Delta v_{ji}^{M} \propto \delta_t\, X_{ji}^{M}\, \sigma'(h_j^{M})\, w'^{M}_{ja'} = \delta_t\, T_{ji}^{M}\, w'^{M}_{ja'} = \delta_t\, e_{ji}^{M}.$$

Thus, if the decay factor φ_j of the synaptic trace X_{ji}^{M} in Equation (14) matches the decay factor of the memory unit in Equation (4), then the update rule for eligibility traces, synaptic traces and tags, and weights leads to a reduction of the TD error. However, instead of matching the two φ-s, we could merely use a unique decay factor in the input without affecting the biological plausibility of the algorithm (see Supplementary Material C). Nevertheless, we maintained the original formulation for sake of comparison with the reference AuGMEnT network.

2.3. Simulation and Tasks

All simulation scripts were written in python (https://www.python.org), with plots rendered using the matplotlib module (http://matplotlib.org). These simulation and plotting scripts are available online at https://github.com/martin592/hybrid_AuGMEnT.

We used the parameters listed in **Table 1** for our simulations. Further, for the Hybrid AuGMEnT network, we set $\varphi_j = 1$ for the first half of the memory cells and $\varphi_j = 0.7$ for the second half. To compare with the standard AuGMEnT network (Rombouts et al., 2015), we set $\varphi_j \equiv 1$ for all j, while for a leaky control network we set $\varphi_j \equiv 0.7$ for all j. In general, the leak co-efficients can be tuned to adapt the overall memory dynamics to the specific task at hand, but we did not optimize the parameter φ.

3. RESULTS

AuGMEnT (Rombouts et al., 2015) includes a differentiable memory system and is trained in an RL framework with learning rules based on the joint effect of synaptic tagging, attentional feedback and neuromodulation (see Methods). Here, we study our variant of AuGMEnT, named hybrid AuGMEnT, that has an additional leak factor in a subset of memory units, and compare it to the original AuGMEnT as well as to a control network with uniform leaky memory units.

As a first step, we validated our implementations of standard and hybrid AuGMEnT networks on the Saccade-AntiSaccade (S-AS) task, used in the reference paper (Rombouts et al., 2015) (Supplementary Material D). We next simulated the networks on two other cognitive tasks with different structure and memory demands: the sequence prediction task (Cui et al., 2015) and the 12AX task (O'Reilly and Frank, 2006). In the former, the

TABLE 1 | Parameters for the AuGMEnT network.

Parameter	Value
β : Learning parameter	0.15
λ : Eligibility persistence	0.15
γ : Discount factor	0.9
α : Eligibility decay rate	$1 - \gamma\lambda$
ϵ : Exploration rate	0.025
t^* : Softmax time scale	2000 trials
m : Softmax scaling factor	10

Test dataset consists of 1, 000 random sequences, while averages are computed over 100 simulations.

agent has to predict the final letter of a sequence depending only on its starting letter, while in the latter, the agent has to identify target pairs inside a sequence of hierarchical symbols. The S-AS task maps to a temporal XOR task (Abbott et al., 2016); thus the hidden layer is essential for the task (Minsky and Papert, 1969; Rumelhart et al., 1985). The 12AX also resembles an XOR structure, but is more complex due to an additional dimension and distractors in the inner loop (Figure S1). The complexity of the sequence prediction task is less compared to the 12AX task, and can be effectively solved by `AuGMEnT`. We will show that hybrid `AuGMEnT` performs well on both cognitive tasks, whereas standard `AuGMEnT` fails on the 12AX task. The parameters involving the architecture of the networks on each task are reported in **Table 2**. We now discuss each of the tasks in more detail.

3.1. Task 1: Sequence Prediction

In the sequence prediction task (Cui et al., 2015), letters appear sequentially on a screen and at the end of each trial the agent has to correctly predict the last letter. Each sequence starts either with an `A` or with an `X`, which is followed by a fixed sequence of letters (e.g., `B-C-D-E`). The trial ends with the prediction of the final letter, which depends on the initial cue: if the sequence started with `A`, then the final letter has to be a `Z`; if the initial cue was an `X`, then the final letter has to be a `Y`. In case of correct prediction the agent receives a reward of 1 unit, otherwise it is punished with a negative reward of -1. A scheme of the task is presented in **Figure 3** for sequences of four letters.

The network has to learn the task for a given sequence length, kept fixed throughout training. The agent must learn to maintain the initial cue of the sequence in memory until the end of the trial, to solve the task. At the same time, the agent has to learn to neglect the information coming from the intermediate cues (called distractors). Thus the difficulty of the task is correlated with the length of the sequence.

TABLE 2 | Network architecture parameters for the simulations.

Network parameter	Sequence prediction task (L = sequence length)	12AX task
S : Number of sensory units	$L-1$	8
R : Number of regular units	3	10
M : Number of memory units	8	20
A : Number of activity units	2	2

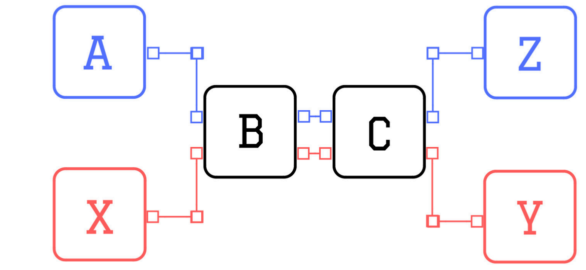

FIGURE 3 | Scheme of the sequence prediction task. Scheme of sequence prediction trials with sequence length equal to 4 (i.e., 2 distractors): the two possible sequences are: `A-B-C-Z` (blue) or `X-B-C-Y` (red) .

We studied the performance of the `AuGMEnT` network (Rombouts et al., 2015) and our hybrid variant on the sequence prediction task. The mean trend of the TD loss function defined in Equation (16) (**Figure 4A**) shows that both models converge in a few hundred iterations. As a control, we also simulated a variant in which all memory units were leaky. We observed that hybrid and standard `AuGMEnT` networks are more efficient than the purely leaky control. This is not surprising because the key point in the sequence prediction task consists in maintaining the initial stimulus in memory—which is simpler with non-leaky memory units than with leaky ones. We notice that the hybrid model has a behavior similar to `AuGMEnT`.

We also analyzed the effect of the temporal length of the sequences on the network performance, by varying the number of distractors (i.e., the intermediate letters) per sequence (**Figure 4B**). For each sequence length, the network was retrained ab initio. We required 100 consecutive correct predictions as the criterion for convergence. We ran 100 simulations starting with different initializations for each sequence length and averaged the convergence time. Again, `AuGMEnT` and Hybrid `AuGMEnT` show good learning performance, maintaining an average of about 250 trials to convergence for sequences containing up to 10 distractors, whereas a network with purely leaky units is much slower to converge.

The leaky dynamics are not helpful for the sequence prediction task, because the intermediate cues are not relevant for the final model performance. Therefore, we expect the learning rule to suppress the weight values in the $\mathbf{V}^M$ matrix for distractors, and increase those of the initial `A`/`X` letter. This is confirmed by the structure of the weight matrix from transient units to memory units shown after convergence (**Figure 5**), in simulations of the sequence prediction task on sequences with $D = 3$ or $D = 8$ distractors. The weight values are highest in absolute value for connections from transient units representing letters `A` and `X`, for both the ON (+) and OFF (−) type. We emphasize that Hybrid `AuGMEnT` employs mainly the conservative (non-leaky) memory units (`M1−C` and `M2−C`) rather than the leaky ones (`M1−L` and `M2−L`) to solve the task, showing that the learning rule is able to focus updates on the connections that are most relevant for the specific task.

To confirm the better performance of the network using conservative units over leaky ones, we tested the networks on a modified task never seen during training. Specifically, the test sequences were one letter longer than training sequences and the distractors were not anymore in alphabetical order but were sampled uniformly. For instance, if the network was trained with distractors `B-C-D-E`, the test sequences may be `A-C-D-C-B-E` or `X-D-B-B-B-E` (the last letter remains fixed because it is the *go* signal for the network). In this way, the network experiences different forms of sequence alterations (e.g., prolongation, inversion and skipping of distractors) and we can test how the network generalizes on new versions of the problems.

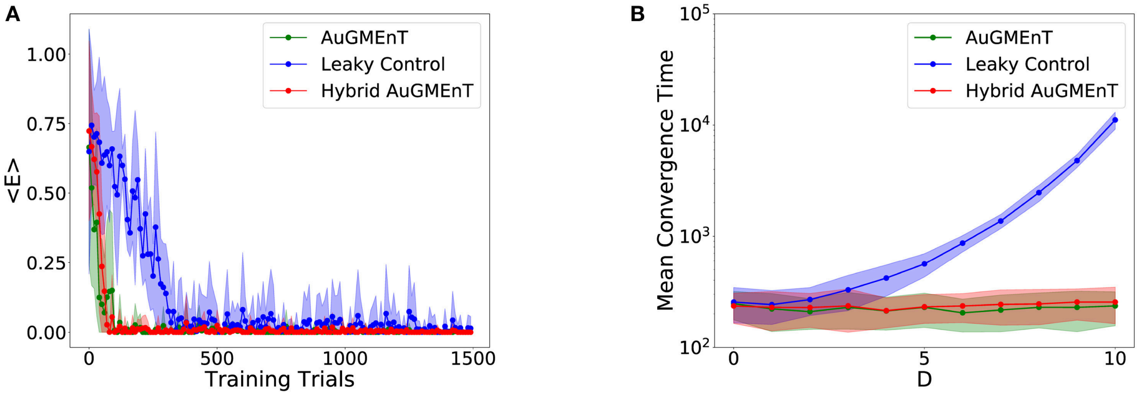

FIGURE 4 | Convergence in the sequence prediction task. **(A)** Time course of error of the models on the sequence prediction task with sequences of five letters (three distractors): the mean squared TD error decays to zero for all networks, but the leaky control network (blue) is much slower than `AuGMEnT` (green) and Hybrid `AuGMEnT` (red). **(B)** Convergence time of the `AuGMEnT` network and its variants on the sequence prediction task with increasing number *D* of distractors, i.e., intermediate cues before final prediction.

Different versions of `AuGMEnT` are compared in the test phase by observing the mean prediction accuracy over 1,000 test sequences (**Table 3**). The results show again that leaky dynamics penalize the performance on the sequence prediction task; in fact, when `AuGMEnT` includes conservative units (either totally or partially) the mean percentage of correct predictions is higher than 98%, otherwise the accuracy drops down to 85.8% in case of purely leaky control. However, if training is long enough to emphasize more the initial information also on leaky units, then the final test performance improves notably in both cases and the accuracy gap is greatly reduced (100% vs. 99.4%).

3.2. Task 2: 12AX

The 12AX task is a standard cognitive task used to test working memory and diagnose behavioral and cognitive deficits related to memory dysfunctions (O'Reilly and Frank, 2006; Alexander and Brown, 2015). The task involves identifying some target sequences among a group of symbols that appear on a screen.

The general procedure of the task is schematized in **Figure 1A** and details involving the construction of the 12AX dataset are collected in **Table 4**. The set of possible stimuli consists of 8 symbols: two digit cues (`1` and `2`), two context cues (`A` and `B`), two target cues (`X` and `Y`), and two additional distractors, a context distractor `C` and a distractor target `Z`. Each trial (or outer loop) starts with a digit cue and is followed by a random number of context-target pairs, such as `A-X`, `B-X` or `B-Y`. The cues are presented one by one on a screen and the agent has two possible actions for each of them: Target (`R`) and Non-Target (`L`). There are only two valid Target cases: in trials that start with digit `1`, the Target is associated with the target cue `X` if preceded by context `A` (`1-...-A-X`); otherwise, in case of initial digit `2`, the Target occurs if the target cue `Y` comes after context `B` (`2-...-B-Y`). The dots are inserted to stress that the target pair can occur a long time after the digit cue, as happens in the following example sequence: **`1`**`-A-Z-B-Y-C-X-`**`A`**`-`**`X`** (whose sequence of correct responses is `L-L-L-L-L-L-L-L-`**`R`**). The variability in the temporal length of each trial is the main challenge in solving the 12AX task. Moreover, since `1-A-X` and `2-B-Y` are target sequences, whereas `2-A-X` and `1-B-Y` are not, the task can be seen as a generalization of temporal XOR (Figure S1).

The inserted pairs are determined randomly, with a probability of 50% to have pairs `A-X` or `B-Y`. As a result, combined with the probability to have either `1` or `2` as starting digit of the trial, the overall probability to have a target pair is 25%. Since the Target response `R` has to be associated only with an `X` or `Y` stimulus that appears in the correct sequence, the number of Non-Targets `L` is generally much larger, on average 8.96 Non-Targets to 1 Target. We rewarded the correct predictions of a Non-Target with 0.1 and of Targets with 1, and punished wrong predictions with a reward of −1. In effect, we balanced the positive reward approximately equally between Targets and Non-Targets based on their relative frequencies, which aids convergence.

We simulated the Hybrid `AuGMEnT` network, as well as the standard `AuGMEnT` and the leaky control on the 12AX task, in order to see whether in this case the introduction of the leaky dynamics improves learning performance. **Figure 6A** shows the evolution of the mean squared TD error for the three networks. After a sharp descent, all networks converge to an error level that is non-zero, in part from ongoing action exploration and remaining part from inability to learn possibly due to memory interference. Here, hybrid `AuGMEnT` and leaky control saturate at a lower error value than base `AuGMEnT`. This difference can be attributed mainly to the errors in responding to the Target cues (**Figure 6B**), whereas Non-Target cues are well learned (**Figure 6C**). Note that, since a response is required at each step in the 12AX task, the error is also computed at each iteration—including for the more frequent and trivial Non-Target predictions—and averaged over 2,000 consecutive predictions. All networks quickly learn to recognize the Non-Target cues (`1`, `2`, `A`, `B`, `C`, `Z` are always Non-Targets) (**Figure 6C**). However, hybrid `AuGMEnT` and leaky control learn the more complex identification of Target patterns within a trial when `X`

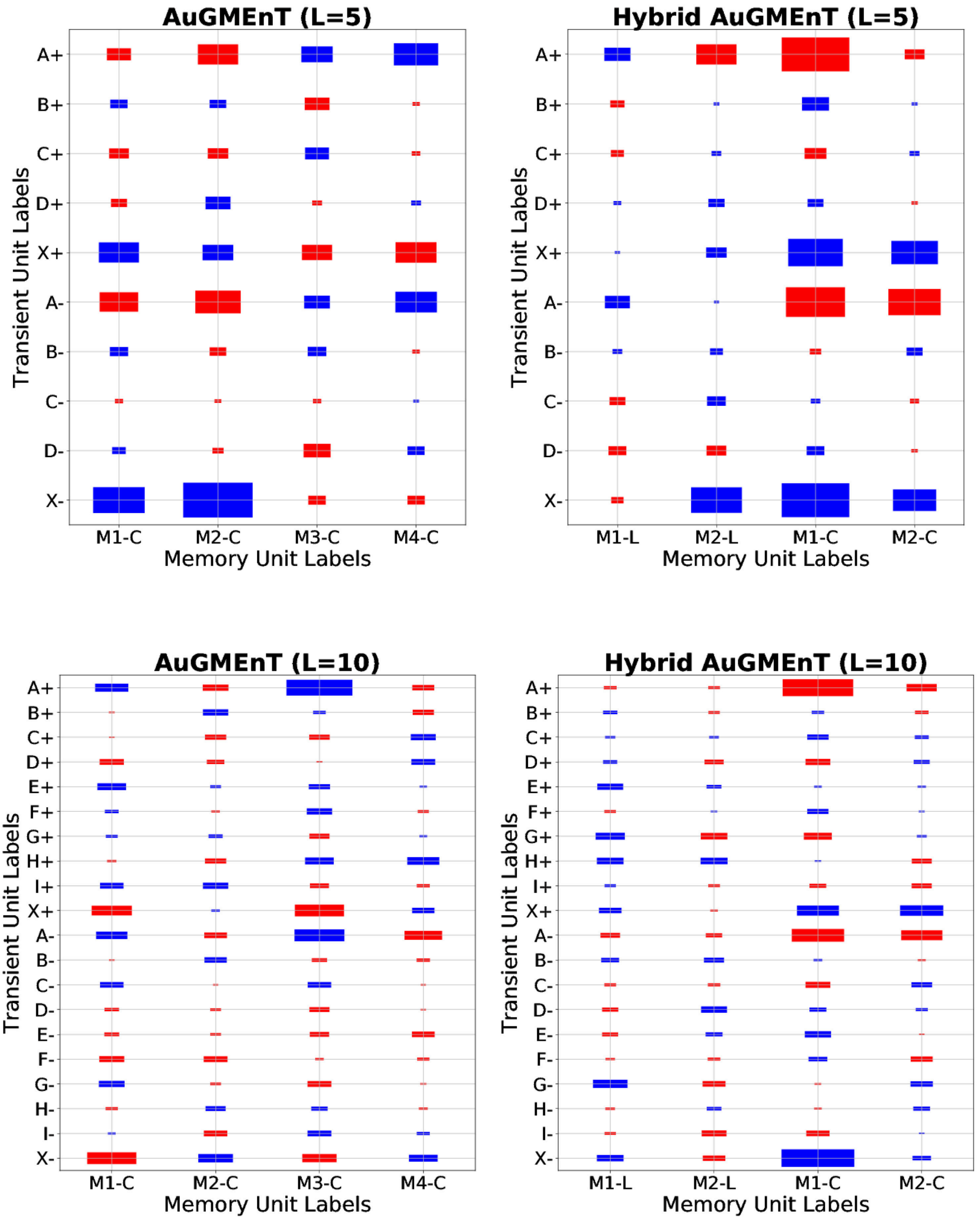

FIGURE 5 | Memory weights of `AuGMEnT` networks in sequence prediction task for $D = 3$ and $D = 8$ distractors. Memory weight matrices (from transient units to memory units) after convergence in a representative simulation, for `AuGMEnT` **(left)** and Hybrid `AuGMEnT` **(right)** networks, on the sequence prediction task with sequences of length five (first row) and 10 (second row). Note that the first two memory units in Hybrid `AuGMEnT` are leaky (`M1-L` and `M2-L`), while the last ones are conservative (`M1-C` and `M2-C`). The size of each rectangle is a linear transformation of the absolute value of the weight, while the color indicates the sign (red: positive value, blue: negative value). Above: Min value = 0.01, Max value = 1.37. Below: Min value = 0.01, Max value = 1.49.

or `Y` are presented to the network, better than base `AuGMEnT` (**Figure 6B**). The gap in the mean squared TD error between hybrid `AuGMEnT` and leaky control versus standard `AuGMEnT` is wider when only potential Target cues are considered in the mean squared TD error as in **Figure 6B**, than when only Non-Targets are considered as in **Figure 6C**.

TABLE 3 | Statistics of different versions of AuGMEnT networks tested on untrained longer-length sequences in the sequence prediction task.

Network	Mean test accuracy	
	After convergence (%)	After 10,000 training sequences (%)
Standard AuGMEnT	98.1	100
Purely Leaky Control	85.8	99.4
Hybrid AuGMEnT	98.3	100

Test dataset consists of 1,000 random sequences, while averages are computed over 100 simulations. The test performance is evaluated both after convergence (i.e., after few hundreds of training sequences) and after training on 10,000 trials.

TABLE 4 | The 12AX task: table of key information.

Task feature	Details
Input	8 possible stimuli: 1,2,A,B,C,X,Y,Z.
Action	Non-Target (L) or Target (R).
Target sequences	1-...-A-X or 2-...-B-Y. Probability of target sequence is 25%.
Training dataset	Maximum number of training trials is 1, 000, 000.
Pairs	Each trial starts with 1 or 2, followed by a random number (between 1 and 4) of pairs chosen from {A-X, A-Y, B-X, B-Y, C-X, C-Y, A-Z, B-Z, C-Z}.

With the convergence criterion of 1,000 consecutive correct predictions (corresponding to ~167 trials) (Alexander and Brown, 2015), standard AuGMEnT network was unable to converge (0% success), over 1,000,000 trials, in any of 100 simulations (**Figure 7**). However, hybrid AuGMEnT and leaky control reached 100% convergence (**Figure 7**), suggesting that leaky memory units are necessary for the 12AX task. The leaky control needs roughly the same time (learning time mean = 30,032.2 trials and s.d. = 11,408.9 trials) to reach convergence criterion as hybrid AuGMEnT (learning time mean = 34,263.6 trials and s.d. = 12,737.3 trials). In line with standard AuGMEnT (Rombouts et al., 2015), the memory was reset after every trial (here every outer loop), and hence the networks were not required to learn digit context switches. In Supplementary Material E, we show that leaky control and to an extent hybrid AuGMEnT also learn to switch between digit contexts, without needing the manual reset of memory. However, for the sake of comparison with the original implementation of AuGMEnT (Rombouts et al., 2015), here we default to the case with memory reset at the end of each outer loop.

Success of learning refers to the fulfillment of the convergence criterion (Alexander and Brown, 2015) and **Figure 7** indicates that hybrid AuGMEnT learns well enough to reach criterion (unlike standard AuGMEnT). However, despite reaching convergence criterion after about 30,000 trials, the network may occasionally make mistakes even at the end of learning after 150, 000 trials, as indicated by the non-zero error in **Figure 7**. Further analysis of this result shows that the remaining errors are mainly due to our ϵ-greedy action selection policy. With a standard ϵ-greedy policy used during a separate test phase with fixed weights, about 94% of trials are successful (Table S2); however, the same network with the same synaptic weights, but a greedy policy during the test phase passes more than 98% of trials. The exact performance numbers depend on how the exploration-exploitation trade-off is implemented (see Supplementary Material B).

In order to understand how the hybrid memory works on the 12AX task, we analyzed the weight structure of the connectivity matrices which belong to the memory branch of the hybrid AuGMEnT network (**Figure 8**). Unlike in the sequence prediction task, here the hybrid network employs both the leaky and the non-leaky memory units. The highest absolute values are found for the weights associated with 1(±) and 2(±) as well as X(±) and Y(±) (e.g., on M4, M9, M17 and M20). All memory units contribute to the definition of the activity *Q*-values (**Figure 8**, right panel) consistent with a distributed representation.

The memory units show an opposing behavior on activation versus on deactivation of Target cues: for instance, if X+ has strong positive weights, then X− shows negative weights (see M14, M17 or M20). In this way, the network tries to reduce the problems of memory interference between subsequent pairs by subtracting from the memory during deactivation, an amount that balances the information stored during the previous activation, effectively erasing the memory. Further, the difference in absolute value between activation and deactivation is higher in the case of the leaky cells, because the deactivation at the next iteration has to remove only a lower amount of information from the memory due to leakage. However, for the digit cues 1 and 2, the weights for activation and deactivation have typically the same sign in order to reinforce the digit signal in memory in two subsequent timesteps (e.g., on M4 and M9). More importantly, we can observe that in the leaky units, the highest weight values are generally associated to the digit information and the other cues are less represented, while in the conservative ones, the target cues are also emphasized in the memory. This means that the role of the leaky units consists mainly in the storage of the digit cue, while the conservative ones are also responsible for the storage of the information coming from the inner loops.

4. DISCUSSION

The conservative dynamics of the memory in standard AuGMEnT can be a limitation for the learning ability of the model, especially in cases of complex tasks with long trials. In fact, even though the 12AX task is less complex compared to more recent RL tasks (Mnih et al., 2015), the standard AuGMEnT network fails to satisfy the convergence criterion. The introduction of the leak factor (Equation 4) in hybrid AuGMEnT leads to a network that solves the 12AX task. Hybrid AuGMEnT also does as well as standard AuGMEnT on the sequence prediction task, while the purely leaky control cannot solve this task in a reasonable time. Thus, hybrid AuGMEnT solves both tasks combining conservative and leaky memory units. Hybrid AuGMEnT can be adapted to different task structures and to different temporal

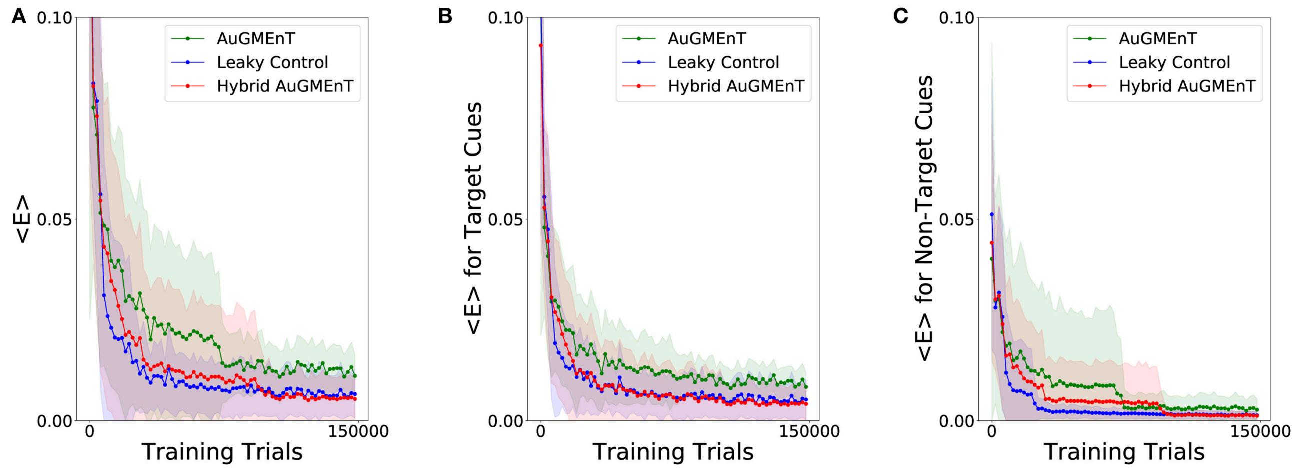

FIGURE 6 | Learning convergence of the AuGMEnT variants in the 12AX task. Minimization of the TD loss function during training on the 12AX task. **(A)** All networks show a good decay of the mean squared TD error, but they seem to converge to a non-zero regime and, in particular, the base AuGMEnT network (green) maintains a higher mean squared TD error level when compared to leaky control (blue) and Hybrid AuGMEnT (red). **(B)** Mean squared TD error associated with only potential Target cues X and Y. **(C)** Mean squared TD error related to only Non-Target cues.

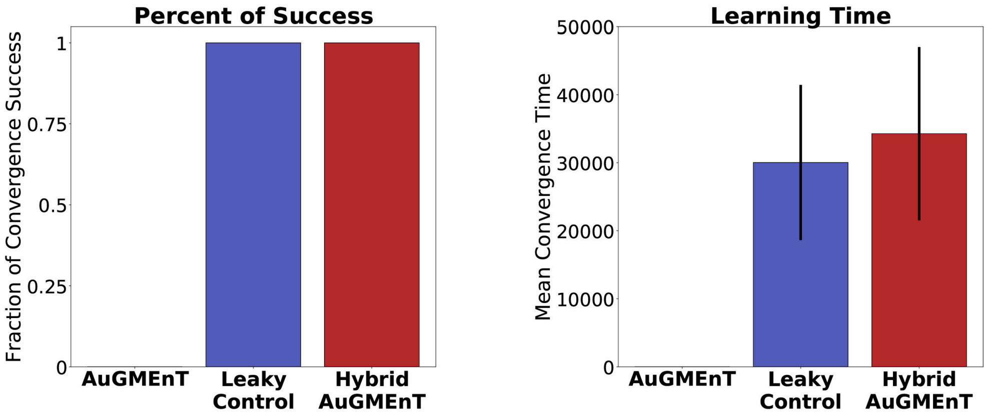

FIGURE 7 | Comparative statistics of the AuGMEnT variants on performance on the 12AX task. Barplot description of the learning behavior of the three networks on the 12AX task according to the convergence criterion given by Alexander and Brown (2015). After 100 simulations, we measured the fraction of times that the model satisfies the convergence condition **(left)** and the average number of training trials needed to meet the convergence criterion **(right)**. Although training dataset consists of 1,000,000 trials, the standard AuGMEnT network never manages to satisfy the convergence criterion, while the leaky (blue) and hybrid (red) models have similar convergence performance with a learning time of about 30,000 trials.

scales by varying the size and the composition of the memory, for example by considering multiple subpopulations of neurons with distinct memory timescales, say in a power law distribution.

A key goal of the computational neuroscience community is to develop neural networks that are at the same time biologically plausible and able to learn complex tasks similar to humans. The embedding of memory is certainly an important step in this direction, because memory plays a central role in human learning and decision making. Our interest in the AuGMEnT network (Rombouts et al., 2015) derives from the biological plausibility of its learning and memory dynamics. In particular, the biological setting of the learning algorithm is based on synaptic tagging, attentional feedback and neuromodulation, providing a possible biological interpretation to backpropagation-like methods. Hybrid AuGMEnT inherits the biological plausibility of standard AuGMEnT. Our proposed memory mechanism is also biologically plausible with synaptic traces decaying at the memory time scale (in addition, see Supplementary Material C).

We have no convergence guarantees for our algorithm and network. While on-policy TD learning methods have convergence guarantees for fully observable Markov Decision Processes (MDPs) (Singh et al., 2000), the 12AX task is a Partially Observable Markov Decision Process (POMDP) (Monahan, 1982). Even though there are no theoretical convergence guarantees for POMDPs, there is various experimental support for solving specific POMDPs with TD learning, either using

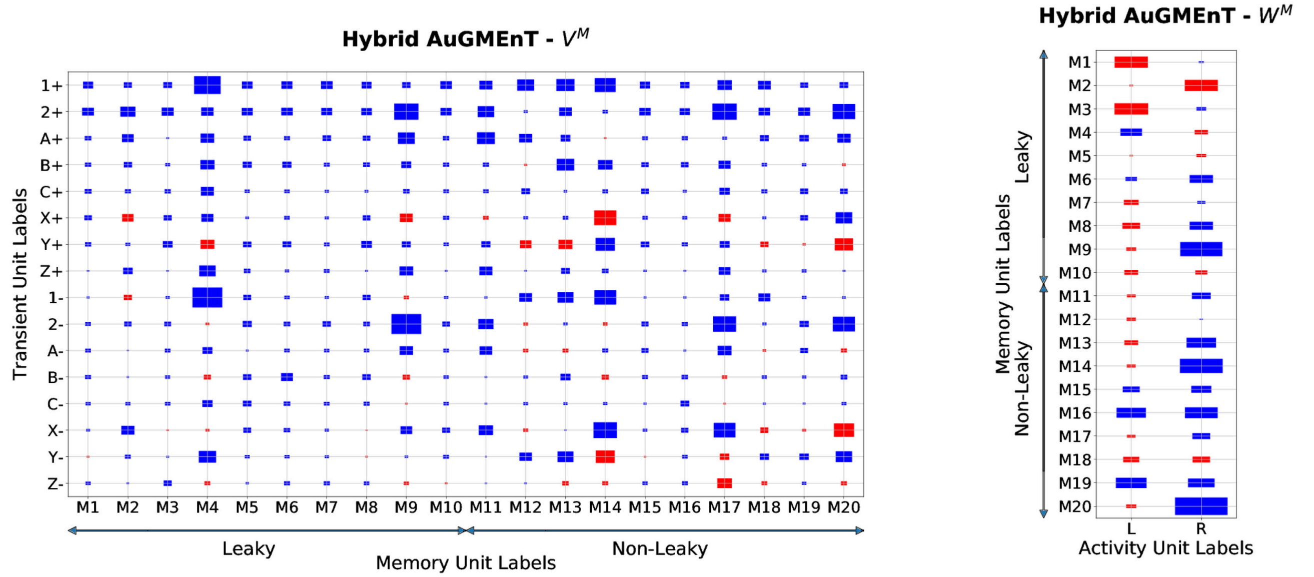

FIGURE 8 | Memory weights of Hybrid AuGMEnT in the 12AX task. Plot of the weight matrices in the memory branch of hybrid AuGMEnT network after convergence in a representative simulation on the 12AX Task. The size of each rectangle is a linear transformation of the absolute value of the weight, while the color indicates the sign (red: positive value, blue: negative value). **Left:** weights from the transient stimulus into the 20 memory units (half leaky, half conservative). Min value = 0.001, Max value = 4.67. **Right:** weights from the memory cells into the output units. Min value = 0.01, Max value = 2.0.

eligibility traces i.e., SARSA(λ) (Loch and Singh, 1998), or by storing observations in memory (Lin and Mitchell, 1993; McCallum, 1993; Todd et al., 2009). The memory serves to hold a history of observations, the right combination of which could represent the latent states of the underlying MDP (Sutton and Barto, 2018). Ideally, the network should learn both when and which observations (and even actions) to store in memory. While our memory does not have time-dependent gating, it does learn to weight stimuli appropriately. With gated memory and an actor-critic algorithm, the 12AX task, as well as some finite-state grammar tasks (Cleeremans and McClelland, 1991), have been learned (Todd et al., 2009). The recently proposed biologically-plausible subtractive-inhibition based gating architecture (Costa et al., 2017) could be incorporated into hybrid AuGMEnT to possibly further enhance its task repertoire.

Even apart from the issue of POMDPs, there is the issue of convergence of TD-learning for MDPs using a neural network to approximate the Q-value function. Here, we have used an on-policy method i.e., SARSA(λ), with the output layer being linear in the weights. There are good convergence properties for on-policy TD learning with linear (in the weights) function approximation (Tsitsiklis and Roy, 1997; Melo et al., 2008). In our network though, we also change the weights from the stimuli to the hidden units, which non-linearly affect the output. Perhaps, this can be imagined as a form of feature learning on the stimuli, these features are then combined linearly at the output (Sutton and Barto, 2018). Further, we showed that our network performs stochastic gradient descent on the squared projected TD error, projected because the network approximation might project the true TD error onto a lower-dimensional subspace (Sutton and Barto, 2018). Stochastic gradient descent on the squared projected TD error has been shown to converge with linear (Sutton et al., 2009) and non-linear (Bhatnagar et al., 2009) function approximation. Thus, even though we neither claim nor show convergence, there exists some partial and indirect theoretical support for convergence in similar architectures.

We now compare hybrid AuGMEnT with other memory-augmented networks, in order to explore different implementations of memory dynamics and possibly take inspiration for further developments on our network.

The Hierarchical Temporal Memory (HTM) network (Cui et al., 2015) presents greater flexibility in sequence learning than AuGMEnT on the simple sequence prediction task. Utilizing a complex column-based architecture and an efficient system of inner inhibitions, the HTM network is able to maintain a dual neural activity, both at column level and at unit level, that allows to have sparse representations of the input and give multi-order predictions using an unsupervised Hebbian-like learning rule. Thus, HTM has high sequence learning ability with the possibility to solve a large variety of sequence tasks, like sequence classification and anomaly detection. Nonetheless, it is unclear how the HTM network can be applied to reward-based learning, in particular to tasks like the 12AX, with variable number of inner loops.

Although the hybrid memory in the AuGMEnT network remarkably improved its convergence performance on the 12AX task, its learning efficiency is still lower than the reference Hierarchical Error Representation model (HER) (Alexander and Brown, 2015, 2016). In fact, in our simulations, hybrid AuGMEnT showed a mean convergence time of 34, 263.6 outer loops, while the average learning time of HER on the same convergence condition is around 750 outer loops. The main reason for this large gap in the learning performance resides in the gating mechanism of HER network that is specifically developed for hierarchical tasks and is used to decide at each iteration whether to store the new input or maintain the previous content in memory. Unlike HER model, the memory in AuGMEnT does not include any gating mechanism, meaning that the network does not learn when to store and recall information but the memory dynamics are entirely developed via standard weight modulation. On the other hand, the HER model is not as biologically plausible as the AuGMEnT network, because, although its hierarchical structure is inspired from the supposed organization of the prefrontal cortex, its learning scheme is artificial and based on standard backpropagation.

In addition, the recent delta-RNN network (Ororbia et al., 2017) presents interesting similarities with hybrid AuGMEnT in employing two timescales, maintaining memory via interpolation of fast and slow changing inner representations. In fact, the approach is similar to what we proposed in hybrid AuGMEnT, where the output of the memory branch is the linear combination of the activity of leaky (changing) and non-leaky (non-changing) units in the hybrid memory. The delta-RNN is a generalization of the gating mechanisms of LSTM and GRU and likely has a better learning ability than hybrid AuGMEnT, but it is less convincing in terms of biological plausibility.

The lack of a memory gating system is a great limitation for AuGMEnT variants, when compared with networks equipped with a gated memory, like HER (Alexander and Brown, 2015, 2016) or LSTM (Hochreiter and Schmidhuber, 1997; Gers et al., 2000), especially on complex tasks with high memory demand. Still, even though it cannot be properly defined as a gating system, the forgetting dynamics introduced in hybrid AuGMEnT has an effect similar to the activity of the forget gates in LSTM or GRU. However, unlike forget gates, the decay coefficients are not learnable and are not input-dependent for each memory cell.

The Hybrid AuGMEnT network could be further enhanced by adding controls on the loading, amount of leakage, and readout on the memory units, similar to input, forget, and output gates in LSTM (Hochreiter and Schmidhuber, 1997; Gers et al., 2000) and GRU (Cho et al., 2014), though only the leakage control may be most important (van der Westhuizen and Lasenby, 2018). To be specific, the value of each gate or control parameter could be set by additional units of the network that serve as a controller. In this way, the loading, leakage and output of the memory units would become stimulus- or even history-dependent. On the other hand, such a control system would make the network more complex and learning of the control variables with error backpropagation would compromise the biological plausibility of the AuGMEnT learning dynamics. However, the recently introduced subLSTM network uses subtractive inhibition in a network of excitatory and inhibitory neurons as a biologically plausible gating mechanism (Costa et al., 2017), while biologically plausible versions of backpropagation are also being developed (Lillicrap et al., 2016; Guerguiev et al., 2017; Baldi et al., 2018).

Alternatively, inspired by the hierarchical architecture of HER (Alexander and Brown, 2015), the memory in AuGMEnT could be divided into multiple levels each with its own memory dynamics: each memory level could be associated with distinct synaptic decay and leaky coefficients, learning rates, or gates, in order to cover different temporal scales and encourage level specialization. Compared with hybrid AuGMEnT, the memory would be differentiated not only in the leaky dynamics, but also in the temporal dynamics associated with attentional feedback and synaptic potentiation. Using this hierarchical structure of the memory requires additional modification of the network architecture: since the input information is separated among the memory levels, we have to introduce a system to aggregate the information. To achieve this, we could either feed the output of the hierarchical memory to the associative layer of the controller branch, or we could define a read gating system that depends on the memory content.

In the past years, the reinforcement learning community has proposed several deep RL networks, like deep Q-networks (Mnih et al., 2015) or the AlphaGo model (Chen, 2016), that combine the learning advantages of deep neural networks with reinforcement learning (Li, 2017). Thus, it may be interesting to consider a deep version of the AuGMEnT network with additional hidden layers of neurons. While conventional error backpropagation in AuGMEnT may not yield plausible synaptic plasticity rules, locality might be retained with alternative backpropagation methods (Lillicrap et al., 2016; Guerguiev et al., 2017; Baldi et al., 2018).

AUTHOR CONTRIBUTIONS

AG, MM, and WG contributed to the conception and design of the study. MM developed and performed the simulations, and wrote the first draft of the manuscript. WG, MM, and AG revised the manuscript, and read and approved the submitted version. MM, AG and WG further revised the manuscript in accordance with the reviewers' comments and read and approved the final version.

ACKNOWLEDGMENTS

We thank Vineet Jain and Johanni Brea for helpful discussions, and Vasiliki Liakoni for comments on the manuscript. Financial support was provided by the European Research Council (Multirules, grant agreement no. 268689), the Swiss National Science Foundation (Sinergia, grant agreement no. CRSII2_147636), and the European Commission Horizon 2020 Framework Program (H2020) (Human Brain Project, grant agreement no. 720270).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fncom.2018.00050/full#supplementary-material

REFERENCES

Abbott, L., De Pasquale, B., and Memmesheimer, R.-M. (2016). Building functional networks of spiking model neurons. *Nat. Neurosci.* 19, 350–355. doi: 10.1038/nn.4241

Alexander, W. H., and Brown, J. W. (2015). Hierarchical error representation: a computational model of anterior cingulate and dorsolateral prefrontal cortex. *Neural Comput.* 27, 2354–2410. doi: 10.1162/NECO_a_00779

Alexander, W. H., and Brown, J. W. (2018). Frontal cortex function as derived from hierarchical predictive coding. *Sci. Rep.* 8:3843. doi: 10.1038/s41598-018-21407-9

Baldi, P., Sadowski, P., and Lu, Z. (2018). Learning in the machine: random backpropagation and the deep learning channel. *Artif. Intell.* 260, 1–35. doi: 10.1016/j.artint.2018.03.003

Barak, O., and Tsodyks, M. (2014). Working models of working memory. *Curr. Opin. Neurobiol.* 25, 20–24. doi: 10.1016/j.conb.2013.10.008

Bhatnagar, S., Precup, D., Silver, D., Sutton, R. S., Maei, H. R., and Szepesvri, C. (2009). " Convergent temporal-difference learning with arbitrary smooth function approximation," in *Advances in Neural Information Processing Systems 22*, eds Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta Vancouver, BC: Curran Associates, Inc., 1204–1212.

Brzosko, Z., Schultz, W., and Paulsen, O. (2015). Retroactive modulation of spike timing-dependent plasticity by dopamine. *eLife* 4:e09685. doi: 10.7554/eLife.09685

Brzosko, Z., Zannone, S., Schultz, W., Clopath, C., and Paulsen, O. (2017). Sequential neuromodulation of hebbian plasticity offers mechanism for effective reward-based navigation. *eLife* 6:e27756. doi: 10.7554/eLife.27756

Chen, J. X. (2016). The evolution of computing: alphago. *Comput. Sci. Eng.* 18, 4–7. doi: 10.1109/MCSE.2016.74

Cho, K., van Merrienboer, B., Bahdanau, D., and Bengio, Y. (2014). "On the properties of neural machine translation: encoder-decoder approaches," in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, 103–111.

Cleeremans, A., and McClelland, J. L. (1991). Learning the structure of event sequences. *J Exp Psychol Gen.* 120, 235–253. doi: 10.1037/0096-3445.120.3.235

Compte, A., Brunel, N., Goldman-Rakic, P. S., and Wang, X.-J. (2000). Synaptic mechanisms and network dynamics underlying spatial working memory in a cortical network model. *Cereb. Cortex* 10, 910–923. doi: 10.1093/cercor/10.9.910

Costa, R., Assael, I. A., Shillingford, B., de Freitas, N., and Vogels, T. (2017). "Cortical microcircuits as gated-recurrent neural networks," in *Advances in Neural Information Processing Systems 30*, eds I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Long Beach, CA: Curran Associates, Inc.), 272–283.

Cui, Y., Ahmad, S., and Hawkins, J. (2015). Continuous online sequence learning with an unsupervised neural network model. *Neural Comput.* 28, 2474–2504. doi: 10.1162/NECO_a_00893

Frank, M. J., Loughry, B., and O'Reilly, R. C. (2001). Interactions between frontal cortex and basal ganglia in working memory: a computational model. *Cogn. Affect. Behav. Neurosci.* 1, 137–160. doi: 10.3758/CABN.1.2.137

Frémaux, N., and Gerstner, W. (2016). Neuromodulated spike-timing-dependent plasticity, and theory of three-factor learning rules. *Front. Neural Circ.* 9:85. doi: 10.3389/fncir.2015.00085

Gers, F. A., Schmidhuber, J. A., and Cummins, F. A. (2000). Learning to forget: continual prediction with LSTM. *Neural Comput.* 12, 2451–2471. doi: 10.1162/089976600300015015

Gottlieb, J., and Goldberg, M. E. (1999). Activity of neurons in the lateral intraparietal area of the monkey during an antisaccade task. *Nat. Neurosci.* 2, 906–912. doi: 10.1038/13209

Graves, A., Wayne, G., and Danihelka, I. (2014). Neural turing machines. *arXiv[Preprint]* arXiv:1410.5401. Available online at: https://arxiv.org/abs/1410.5401

Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwińska, A., et al. (2016). Hybrid computing using a neural network with dynamic external memory. *Nature* 538, 471–476. doi: 10.1038/nature20101

Guerguiev, J., Lillicrap T. P., and Richards, B. A. (2017). Towards deep learning with segregated dendrites. *eLife* 6:e22901. doi: 10.7554/eLife.22901

He, K., Huertas, M., Hong, S. Z., Tie, X., Hell, J. W., Shouval, H., et al. (2015). Distinct eligibility traces for ltp and ltd in cortical synapses. *Neuron* 88, 528–538. doi: 10.1016/j.neuron.2015.09.037

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9, 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Larkum, M. (2013). A cellular mechanism for cortical associations: an organizing principle for the cerebral cortex. *Trends Neurosci.* 36, 141–151. doi: 10.1016/j.tins.2012.11.006

Larkum, M. E., Zhu, J. J., and Sakmann, B. (1999). A new cellular mechanism for coupling inputs arriving at different cortical layers. *Nature* 398, 338–341. doi: 10.1038/18686

Legenstein, R., Pecevski, D., and Wolfgang, M. (2008). A learning theory for reward-modulated spike-timing-dependent plasticity with application to biofeedback. *PLoS Comput. Biol.* 4:e1000180. doi: 10.1371/journal.pcbi.1000180

Li, Y. (2017). Deep reinforcement learning: an overview. *arXiv:1701.07274*.

Lillicrap, T. P., and Cownden, D., and Tweed, D. B., and Akerman, C. J. (2016). Random synaptic feedback weights support error backpropagation for deep learning. *Nat. Commun.* 7:13276. doi: 10.1038/ncomms13276

Lin, L.-J., and Mitchell, T. M. (1993). "Reinforcement learning with hidden states," in *Proceedings of the Second International Conference on From Animals to Animats 2 : Simulation of Adaptive Behavior: Simulation of Adaptive Behavior*, MIT Press, 271–280.

Loch, J., and Singh, S. (1998). "Using eligibility traces to find the best memoryless policy in partially observable markov decision processes," in *Proceedings of the Fifteenth International Conference on Machine Learning*, (Morgan Kaufmann).

McCallum, R. A. (1993). "Overcoming incomplete perception with utile distinction memory," in *Proceedings of the Tenth International Conference on Machine Learning*, (Morgan Kaufmann).

Melo, F. S., Meyn, S. P., and Ribeiro, M. I. (2008). "An analysis of reinforcement learning with function approximation," in *Proceedings of the 25th International Conference on Machine Learning* (ICML '08) (New York, NY: ACM).

Mink, J. W. (1996). The basal ganglia: focused selection and inhibition of competing motor programs. *Prog. Neurobiol.* 50, 381–425. doi: 10.1016/S0301-0082(96)00042-1

Minsky, M., and Papert, S. (1969). *Perceptrons*. Cambridge, MA: MIT Press.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., et al. (2015). Human-level control through deep reinforcement learning. *Nature* 518, 529–533. doi: 10.1038/nature14236

Monahan, G. E. (1982). A survey of partially observable markov decision processes: theory, models, and algorithms. *Manag. Sci.* 28, 1–16. doi: 10.1287/mnsc.28.1.1

Moore, T., and Armstrong, K. M. (2003). Selective gating of visual signals by microstimulation of frontal cortex. *Nature* 421, 370–373. doi: 10.1038/nature01341

Morrison, A., Diesmann, M., and Gerstner, W. (2008). Phenomenological models of synaptic plasticity based on spike timing. *Biol. Cybernet.* 98, 459–478. doi: 10.1007/s00422-008-0233-1

Okano, H., Hirano, T., and Balaban, E. (2000). Learning and memory. *Proc. Natl. Acad. Sci. U.S.A.* 97, 12403–12404. doi: 10.1073/pnas.210381897

O'Reilly, R. C., and Frank, M. J. (2006). Making working memory work: a computational model of learning in the prefrontal cortex and basal ganglia. *Neural Comput.* 18, 283–328. doi: 10.1162/089976606775093909

Ororbia, A. G., Mikolov, T., and Reitter, D. (2017). Learning simpler language models with the differential state framework. *Neural Comput.* 29, 3327–3352. doi: 10.1162/neco_a_01017

Pfister, J.-P., and Gerstner, W. (2006). Triplets of spikes in a model of spike timing-dependent plasticity. *J. Neurosci.* 26, 9673–9682. doi: 10.1523/JNEUROSCI.1425-06.2006

Roelfsema, P. R., and van Ooyen, A. (2005). Attention-gated reinforcement learning of internal representations for classification. *Neural Comput.* 17, 2176–2214. doi: 10.1162/0899766054615699

Roelfsema, P. R., van Ooyen, A., and Watanabe, T. (2010). Perceptual learning rules based on reinforcers and attention. *Trends Cogn. Sci.* 14, 64–71. doi: 10.1016/j.tics.2009.11.005

Rombouts, J. O., Bohte, S. M., and Roelfsema, P. R. (2015). How attention can create synaptic tags for the learning of working memories in sequential tasks. *PLoS Comput. Biol.* 11, 1–34. doi: 10.1371/journal.pcbi.1004060

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). *Learning Internal Representations by Error Propagation.* Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

Samsonovich, A., and McNaughton, B. L. (1997). Path integration and cognitive mapping in a continuous attractor neural network model. *J. Neurosci.* 17, 5900–5920. doi: 10.1523/JNEUROSCI.17-15-05900.1997

Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., and Lillicrap, T. (2016). One-shot learning with memory-augmented neural networks. *arXiv:1605.06065*.

Schultz, W., Apicella, P., and Ljungberg, T. (1993). Responses of monkey dopamine neurons to reward and conditioned stimuli during successive steps of learning a delayed response task. *J. Neurosci.* 13, 900–913. doi: 10.1523/JNEUROSCI.13-03-00900.1993

Schultz, W., Dayan, P., and Montague, P. R. (1997). A neural substrate of prediction and reward. *Science* 275, 1593–1599. doi: 10.1126/science.275.5306.1593

Singh, S., Jaakkola, T., Littman, M. L., and Szepesvári, C. (2000). Convergence results for single-step on-policy reinforcement-learning algorithms. *Mach. Learn*. 38, 287–308. doi: 10.1023/A:1007678930559

Sutton, R. S. (1984). *Temporal Credit Assignment in Reinforcement Learning*. Ph.D. thesis, University of Massachusetts, Amherst, MA.

Sutton, R. S., and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press.

Sutton, R. S., Maei, H. R., and Szepesvári, C. (2009). "A convergent o(n) temporal-difference algorithm for off-policy learning with Linear function approximation," in *Advances in Neural Information Processing Systems 21*, eds D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, (Vancouver, BC: Curran Associates, Inc.), 1609–1616.

Tetzlaff, C., Kolodziejski, C., Markelic, I., and Wörgötter, F. (2012). Time scales of memory, learning, and plasticity. *Biol. Cybern.* 106, 715–726. doi: 10.1007/s00422-012-0529-z

Todd, M. T., Niv, Y., and Cohen, J. D. (2009). "Learning to use working memory in partially observable environments through dopaminergic reinforcement," in *Advances in Neural Information Processing Systems 21*, eds D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, (Vancouver, BC: Curran Associates, Inc.), 1689–1696.

Tsitsiklis, J. N., and Roy, B. V. (1997). An analysis of temporal-difference learning with function approximation. *IEEE Trans. Autom. Control* 42, 674–690. doi: 10.1109/9.580874

Urbanczik, R., and Senn, W. (2014). Learning by the dendritic prediction of somatic spiking. *Neuron* 81, 521–528. doi: 10.1016/j.neuron.2013.11.030

van der Westhuizen, J., and Lasenby, J. (2018). The unreasonable effectiveness of the forget gate. *arXiv:1804.04849*

Vasilaki, E., Frémaux, N., Urbanczik, R., Senn, W., and Gerstner, W. (2009). Spike-based reinforcement learning in continuous state and action space: when policy gradient methods fail. *PLoS Comput. Biol.* 5:e1000586. doi: 10.1371/journal.pcbi.1000586

Waelti, P., Dickinson, A., and Schultz, W. (2001). Dopamine responses comply with basic assumptions of formal learning theory. *Nature* 412, 43–48. doi: 10.1038/35083500

Wiering, M., and Schmidhuber, J. (1998). Fast online q(λ). *Machine Learn.* 33, 105–115. doi: 10.1023/A:1007562800292

Xie, X., and Seung, H. S. (2004). Learning in neural networks by reinforcement of irregular spiking. *Phys Rev E.* 69:041909. doi: 10.1103/PhysRevE.69.041909

Yagishita, S., Hayashi-Takagi, A., Ellis-Davies, G. C., Urakubo, H., Ishii, S., and Kasai, H. (2014). A critical time window for dopamine actions on the structural plasticity of dendritic spines. *Science* 345, 1616–1620. doi: 10.1126/science.1255514

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.