Neurocomputing xxx (xxxx) xxx

Contents lists available at ScienceDirect

Neurocomputing

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

Learning continuous-time working memory tasks with on-policy neural reinforcement learning

Davide Zambrano [a,c,*], Pieter R. Roelfsema [b], Sander Bohte [c]

[a] Laboratory of Intelligent Systems of the Ecole polytechnique féd érale de Lausanne, Lausanne, Switzerland and Centrum Wiskunde & Informatica, Amsterdam, The Netherlands
[b] Netherland Institute for Neuroscience, Amsterdam, The Netherlands.and Department of Integrative Neurophysiology, Center for Neurogenomics and Cognitive Research, VU University, Amsterdam, The Netherlands
[c] Centrum Wiskunde & Informatica, Amsterdam, and Swammerdam Institute of Life Sciences, University of Amsterdam, and Dept of Computer Science, Rijksuniversiteit Groningen, The Netherlands

ARTICLE INFO

Article history:
Received 28 February 2020
Revised 10 November 2020
Accepted 21 November 2020
Available online xxxx
Communicated by Zidong Wang

Keywords:
Reinforcement learning
Neural networks
Working memory
Selective attention
Continuous-time SARSA

ABSTRACT

An animals' ability to learn how to make decisions based on sensory evidence is often well described by Reinforcement Learning (RL) frameworks. These frameworks, however, typically apply to event-based representations and lack the explicit and fine-grained notion of time needed to study psychophysically relevant measures like reaction times and psychometric curves. Here, we develop and use a biologically plausible continuous-time RL scheme of CT-AuGMEnT (Continuous-Time Attention-Gated MEmory Tagging) to study these behavioural quantities. We show how CT-AuGMEnT implements on-policy SARSA learning as a biologically plausible form of reinforcement learning with working memory units using 'attentional' feedback. We show that the CT-AuGMEnT model efficiently learns tasks in continuous time and can learn to accumulate relevant evidence through time. This allows the model to link task difficulty to psychophysical measurements such as accuracy and reaction-times. We further show how the implementation of a separate accessory network for feedback allows the model to learn continuously, also in case of significant transmission delays between the network's feedforward and feedback layers and even when the accessory network is randomly initialized. Our results demonstrate that CT-AuGMEnT represents a fully time-continuous biologically plausible end-to-end RL model for learning to integrate evidence and make decisions.

1. Introduction

The environment we live in presents a stream of information, where new events have to be recognised, some elements have to be maintained in memory, and behavior has to be adapted to optimally respond to the perceived state of the environment. Reinforcement Learning (RL) is the theoretical framework for learning from interaction with the environment, and it is deeply linked with neuroscience and psychology [1,2]. RL has been used to explain how an agent can solve complex problems, learning from very sparse and often delayed signals – rewards or punishments [3]. In many tasks, the behaviorally relevant state of the environment includes past events, like past roadsigns determining a current roadturn - learning what to remember is then crucial to constructing a compact state representation on which to act. While RL combined with modern deep learning has demonstrated impressive results in various game settings [4,5], from a biological perspective the question is how the brain accomplish such working memory tasks.

A line of recent work has proposed a central role for attentionally gated feedback in biologically plausible deep reinforcement learning [6–10]. In particular, [6,7] propose AuGMEnT (Attention-Gated MEmory Tagging) as a biologically plausible neural network RL framework that implements SARSA. In AuGMEnT, a feedforward pass through the neural network computes q-values for the various available actions from sensory inputs and an action is selected as a function of the q-values. Then, attentional feedback from the action selection stage is used for spatial credit assignment: feedback signals highlight only those weights that are responsible for the selection of the winning action. These connections are subsequently modified according to a globally released neuromodulator implementing a biologically plausible form of error-backpropagation. AuGMEnT includes working memory units to store relevant sen-

* Corresponding author.
E-mail addresses: davide.zambrano@synergysports.com (D. Zambrano), p.roelfsema@nin.knaw.nl (P.R. Roelfsema), sbohte@cwi.nl (S. Bohte).

https://doi.org/10.1016/j.neucom.2020.11.072
0925-2312/

sory information – similar to Long Short-Term Memory (LSTM) [11,12].

Still, in most such models of RL, the representation of time is abstracted into discrete events sampled in the ordered presentation: the agent is given only behaviourally meaningful new observations to elicit an update in the agent's state and the selection of a new action. Events are thus externally defined and effectively provide the agent the information on *when* a decision has to be made. Alternatively, for example in problems involving sequences of video frames like in video games, complicated frame-selection schemes are used to limit the number of actions selected [4], as standard RL methods scale poorly with the fine-grained timescale effected by higher framerates. To map closely to typical psychophysically meaningful measures like psychometrical curves and reaction times, we need to take serious the issue of continuous-time in biologically plausible reinforcement learning.

Work on powerful yet plausible models of continuous-time reinforcement learning is sparse: Bellec et al. [13] implement approximate backpropagation through time (BPTT) to implement Proximal policy optimization algorithms in a reinforcement learning with spiking neurons for tasks which require only limited memory, and Zambrano et al. [14] developed a first continuous-time version of AuGMEnT. This continuous-time RL framework, CT-AuGMEnT, solves working memory RL problems in machine learning in continuous-time through a continuous-time version of SARSA reinforcement learning coupled to an action-selection system modeled after the basal ganglia model developed by [15].

Here we expose and expand the CT-AuGMEnT framework to include effective exploration strategies, and we show how this CT-AuGMEnT decision-making framework can learn time-continuous versions of classical cognitive tasks from the literature much more efficiently as compared to a fine-grained time-stepped version of AuGMEnT. The framework thus allows us to study an important open question in neuroscience: what is the role of time in reinforcement learning? Specifically, we here study how we can use networks trained with CT-AuGMEnT to obtain reaction times and psychometrical curves for classical RL tasks studied in neuroscience.

Evidence integration in continuous-time has been studied in monkeys during decision-making tasks, such as motion-discrimination tasks where the optimal integration of sensory information is critical for an accurate response. We show that CT-AuGMEnT allows networks to *learn* to accumulate the relevant evidence by tuning the memory units to the appropriate perceptual inputs, with a performance that is comparable to that of the animals. Indeed, networks trained with CT-AuGMEnT predict how the task difficulty affects both performance and reaction-time: these aspects can only be modelled when continuous-time is considered.

We also study a related issue in continuous-time learning: the phased nature of learning in neural networks. In standard neural networks, a feedforward 'inference' phase in the network to compute outputs (or q-values) is alternated with a feedback phase where credit is assigned synapses for learning [16]. We show how non-alternating continuous learning can be implemented via a separate accessory network that carries the feedback signal [17,18]. The use of a separate accessory network to carry feedback also allows us to study network with both symmetrical feedforward and feedback weights, and asymmetrical weights, where the former is problematic from a biological perspective [19]. We find that an asymmetric accessory network incurs little cost in terms of network convergence.

Since, in a biological setting, such an accessory network will cause feedforward and feedback activity to be out of phase due to inevitable transmission delays between network layers, we investigate to what degree our CT-AuGMEnT networks can cope with delays in the propagation of information between layers: we find that the trained networks perform well, even for significant delays in an accessory network.

The paper is organised as follows: in Section 2, we summarize CT-AuGMEnT and introduce all the relevant components and the learning rule. In Section 3, we demonstrate the CT-AuGMEnT framework by illustrating continuous-time implementations of a number of standard working memory tasks from the neuroscience literature that expose different aspects of complexity in task learning, and show how we can model reaction-time experiments within the CT-AuGMEnT framework. In Section 4, we examine the impact of delays between layers when feedback is carried by a separate and randomly initialized accessory network. Finally, in Section 5, we discuss our findings and their context.

1.1. Related work

RL algorithms are typically derived as a solution of the Bellman equation [3] and aim to find policies for agents that optimize the obtained sum of (discounted) future rewards in an environment where the agent can select actions in a succession of specific state transitions. Reinforcement algorithms exist in on-policy and off-policy flavors, where on-policy algorithms like SARSA use only the experienced state-action transitions to update their policy, in contrast to off-policy algorithms like Q-Learning [3]. Both SARSA and Q-Learning algorithms are value-based RL algorithms as they aim to estimate the value of a state-action pair as the expected sum of future rewards, so-called action-values. On-policy RL algorithms like SARSA result in more conservative policies, and monkey studies have provided evidence that their behaviour is only compatible with on-policy algorithms [20–22]; experimental work by [23] suggests that working memory comprises an intrinsic and crucial part of RL in humans.

For event-based and discrete-time optimization problems, reinforcement learning has been used to successfully train deep [4,5] and recurrent neural networks [12,8]. For working memory tasks, [12] demonstrated that LSTMs can be trained with the RL Advantage Learning algorithm.

In a biological context, Todd et al. [24] used a tabular actor-critic representation, where working memory is explicitly represented as a second actor – a gating actor – which augments the current observation with past observation. The gating actor can choose to maintain or replace its element memory with the current observation. Lloyd extends the gating model by comparing two learning algorithms, Actor-Critic and SARSA, with learning patterns in rats [25]: the authors suggest that only SARSA provides faster learning as seen in animals. In this model, the motor actor is only used in the final stage of the task. Song et al. [26] proposed a working memory neural network model for decision making trained with an actor-critic off-policy algorithm called REINFORCE [27,28]. Their model shows comparable results as presented in this paper on similar tasks, but the learning algorithm is off-policy and still formulated in discrete-time. In [29] the authors proposed a biologically-plausible continuous-time approximation to gradient-based supervised learning to train a recurrent neural network. For the locality constraint, the algorithm builds on the feedback alignment theory proposed in [30].

The AuGMEnT framework [6,7] implements the SARSA RL algorithm in a neural network with working memory using a biologically plausible local learning rule.

For continuous-time (or very fine time-steps), RL algorithms can be obtained by solving the continuous-time equivalent of the Bellmann equation, the Hamilton–Jacobi-Bellman (HJB) equation [31–33]. When learning action-values, Baird demonstrated that RL using the off-policy Q-learning algorithm [34] and the on-policy SARSA RL algorithm are theoretically infeasible in

continuous-time: when the time resolution increases, the effect of a single infinitesimal action on the total reinforcement becomes undetectable [35]. Advantage Learning has been proposed as a continuous-time formalisation of Q-learning [36,35]; Advantage Learning however is still an off-policy method that computes updates using the best available action rather than the actually taken action, and is therefore insensitive to large and negative rewards (potentially fatal) during exploration [37,3, Chapter 6.5]. Continuous-time actor-critic architectures have also been proposed, where the control of actions is computed separately from an estimate of the value of the current state [33], and Bellec et al. [13] developed a spiking neural network version to implement proximal policy optimization [38]. [39] proposed a neural network model with working memory units but without hidden layers, that was trained with continuous-time TD learning. However, a continuous-time solution for SARSA has not been developed yet, and given the evidence for on-policy RL in the brain, this is an important hiatus.

In continuous-time formulations of RL, the process of decision making has to be addressed, as potentially noisy perceptual input needs to be integrated across time to make optimal decisions. In a decision-making process, the sensory-motor mapping is thought to involve cortical and subcortical structures that contribute to sensory processing, decision making and actions selection. Evidence suggests that the basal ganglia contribute to the action selection process [40,41]. [42] demonstrate how an architecture composed of an evidence accumulator implemented in the cortex together with an action selection system modelled by the basal ganglia model of [15] can optimally solve the Multiple Sequential Probability Ratio Test (MSPRT), a multi-hypothesis version of the Sequential Probability Ratio Test [43] often used to explain the brain's decision-making process. However, the decision making model from [42] does not include learning *what* perceptual evidence should be integrated; [44] proposed an actor-critic architecture for learning to make decisions, but this model lacks working memory and is not defined in continuous-time. Rao in [45] studies a combination of Bayesian inference, Partially Observable Markov Decision Processes (POMDP) and TD learning and shows how this approach can also solve MSPRT problems. CT-AuGMEnT differences from this work in the sense that it formulates TD learning in continuous-time and studies its implications in the tasks; and it adds an explicit representation of the action-value functions typical of the on-policy SARSA learning framework.CT-AuGMEnT thus serves as a model for studying how decision-making is learned in the brain based on reinforcement learning and integration of sensory evidence within working memory.

2. Continuous-time on-policy reinforcement learning

2.1. Continuous-time action-value functions

The CT-AuGMEnT algorithm [14] is a continuous-time formalisation of the on-policy SARSA neural reinforcement learning framework described in [6,7]. The working memory units in CT-AuGMEnT employ a similar linear memory principle as the Constant-Error-Carousel in LSTMs [11,12] but substitute gating mechanisms with rectified derivative inputs [6,7] for lower learning complexity; CT-AuGMEnT is also formulated strictly in terms of RL. The model framework for CT-AuGMEnT [14] is described below for discrete time-steps of size dt: by decreasing this time-step, the model approximates continuous-time.

We consider a POMDP as a continuous-time dynamical system, $f(t)$, with a discrete state set, $\mathbf{S}$, and a discrete action set, $\mathbf{A}$. For every time-step t, the system is in a state $\mathbf{s} \in \mathbf{S}$, and an action $\mathbf{a} \in \mathbf{A}$ is selected. The system receives a reward r as a function of the current state and the selected action:

$$r(t) = r(\mathbf{s}(t), \mathbf{a}(t)). \tag{1}$$

The goal is to find a state-dependent policy for selecting actions, $\mu(t)$,

$$\mathbf{a}(t) = \mu(\mathbf{s}(t)), \tag{2}$$

that maximises the cumulative future rewards,

$$Q^{\mu}(\mathbf{s}(t), \mathbf{a}(t)) = \int_t^{\infty} \mathrm{e}^{-\frac{\zeta - t}{\tau}} r(\mathbf{s}(\zeta), \mathbf{a}(\zeta)) d\zeta, \tag{3}$$

for any initial state $\mathbf{s}(t)$. $Q^{\mu}(\mathbf{s}, \mathbf{a})$is called the *action-value function* of the state $\mathbf{s}$ and action $\mathbf{a}$, and τ is the time constant for discounting future rewards as defined in [33, for value functions]. The optimal action-value function $Q^*(\mathbf{s}, \mathbf{a})$ for the optimal policy μ^* is defined as:

$$Q^*(\mathbf{s}(t), \mathbf{a}(t)) = \max_{\mu} Q^{\mu}(\mathbf{s}(t), \mathbf{a}(t)). \tag{4}$$

For optimal action-value functions, the condition for optimality at time t is given by

$$\frac{1}{\tau} Q^*(\mathbf{s}(t), \mathbf{a}(t)) = \max_{\mu} \left[r(\mathbf{s}(t), \mathbf{a}(t)) + \frac{\partial Q^*(\mathbf{s}(t), \mathbf{a}(t))}{\partial \mathbf{s}(t), \mathbf{a}(t)} f(\mathbf{s}(t), \mathbf{a}(t)) \right], \tag{5}$$

which is a discounted version of the Hamilton–Jacobi-Bellman (HJB) equation, and can be obtained by differentiating Eq. (4) (described in Appendix A).

We estimate the action-value function with a function approximator. This estimate can be updated by taking advantage of a self-consistency condition, which is derived by differentiating Eq. (3) by t:

$$\dot{Q}^{\mu}(\mathbf{s}(t), \mathbf{a}(t)) = \frac{1}{\tau} Q^{\mu}(\mathbf{s}(t), \mathbf{a}(t)) - r(t). \tag{6}$$

This condition holds for any policy including the optimal policy $Q^*(\mathbf{s}, \mathbf{a})$ (see Eq. (4)), and can be used to compute the so-called Temporal Difference (TD) error [3] as:

$$\delta(t) = r(t) - \frac{1}{\tau} Q^{\mu}(\mathbf{s}(t), \mathbf{a}(t)) + \dot{Q}^{\mu}(\mathbf{s}(t), \mathbf{a}(t)). \tag{7}$$

By combining the backwards Euler approximation as $\dot{Q}(t) = (Q(t) - Q(t - dt))/dt$ and Eq. (7), we can derive the following discrete TD update:

$$\delta(t) = r(t) + \frac{1}{dt} \left[\left(1 - \frac{dt}{\tau}\right) Q^{\mu}(\mathbf{s}(t), \mathbf{a}(t)) - Q^{\mu}(\mathbf{s}(t - dt), \mathbf{a}(t - dt)) \right], \tag{8}$$

where the reward $r(t)$ has been rescaled as $r(t)/dt$. If $dt = 1$, and if the discount factor is $\gamma = 1 - \frac{dt}{\tau}$, we obtain the standard formulation of the discrete time TD error. Note that the previous equation is exact when Q^{μ} is differentiable over t. Therefore for abrupt changes in the state or action an error may occur. These abrupt changes, however, do not exist in real systems. For example, in brains the perception system acts as a filter for the unexpected events in the environment.

2.2. Continuous-time AuGMEnT

In CT-AuGMEnT, we use as a function approximator an artificial neural network (ANN) composed of three layers of units connected by modifiable synapses (see Fig. 1), plus an action layer – the Z-layer – which specifies the currently selected action. The ANN is an abstracted representation of neural computation in the brain: in the input layer, the sensory neurons represent the stimuli, in

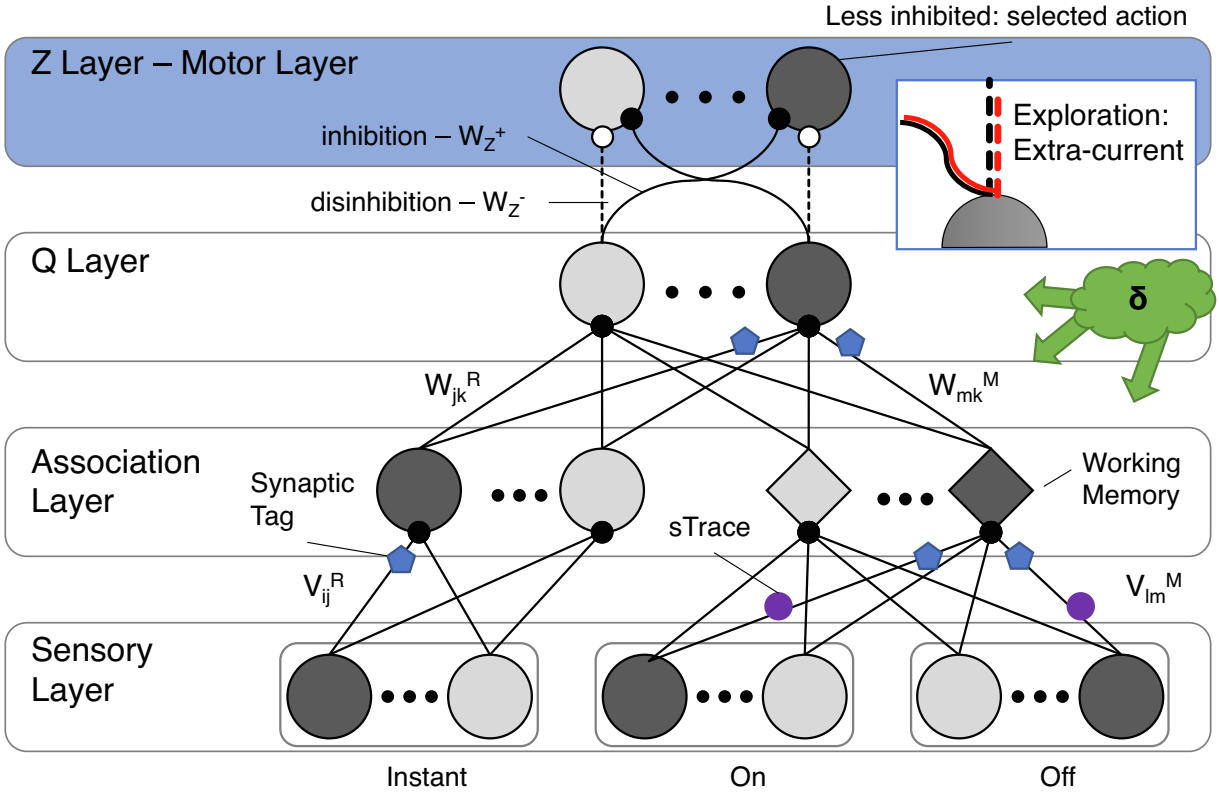

Fig. 1. CT-AuGMEnT architecture with the action selection system (feedforward dis-inhibition) in the output layer. Higher activity is depicted in dark grey: the cell's activity is proportionally responsible for the action selection and their connections are tagged correspondingly for later updating. Synaptic Tag (blue pentagons) and synaptic Traces *purple circles) are also shown. Inset: during exploration an extra input is added from the explorative Q-neuron, and consequently to all the units of the motor layer. **b** Summary of Equations for the AuGMEnT versus CT-AuGMEnT models.

the association layer the stimuli are processed further, and in the Q-layer action-values are computed. The stimuli are represented in the input layer by instantaneous ("instant" in Fig. 1) and transient (on/off) units, mimicking the behaviour of cells found in the early stages of visual cortex [46]. Instantaneous units, x, are active as long as the stimuli is present, while transient units x^+/x^- represent positive and negative changes in sensory input:

$$\begin{aligned} x^+(t) &= [\dot{x}(t)]_+ = \tfrac{1}{dt}[x(t) - x(t-dt)]_+, \\ x^-(t) &= [\dot{x}(t)]_- = \tfrac{1}{dt}[x(t-dt) - x(t)]_+, \end{aligned} \tag{9}$$

where $[.]_+$ is a threshold operation that returns 0 for negative values; as before, we assume backward Euler approximation of the time derivative of $\dot{x}^+(t)$ and $\dot{x}^-(t)$ for small dt.

The instantaneous units i from the input layers are fully connected to regular (R) units j in the association layer through connections v_{ij}^R, while transient units l of the input layer are fully connected to memory (M) units m through connections v_{lm}^M (Fig. 1). The activations for regular units y_j^R and memory units y_m^M are then computed as:

$$\begin{aligned} y_j^R(t) &= \sigma(inp_j^R(t)) = \sigma\left(\sum_i v_{ij}^R x_i(t)\right), \\ y_m^M(t) &= \sigma(inp_m^M(t)) = \sigma\left(inp_m^M(t-dt) + dt\left(\sum_l v_{lm}^M x_l^{\pm}(t)\right)\right), \end{aligned} \tag{10}$$

where $x^{\pm}(t) = [x^+(t), x^-(t)]$ is shorthand for the on/off inputs to memory units, and σ is the sigmoidal activation function $\sigma(inp) = \frac{1}{1+\exp(\theta - inp)}$, where θ is a threshold parameter set to 2.5 and with derivative:

$$\frac{\partial \sigma(t)}{\partial inp(t)} = \sigma(t)(1-\sigma(t)) = y(t)(1-y(t)). \tag{11}$$

The memory units are modeled as perfect integrators: their persistent activity mimics the behaviour of cells found for example in frontal cortex or in area LIP area of the parietal cortex [47–49]. The second part of Eq. (10) is derived from the temporal gradient approximation of $\frac{d}{dt} inp_m^M(t)$:

$$\begin{aligned} \tfrac{d}{dt} inp_m^M(t) &= \sum_l v_{lm}^M x_l^{\pm}(t), \\ \tfrac{inp_m^M(t) - inp_m^M(t-dt)}{dt} &= \sum_l v_{lm}^M x_l^{\pm}(t). \end{aligned}$$

The Q-layer receives input from the association layer by the connections w_{jk}^R and w_{mk}^M. Every neuron in this layer, q_k, computes the action-value $Q(\mathbf{s}, k)$ of the action k in the current state $\mathbf{s}$ as:

$$q_k(t) = \sum_m w_{mk}^M y_m^M(t) + \sum_j w_{jk}^R y_j^R(t). \tag{12}$$

2.3. Action selection

In CT-AuGMEnT, actions associated with the estimated action-values computed in the Q-layer continuously compete for control over behaviour. An action selection mechanism typically resolves this competition by selecting the action with the highest action-value by default and by occasionally randomly sampling from lower-valued actions. Such exploration/exploitation strategies allow the agent to find novel, more rewarding paths in the state space [3]. CT-AuGMEnT uses the Max-Boltzmann strategy [50]: the highest action value is selected with a probability of $1-\epsilon$,

and with probability ϵ a random exploration action is selected by sampling from the Boltzmann distribution:

$$P_B(\mathbf{a}) = \frac{\exp(q_a)}{\sum_k \exp(q_k)}. \tag{13}$$

This action selection rule however cannot be directly applied to a continuous-time setting. For action-values, [35] already demonstrated that the reduction of the time-step duration negatively affects the convergence rate. Intuitively, a shorter time-step corresponds to more state-action transitions, thus to a smaller effect of that action on the final reward. Moreover, function approximators, such as neural networks introduce their own imprecision in the action-value computation, exacerbating the problem. It also seems intuitively incorrect that the duration of an action, and thus its effect on the environment, depends on the dt size of the algorithm's update.

Starting from the observation that actions in the real-world have their own time requirements, CT-AuGMEnT uses an action selection system that dynamically solves the competition among the actions in the form of a simplified model of basal ganglia operation [15]. The basal ganglia *inhibits* all the non selected actions, and *disinhibits* the selected action – the action with the highest action-value. In CT-AuGMEnT, this is achieved by connecting the Z-layer to the Q-layer with off-centre and on-surround connectivity: each neuron in the Q-layer transmits a disinhibitory signal to the corresponding neuron in the Z layer through the connection w_z^- (the negative sign in Eq. (15) represents the inhibitory contribution), and it transmits a positively valued signal to inhibit all the other neurons in the Z-layer through the weights w_z^+ (see the connections between the Q and Z layers in Fig. 1).

The activity of the action selection system (Z) is computed as:

$$a_i^Z(t+dt) = a_i^Z(t) + (1-\rho) * (u_i^Q(t) - a_i^Z(t)), \tag{14}$$

$$\text{where } u_i^Q(t) = -w_z^- q_i(t) + w_z^+ \sum_{j \neq i}^{n} q_j(t). \tag{15}$$

Here, the balance between disinhibition and inhibition has been chosen as $v = w_z^+/w_z^- = 1/n$; where n is the number of actions (i.e. the optimal solution; [15]): in the case of equal q-values for all actions, the sum of the positive inhibitory input exactly balances the negative disinhibitory input. The activity of the Z-layer (action-selection), a^Z, is modelled as a leaky integrator where the constant $\rho = \exp(-\frac{dt}{\tau_\rho})$ depends on the time constant τ_ρ of that action. This equation can be viewed as an exponential filter: as τ_ρ approaches 0, there is no filtering – the output equals the new input. In this case, the output action follows any variation of the Q-values. As the time constant becomes large, transient inputs are ignored. In principle, different actions can have different time constants; however here we endow all actions with an identical τ_ρ. Winner-take-all behaviour in the Z-layer is guaranteed if the action with minimal activation (maximal disinhibited) is selected at every dt. In our model, this action is selected for a continuous period before it can be interrupted by the next action:

$$z_k(t) = \begin{cases} 1 & \text{if } k = \text{argmin}(\mathbf{a}^Z) \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

Exploration takes place with probability ϵdt, and the action is selected from the Boltzmann distribution of the actions' expected values, according to Eq. (13). The exploration mechanism overrides the selection mechanism by adding an external input; this strategy is compatible with the evidence observed in humans [51], where prefrontal regions associated with high-level control are active during a behavioural switch from an exploitation to an exploration strategy.

In the model, an external input I_{ex} is added to the explorative action in case of exploration – in Eq. (15), see insert in Fig. 1 – as:

$$\begin{aligned} I_{ex} &= |2\max(Q) - \min(Q) - q_{ex}|, \\ u_{ex}^Q(t) &= -w_z^-(q_{ex}(t) + I_{ex}) + w_z^+ \sum_{j \neq ex}^{n} q_j(t). \end{aligned} \tag{17}$$

The magnitude of I_{ex} guarantees that the selected action – with Q-value q_{ex} – overcomes the inhibition provided by the highest action-value, which is achieved by setting I_{ex} to the sum of the difference between the highest and the lowest Q-values and the difference between the highest valued action and the selected exploratory action. The signal I_{ex} is added for a fixed amount of time T_{ex}, set to $T_{ex} = 3\tau_\rho$: given the step response of the first-order linear system of Eq. (14), the summed contribution corresponds to the time needed to reach 95% of the maximum activation. The exploration mechanism in our model takes into account the time constant of the selected action: a longer action time constant implies longer exploration; here we used identical time constants for all actions, for simplicity.

2.4. Feedback connections gate learning

In CT-AuGMEnT [6,7], two factors modulate the network plasticity during learning: the TD-error computed by Eq. (14), – which in the brain would be signalled by a global neuromodulatory signal (such as the global release in the brain of dopamine or acetylcholine) – and an attentional feedback signal that is propagated from the response selection stage back to earlier processing levels and gates the plasticity: both signals contribute to learning. Since the action-value function is estimated by a function approximator, a convenient way to reduce the TD-error is to use gradient descent on the squared prediction error as [52,53,7]:

$$\frac{\partial E(t)}{\partial w_i} = \frac{1}{2}\frac{\partial}{\partial w_i}\delta(t)^2 = \delta(t)\frac{\partial}{\partial w_i}\left[r(t) - \frac{1}{\tau}Q^\mu(\mathbf{s}(t),\mathbf{a}(t);\mathbf{w}) + \frac{\partial Q^\mu(\mathbf{s}(t),\mathbf{a}(t);\mathbf{w})}{\partial t}\right], \tag{18}$$

where $\mathbf{w}$ is the vector of the ANN's parameters and delta is the TD-error also known as reward prediction error. The feedback is provided by the unit that encodes the selected action $\mathbf{a}$, and it makes the synapses responsible for the current selection eligible for plasticity by creating synaptic tags (blue pentagons in Fig. 1). As defined in [6], synaptic tags are equivalent to eligibility traces, which, as in [33], have the form of:

$$\dot{Tag}_i(t) = -\frac{1}{\phi}Tag_i(t) + \frac{\partial Q(\mathbf{s}(t),\mathbf{a}(t);\mathbf{w})}{\partial w_i}, \tag{19}$$

where $0 < \phi < \tau$ is the time constant of the tag decay. Thus, by discretising Eq. (19), the tag updates for the synapses of regular units (R) and the memory units (M) with the Q-layer are defined as:

$$\begin{aligned} Tag_{jk}^R(t+dt) &= \left(1 - \tfrac{dt}{\phi}\right)Tag_{jk}^R(t) + dt\left[y_j^R(t)z_k(t)\right], \\ Tag_{mk}^M(t+dt) &= \left(1 - \tfrac{dt}{\phi}\right)Tag_{mk}^M(t) + dt\left[y_m^M(t)z_k(t)\right], \end{aligned} \tag{20}$$

with $z_k = 1$ for the selected action ($k = a$), $z_k = 0$ elsewhere ($k \neq a$) as defined by Eq. (16) (Appendix B gives the full derivation of these updates). Hence, the selected action a provides feedback and it thereby enables the plasticity of connections to winning output unit a. Note that to be fully local, the winning action activity has to be visible to the connections in the Q-layer. This can be achieved through an accessory feedback network as described in Section 4 combined with a fast winner-take-all circuit [54]. In the discrete-

time AuGMEnT the tag decay was defined as $\alpha = (1 - \lambda\gamma)$; here, to be consistent, we define: $\lambda = (1 - \frac{dt}{\phi})/(1 - \frac{dt}{\tau})$. As a result from the tag update equation, we observe that the association units that provided stronger input to the winning action **a**, also receive stronger feedback – they will be held responsible for the outcome of the action and increase their strength if $\delta(t)$ is positive but decrease their strength if $\delta(t)$ is negative. Equivalently, tags on connections between regular units in the association layer and instantaneous units in the input layer depend on the activity in the input layer units themselves and on the feedback activity from the selected action to the regular unit in the association layer:

$$Tag_{ij}(t+dt) = \left(1 - \frac{dt}{\phi}\right) Tag_{ij}(t) + dt\left[x_i(t)\frac{\partial\sigma(inp_j^R(t))}{\partial inp_j^R(t)} w'^R_{aj}\right], \quad (21)$$

where x_i is the presynaptic activity in the input layer, $\frac{\partial\sigma(inp_j^R(t))}{\partial inp_j^R(t)}$ depends on the postsynaptic activity in the regular association layer unit, and w'^R_{aj} is feedback from the winning output unit to unit j; all three signals are locally available at the synapse (see also the full derivation in Appendix B). In this formulation, the feedback connections w'^R_{aj} and feedforward connections w^R_{ja} have the same strength, though as pointed out in [18] this is not a necessary requirement and can emerge during the learning process.

As in AuGMEnT [7], we use synaptic traces (purple circles in Fig. 1) between sensory units and memory cells for working memory learning:

$$sTrace_{lm}(t+dt) = sTrace_{lm}(t) + dt[x_l^{\pm}(t)]. \quad (22)$$

The traces build up over time if the pre-synapse is active. Traces can be transformed in tags by feedback from the response selection stage, just as is the case for the tags on the connections between the input layer and the regular units in the association layer.

$$PTag_{lm}(t+dt) = \left(1 - \frac{dt}{\phi}\right) Tag_{lm}(t) + dt\left[sTrace_{lm}(t)\frac{\partial\sigma(inp_m^M(t))}{\partial inp_j^M(t)} w'^M_{am}\right]. \quad (23)$$

The plasticity of all synapses (either R or M units) follows:

$$\begin{aligned} v_{ij}(t+dt) &= v_{ij}(t) + dt[\beta\delta(t)Tag_{ij}(t)] \\ w_{jk}(t+dt) &= w_{jk}(t) + dt[\beta\delta(t)Tag_{jk}(t)], \end{aligned} \quad (24)$$

which shows that only when tags are formed, the synapses become susceptible to the TD-error $\delta(t)$ as encoded by the neuromodulator. Since the weight update uses the current estimate of the $\delta(t)$ error and the value of the tags, the traces and tags have to be updated after updating of the weights. As is common in discrete-time RL, memory units, tags and traces are reset to zero at the end of every trial, and the transition to the terminal state generates a δ error computed with an expected reward set to zero. In the brain, tags and traces would reset either passively, through temporally spaced trials, or actively, for example using internal reset actions [55]. For CT-AuGMEnT, an equivalent method is to update the network for an entire unit-of-time ($T_{end} = \frac{1}{dt}dt$), with the δ error computed with $Q(\mathbf{s}(t), \mathbf{a}(t)) = 0$ and where T_{end} denotes the final event or time-step in the respective task. In Appendix B we show how these learning rules minimise the expected squared-prediction error.

Summarising the adaptation of AuGMEnT to continuous-time, the equations that map the forward neural computation in AuGMEnT to CT-AuGMEnT are shown in Fig. 2.

3. Solving continuous-time working memory RL tasks

First, we compare the CT-AuGMEnT model to the event-based version of the AuGMEnT method to demonstrate the limits of time-stepped representations for neuroscience modelling and the need for effective decision-making circuits and related exploration mechanisms. For this, we study three classical tasks from the neuroscience literature: a *Saccade/anti-Saccade* task, a *Delayed Match to Category* and a *two- and four-choice* Motion Discrimination Task. The *Saccade/anti-Saccade* (SaS) task as presented in [7] models a classical problem [48] which requires learning a non-linear XOR-like mapping. The *Delayed Match to Category* (DMC) task introduced in [56] demonstrates continuous-time evidence collection and decision making, while the *Two- and Four-choice* motion dis-

	AuGMEnT		CT-AuGMEnT
Feedforward			
	$x^+(t) = [x(t) - x(t-1)]_+$		$x^+(t) = 1/dt\,[x(t) - x(t-dt)]_+$
	$x^-(t) = [x(t-1) - x(t)]_-$	Eq. 9	$x^-(t) = 1/dt\,[x(t-dt) - x(t)]_-$
	$y_j^R(t) = \sigma\left(\sum_i v_{ij}^R x_i(t)\right)$		$y_j^R(t) = \sigma\left(\sum_i v_{ij}^R x_i(t)\right)$
	$y_m^M(t) = \sigma\left(a_m^M(t-1) + \sum_l v_{lm}^M x_l^{\pm}(t)\right)$	Eq. 10	$y_m^M(t) = \sigma\left(a_m^M(t-dt) + dt\sum_l v_{lm}^M x_l^{\pm}(t)\right)$
	$q_k(t) = \sum_m w_{mk}^M y_m^M(t) + \sum_j w_{jk}^R y_j^R(t)$	Eq. 12	$q_k(t) = \sum_m w_{mk}^M y_m^M(t) + \sum_j w_{jk}^R y_j^R(t)$
Action Selection (ε - greedy)			
		Eq. 14	$u_i^Q(t) = -w_z^- q_i(t) + w_z^+ \sum_{j\neq i} q_j(t))$
		Eq. 15	$a_i^z(t+dt) = a_i^z(t) + (1-\rho)\,(u_i^Q(t) - a_i^z(t))$
	argmax(q_k)		argmin(a^z_i)
Learning			
	$\delta(t) = r + \gamma q_{K'}(t) - q_K(t-1)$	Eq. 8	$\delta(t) = r + \frac{1}{dt}[(1 - \frac{dt}{\tau})q_{K'}(t) - q_K(t-dt)]$
	$Tag(t+1) = \lambda\gamma Tag(t) + \frac{\partial q_K}{\partial w}$	Eq. 19	$Tag(t+dt) = (1 - dt/\phi)Tag(t) + \frac{\partial q_K}{\partial w}$
	$w(t+1) = w(t) + \beta\delta(t)Tag(t)$	Eq. 24	$w(t+dt) = w(t) + dt\beta\delta(t)Tag(t)$

Fig. 2. Summary of Equations for the AuGMEnT versus CT-AuGMEnT models.

crimination task (MDT) allows us to study the link between continuous-time learning and psychophysical measurements like reaction times (RT) and performance. In Appendix C, two other tasks are described: the *Motion-or-Colour* (MoC) task from [57] combines continuous-time evidence integration with non-linear mapping; and the *T-Maze* (TM) task from the machine learning literature, where an agent has to reach the end of a corridor of length N and then make a decision; for the latter task, we compare CT-AuGMEnT with a continuous-time version of LSTM.

The meta–parameters for all the simulated tasks are set to $\beta = 0.15; \lambda = 0.20; \gamma = 0.90; \epsilon = 0.025$ and $\theta = 2.5$. To compare with the same parameters used in [7], we use τ and ϕ computed for $dt = 1$ and then λ and γ are scaled accordingly with respect to dt. The initial synaptic weights are drawn from a uniform distribution of $U[-0.25, 0.25]$. At the end of the learning phase, we test the network by evaluating the accuracy of the responses for every condition with β and $\epsilon = 0$ (so that learning and exploration is switched off). The accuracy is reported in Table 2 for the Saccade-anti-Saccade task. For the Delayed Match to Category and Two- and Four-choice tasks, due to the presence of noise in the input, we report the number of networks that achieve the convergence criterion.

Table 1 summarises the full implementation details; all results in this section are reported for networks with symmetric feedback weights as in the original AuGMEnT implementation. As in AuGMEnT, the network architectures use relatively few neurons, which proved sufficient to learn the tasks.

The Saccade/anti-Saccade (SaS) task, [6,7,14] is used to study working memory in monkeys [48] and requires the maintenance of a presented cue in working memory and a cue-dependent action to be taken at the go signal. This task is an example of a task where a non-linear mapping between the state and action space has to be computed so that the network needs to have at least one hidden layer. The continuous-time implementation demonstrates how the learning process depends on the timed phases of the task, such as the time duration of the cue or the delay phase.

As illustrated in Fig. 3a, the SaS task starts with an empty screen, then the *Fixation* phase begins where the respective mark is shown: a small reward is given, r_{fix}, if the agent succeeds in maintaining fixation for 2 s. Then, a cue appears on either the left or the right of the fixation mark – the *Cue* phase (red circle in Fig. 3a). The cue is presented for 1 s, then a *Delay* phase of 1s follows when only the fixation mark is presented. Any interruption of fixation during this phase (e.g. an eye movement towards the red cue) terminates the task without reward. The disappearance of the fixation mark signals the *Go* phase, where an eye movement is requested.

In the SaS task, the type of fixation mark determines the strategy to adopt: a cross mark requires a *pro-saccade* decision, while a triangle mark requires an *anti-saccade*. In the pro-saccade condition, the agent has to move its eyes toward the remembered location of the cue, while in the anti-saccade it has to move its eyes in the opposite direction. Only the correct choice is rewarded with r_{fin}.

The AuGMEnT and CT-AuGMEnT networks are comprised of 4 input units, two that signal the presence of the fixation marks (cross or triangle, signalling the pro-saccade or anti-saccade condition, respectively) and the others signal the two possible cues (left or right). The networks contain 4 regular and 4 memory units in the association layer, and 3 output neurons, corresponding to the 3 actions they can take: fixate in the centre of the display, move eyes left or move eyes right.

The results for the SaS task are plotted in Fig. 3b,c: we see much better scaling behavior for CT-AuGMEnT as compared to AuGMEnT (Fig. 3b), as AuGMEnT quickly fails to converge for smaller dt whereas CT-AuGMEnT successfully learns the task for every dt. and for both action time-constants τ_ρ, with similar learning curves for the two action time-constants (Fig. 3c) . CT-AuGMEnT learns the task with a moderate increase of trials for increasing time resolution (Table A3.3), which is likely due to the structure of the task that induces specific moments when exploration is most effective, and these moments become less likely to be selected for explo-

Table 1
Summary of the tasks' configuration. SaS: Saccade-anti-Saccade; DMC: Delayed Match to Category; MDT: Two- and Four-choice Motion Discrimination Task. Convergence criteria are taken from the original tasks, except for the DMC and the MDT tasks where the criterion was matched to the observed task performance.

Task	Architecture	dt	τ_ρ	Rewards	Trained Nets	Convergence Condition	Pre-training Condition
SaS	In = 4 (Anti, Pro, Cue Left, Cue Right), R = 4, M = 4, Out = 3 (Fixate, Left, Right)	0.5, 0.1	0.5, 0.1	$r_{fix} = 0.2$, $r_{fin} = 1.5$	150	90% in the last 50 trials	None
DMC	In = 13, R = 4, M = 5, Out = 3 (Fixate, Left, Right)	0.1, 0.03	0.5, 0.1	$r_{fix} = 0.2$, $r_{fin} = 1.5$	150	80% in the last 50 trials	80% in the last 50 trials
MDT	9 In, 5 R, 5 M, 5 Out (Fixate, Up, Down, Left, Right)	0.03	0.5	$r_{fix} = 0.2$, $r_{fin} = 1.5$	100	90% in the last 50 trials	80% in the last 50 trials

Table 2
Summary of the results for the comparison between AuGMEnT and CT-AuGMEnT. For the SaS task, accuracy is reported, and for the DMC and MDT task the percentage of networks that converge.

Task	dt	AuGMEnT		CT-AuGMEnT $\tau_\rho = 0.5$		CT-AuGMEnT $\tau_\rho = 0.1$	
		Accuracy (%)	Trials	Accuracy (%)	Trials	Accuracy (%)	Trials
SaS	0.5	42	5874 ± 213	99	4403 ± 190	95	4715 ± 257
	0.1	0	n.c.	97	4966 ± 234	93	5480 ± 248
		Convergence (%)		Convergence (%)		Convergence (%)	
DMC	0.1	7	17119 ± 741	100	2828 ± 162	79	2468 ± 185
	0.03	0	n.c.	97	4004 ± 276	83	3068 ± 307
MDT	0.03	–	–	89	12199 ± 67	98	10510 ± 67

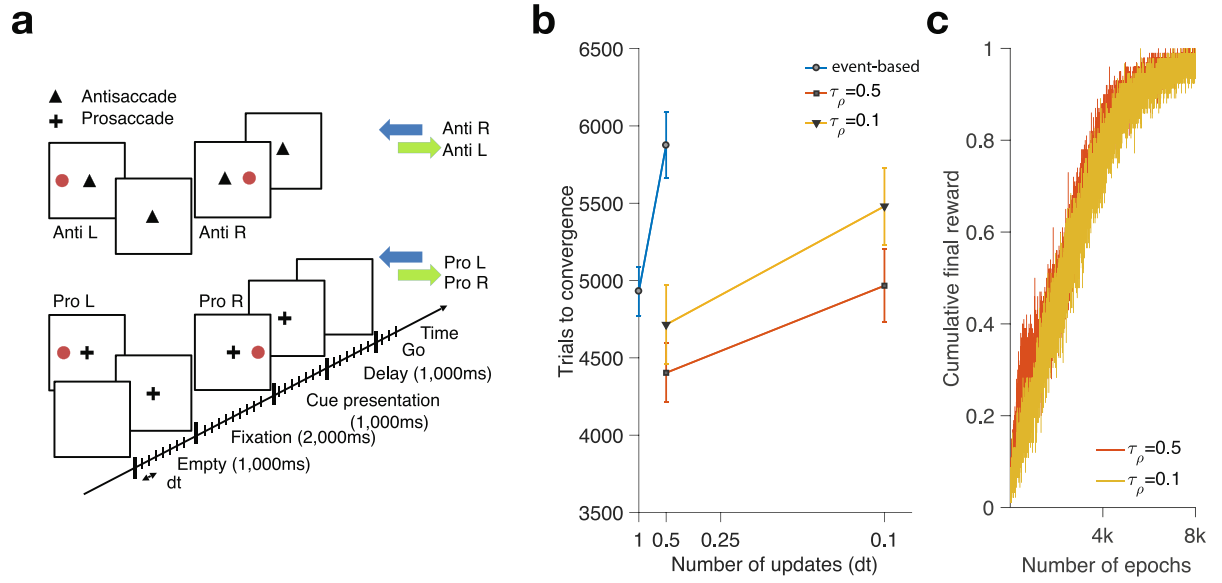

Fig. 3. **a** Saccade/antiSaccade task. The agent has to make an eye movement to the Left (blue arrow) or Right (green arrow) depending on the fixation mark (cross for Pro-saccade and triangle for Anti-saccade) and the position of the Cue (red circle). **b** Comparing the event-based version of AuGMEnT (blue line, event-based) and CT-AuGMEnT (red line $\tau_\rho = 0.5$ and yellow line $\tau_\rho = 0.1$). We plot the number of trials needed to reach convergence for the task; the abscissa denotes the size of *dt* used for the simulations. **c** Comparing the learning curves for the two action time constants (red line $\tau_\rho = 0.5$ and yellow line $\tau_\rho = 0.1$) at $dt = 0.1$.

rative actions as time-resolution increases. For example, an explorative action taken at the beginning of the GO phase is much more effective than the same choice made earlier or later in time, which will necessarily results in a break of fixation aborting the trial without reward. In event-based representations, the explorative or exploitative actions are chosen in specific and crucial task-related moments.

The Delayed-Match-to-Category (DMC) task was originally introduced in [56]. This task is a decision-making task where the sensory evidence has to be collected and memorised to be compared to subsequent sensory inputs. Here the decision is made as soon as enough evidence from the second motion direction is collected. The agent has to map 12 motion directions of a cloud of moving dots onto two categories (Fig. 4a red or blue arrows in the inset). In [7], every motion direction was modelled as a different input signal for one time-step; here, the same task is modeled in continuous-time. The agent has 12 inputs units each tuned to one of 12 different motion directions – from 0° to 330° spaced by 30° – with a Gaussian tuning function ($\mu = i \cdot 30°$, where i is between 0 and 11, $\sigma = 30°$), with a receptive field including all dots, thus each input unit receives input from all the moving dots. At every *dt*, 100 dots are generated, representing one of the motion directions to be categorised, and Gaussian noise ($\mu = 0, \sigma = 15°$) is added to the dots' motion. This process is shown in the inset of Fig. 4a. Here, the amount of noise has been modelled for the task with $dt = 0.03$, which is similar to the update frequency of motion-dots tasks for monkeys. Note that we consider the motion as a property of the dot that is perceived within the receptive field

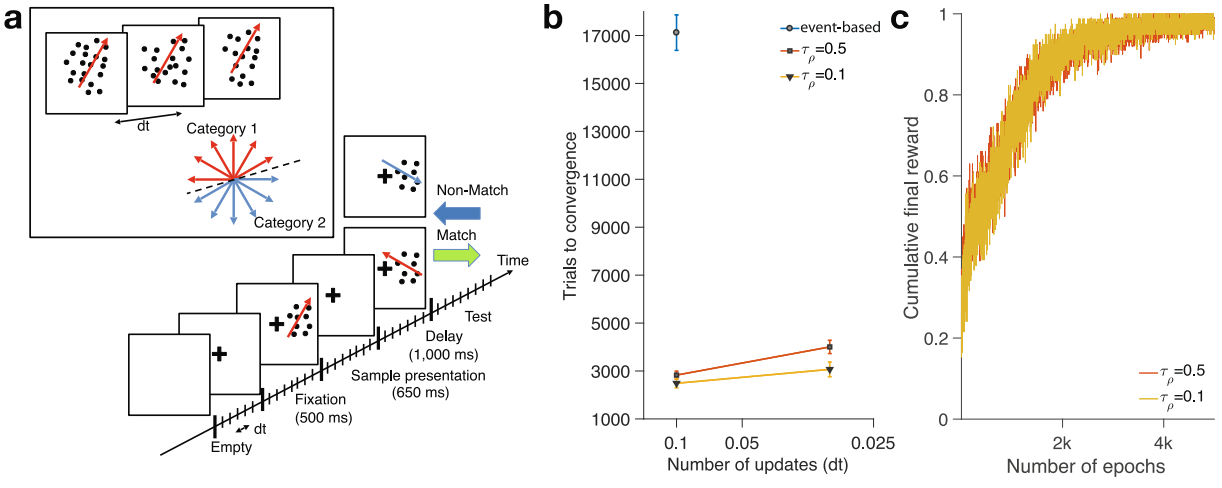

Fig. 4. **a** DMC task. The agent has to discriminate whether the second motion direction belongs to the same or opposite category by making an eye movement to the right or left respectively. In the inset: at every *dt* a new set of dots is presented, each representing the current motion direction plus noise. The motion direction belongs to one of the two categories (blue or red arrows) and is chosen from 12 possible directions (from 0° to 330°, with step size of 30°), with a category boundary separating the two categories. **b** Comparing the event-based version of AuGMEnT (blue line, event-based) and CT-AuGMEnT (red line $\tau_\rho = 0.5$ and yellow line $\tau_\rho = 0.1$). We plot the number of trials needed to reach convergence for the task; the abscissa denotes the effective size of *dt* used for the simulations. **c** Comparing the learning curves for the two action time constants (red line $\tau_\rho = 0.5$ and yellow line $\tau_\rho = 0.1$) at $dt = 0.1$.

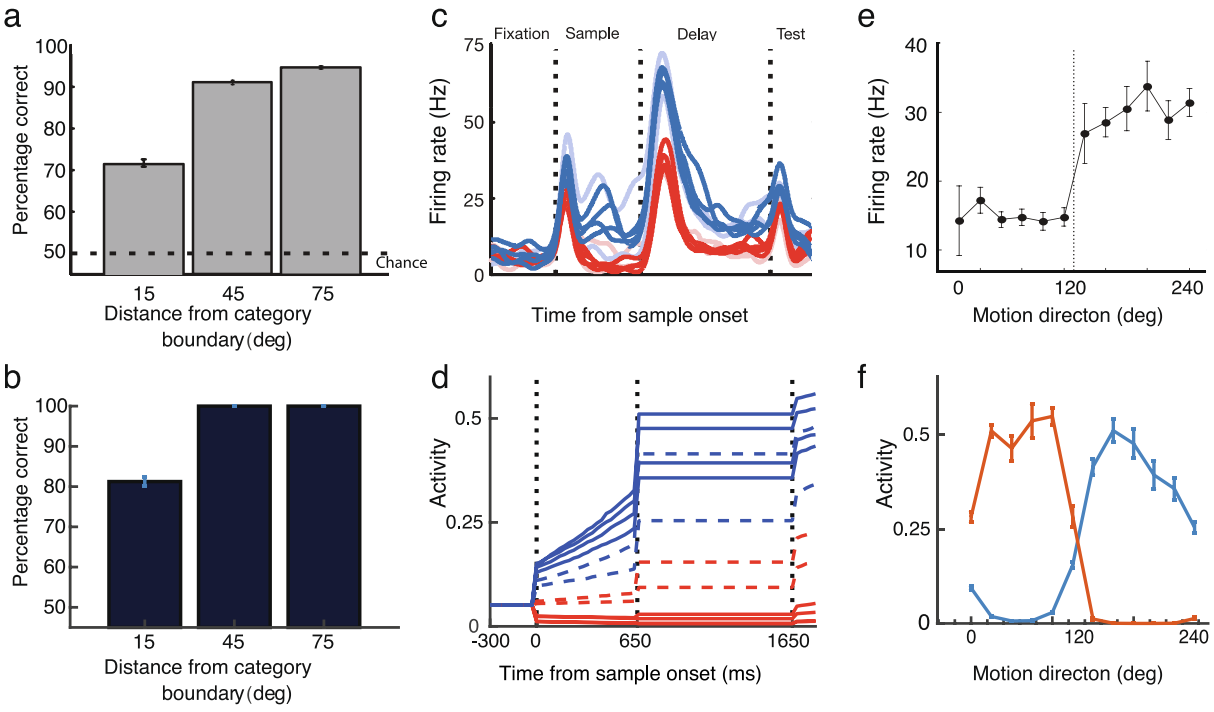

Fig. 5. Comparison between original results for the DMC task (top row, from [56]), and the model results (bottom row). **a**, **b** percentage of correct responses after training for data and model respectively. **c** activity of a LIP neuron for the 12 directions. **d** activity of one memory cell during the sample and delay phases for each of the 12 motion directions. **e** activity of an LIP neuron as a function of the 12 motion directions presented **f** activity of two memory cells in the CT-AuGMEnT network as a function of the 12 motion direc.tions presented.

of the input neurons. For that, the dot motion value is affected by a measurement error, which is the type of noise we modelled.

To speed up learning, and as in [18], we introduce a pre-training session that teaches the agent to group each motion direction in one of the two categories: the agent has to fixate a fixation mark in the centre of the display for 1s – receiving a shape reward of $r_{fix} = 0.2$ – the dots are then presented for 650 ms while the agent has to maintain fixation. Next, the stimulus is removed and the agent has to select one of the two categories – moving its eyes to the left or right. Learning ends when at least 80% of the answers were correct in the last 50 trials for each category. After the pre-training phase, the task begins: the agent has to fixate the fixation mark for 1s – worth r_{fix} – after which a first phase of dots are presented for 650 ms, with the direction of the dots chosen from one of the 12 directions. If the agent selects an action other than *Fixate*, the trial is aborted without reward. After a delay phase of 1s, another motion phase follows, with the dots moving in a new direction: the agent has to choose whether the two motion directions belong to the same category by selecting the action *Right*, or *Left* when the motions do not belong to the same direction (see time line of events in Fig. 4a).

As shown in Fig. 4b,c, CT-AuGMEnT successfully learns the DMC task within considerably fewer trials than AuGMEnT (Fig. 4b), completely and AuGMEnT completely fails to learn the task for smaller dt (Table A3.3) whereas CT-AuGMEnT requires only a small increase in the number of trials needed to reach convergence. Note also the limited dependence of the learning curves on the action time-constant (Fig. 4c).

We illustrated the integration of evidence by the working memory units of fully trained CT-AuGMEnT networks for the DMC task in more detail in Fig. 5, and compare this to neurophysiological data. We plot the activity of a number of example neurons that have been recorded in area LIP of the parietal cortex of monkeys (top row) and a trained model with $dt = 0.03$ and $\tau_\rho = [0.5]$ (bottom row). Motion directions near the category boundary between the two motion categories are more difficult to discriminate in the presence of noise. As a result, the accuracy of the model is lower near the category boundary (Fig. 5a for original data from [56], b for CT-AuGMEnT). Fig. 5c and d show the activity of example memory cells for all the 12 motion directions in the different phases of the task: just like hand-wired models of decision-making with working memory [58], the memory cells learn to maintain sensory evidence about the category of the sample during the delay period. Similar to the neuronal data recorded in LIP of parietal cortex (Fig. 5c), the activity of memory cells in the model is specific for the category of the motion (Fig. 5d, blue or red lines). We also find that the response of individual memory cells ramps according to the amount of evidence encountered in different conditions, just as LIP neurons do. For motion directions close to the category boundary (dashed lines) the category selectivity is less pronounced. When increasing the number of memory units, the category specialization still holds (see Figs. A5.2 and A5.3) In contrast to the LIP data, in CT-AuGMEnT the evidence for one category is integrated through time but does not exhibit the initial transient response visible in LIP; this transient response in LIP data may be related to neural adaptation processes, which are not modelled in CT-AuGMEnT. Fig. 5f plots the tuning of two memory cells to motion direction: it can be seen that memory cells have become selective for the category of the motion stimulus, just like the neurons in area LIP (Fig. 5e).

In the ***two- and four-choice Motion Discrimination Tasks (MDT),*** the accuracy and speed of choices of monkeys are measured to evaluate how multiple alternatives affect the decision process [59]. This decision-making task allows us to study how different degrees of motion coherence affect both accuracy and reaction times of the agent.

The sequence of the task events is shown in Fig. 6a. The agent has to fixate on the fixation mark for 1s – worth $r_{fix} = 0.2$ – then

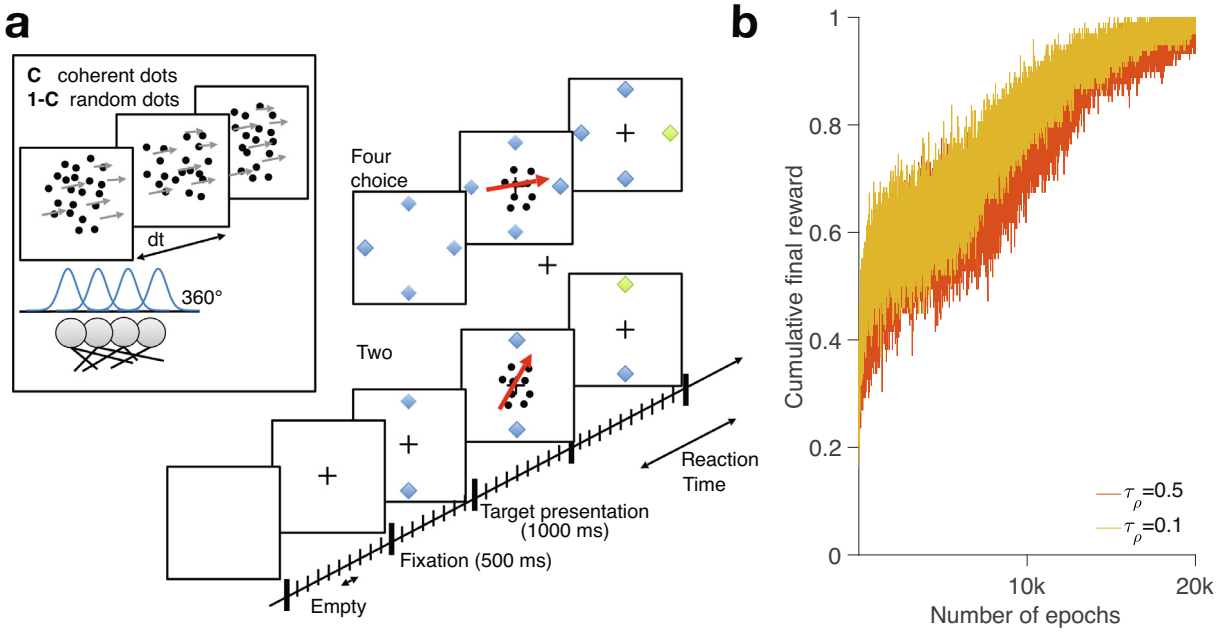

Fig. 6. **a** MDT task. The agent has to discriminate the dots' motion direction as soon as possible by making an eye movement to on one of the two or four targets shown (blue diamonds, the green diamond is the one selected by the model). In the inset: at each *dt* a new set of dots is presented. A fraction *C* moves coherently, toward the target, the others move randomly. **b** The learning curves for the two action time constants (red line $\tau_\rho = 0.5$ and yellow line $\tau_\rho = 0.1$) at $dt = 0.1$.

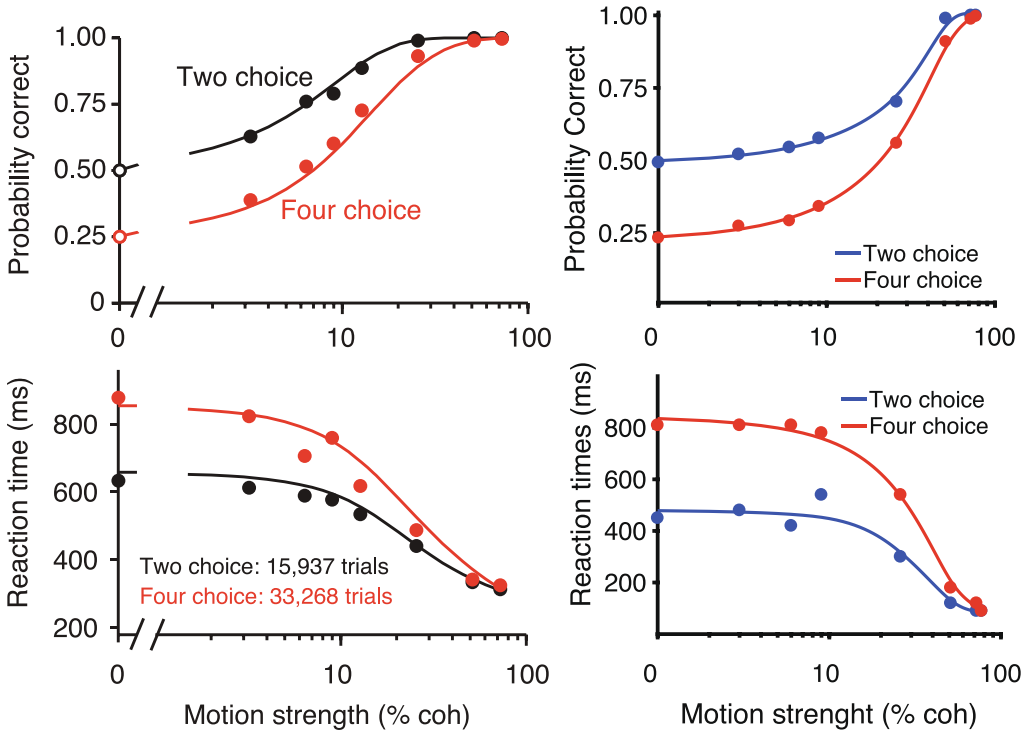

Fig. 7. Comparison between monkey data [59] (left) and model results (right) for the two- and four-choice MDT task. Similar to the monkey data, accuracy increases with increased coherence. In both two and four choice conditions, the model and the monkey learn to choose an action directed to a target, even in the absence of coherent motion rather than to maintain fixation (top row). In the bottom row, we show that the model approximately matches the reaction times observed in the monkey data. More evidence results in faster reaction times and more choices result in s.lower responses.

2 or 4 targets are shown for another second (blue diamonds in Fig. 6a). Next, 100 moving dots are presented, similar to the DMC task (inset in Fig. 6a). The coherence of the dots' motion is varied across trials: a fraction of [0%, 3%, 6%, 9%, 26%, 51%, 72%, 76%] of the dots moves accordingly in the target direction, while the other dots move to random directions (chosen uniformly between 0° and 360°). At every $dt = 0.03$, new dots are presented. The agent has to respond as quickly as possible by making an eye

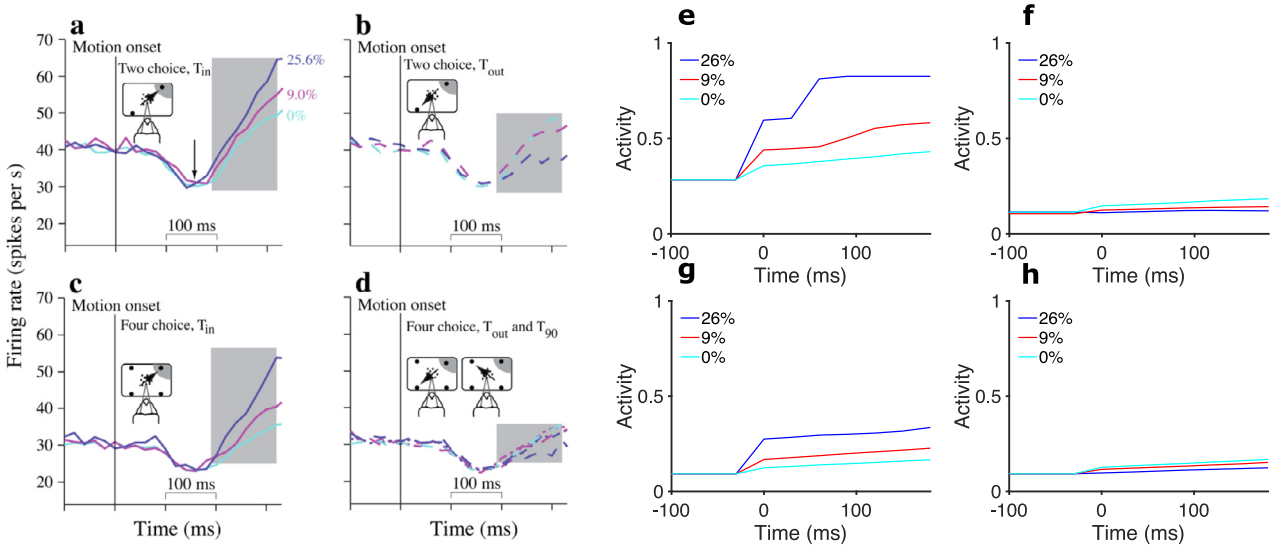

Fig. 8. Activity of memory neurons compared with electrophysiological data. **a–d** Population average firing rates of LIP neurons from [59] during three motion strengths. Top row is for two choices and bottom row for four choices. **e–h** Averaged memory neurons activity during a 10 k random test-set MDT for similar motion coherences. The same memory unit was considered. Time is shown referenced at the motion onset. CT-AuGMEnT shows several similarities with electrophysiological data: 1) Memory units increase their activities during the dots presentation **e,g**; 2) the order for different motion strengths is kept **e,g**; the four-choice condition has lower activity **g**. One-to-one matching it is not possible, however as neural dynamics are no.t modelled here.

movement in the direction cued by the dots' motion. Reaction times are measured starting from the dots' presentation until the model selects one of the saccade targets in the Z-layer. In the four-choice condition the motion directions are 90° apart, while in the two-choice condition they are 180° apart. As in the DMC task that was described in the above, we model the tuning-curve of units in the input layer with a Gaussian function with mean centred on one of the 12 motion directions and $\sigma = 30°$. For the full task, training is interrupted as soon as the model reaches an accuracy of 90% over the last 50 trials, measured across all conditions with at least 51% of motion coherence.

As shown in Fig. 6b, CT-AuGMEnT learns the task, for both action time constants at approximately the same rate of improvement.[1] CT-AuGMEnT also achieves good convergence (Table A3.3), though substantially better and faster for the faster action time-constant.

The MDT task allows us to study the ability of the CT-AuGMEnT model to commit toward a choice when evidence is provided continuously through time. Monkeys exhibit an increase of the reaction times (RTs) as a consequence of a larger number of alternatives [59]. In CT-AuGMEnT, the discount factor γ encourages the model to make decisions as quickly as possible. We compare the measured reaction times in the trained models to the original data from the MDT choice task. Fig. 7 illustrates the similarities between the monkey data (left) and the trained model (right). When more choices are possible, the task becomes more difficult: the number of correct trials decreases and the reaction times increase. When the motion coherence is 0%, the probability of a correct choice is 0.5 for the two-choice condition (blue line) and 0.25 for the four-choice condition; the agent responds always correctly for high motion coherence ($> 75\%$). Importantly, CT-AuGMEnT correctly predicts an increase in reaction time when the number of choices increases. For the MDT task, we also find that action-values are closer when the four-choice condition starts demonstrating higher uncertainty in the action selection, and thus longer reaction times.

As was done for the DMC task, a qualitative comparisons can be made for the MDT between our model predictions and electrophysiological data (Fig. 8). Again, notice that the recorded neural activity only matches our model in the ordering of activity magnitude as we do not explicitly model neuron dynamics.

Overall, for the two working memory tasks (SaS, DMC), we find that the reduction of the time-step size (dt) affects AuGMEnT both in terms of number of trials to reach the convergence criterion and in the percentage of networks that correctly learn the task (Table A3.3, first column). In particular, for $dt = 0.1$ none of the AuGMEnT networks reached convergence for any of the tasks. As shorter action time-constants CT-AuGMEnT behave more like AuGMEnT, we find in that this indeed results in somewhat lower convergence rates (Table A3.3, second and third columns); for the SaS tasks the action time-constant also affects the number of trials to reach convergence in that decreasing the action time-constant in CT-AuGMEnT increases the number of trials needed to reach convergence. This is mainly due to the effect of the action time-constant on the duration of the exploration: longer time-constants imply longer explorations, thus longer explorative actions, which have a higher impact on the task (more credit is assigned). For the DMC task however, the agent reaches the convergence criterion faster with a shorter action time-constant. The reason is likely the kind of task, as the DMC is effectively a decision-making task under uncertain information and a faster action time-constant corresponds to a lower threshold in the decision-making process (since a faster action time-constant induces a more rapid switch of actions and thus a smaller amount of evidence is needed to make a decision); this effect can be seen also in the MDT decision-making task.

Comparing the speed of learning in CT-AuGMEnT to that of animals, we note that monkeys typically require tens of thousands of trials (about 1,200 trials per day for weeks to months to learn a task with a complexity that is similar to the DMC [60]. Learning in networks trained with CT-AuGMEnT is therefore substantially faster than learning in experimental animals. Finally, learning in

[1] Lacking a corresponding representation for motion coherent stimuli to train AuGMEnT, we trained only CT-AuGMEnT for the MDT task.

animals seems compatible with the RL framework. Specifically, the activity of dopaminergic neurons in specific regions of the midbrain has been correlated with the hypothesis of RPE signal [1]. However, during working memory tasks as modelled in this paper – and more importantly during learning – the dopaminergic neurons activity has not been recorded to the best of our knowledge. CT-AuGMEnT can predict, accordingly with other TD learning based models, how this activity should look like. The RPE is generally high at the beginning of the training when an unexpected reward is provided (Fig. 9). During learning, the RPE shifts toward the first clue that is correlated with the upcoming reward. This behavior conforms with data recorded by [1].

4. An accessory feedback network

While CT-AuGMEnT can successfully learn tasks in continuous time, the algorithm requires a feedforward phase followed by a feedback phase after action selection for credit assignment, which is then combined with the reward prediction error to determine synaptic changes. The weight updates result in a biologically plausible implementation of a learning rule that approximates error-backpropagation in standard ANN's, provided that there is enough time for the feedforward and feedback interactions.

As communication between neurons, and between layers of neurons, is not instantaneous, we investigate the possible influence of delays in the feedforward and feedback pathways. [18] already suggested that feedback can be carried by a separate accessory network, similar to the feedback networks proposed by [17] for assigning credit to synapses in lower layers. Additionally, learning should still converge even when the weights of the accessory network are different from that of the feedforward network [18]. We, here, wished to investigate the impact of transmission delays, which create a (partial) temporal mismatch between the forward and backward phases.

We implement an accessory network as shown in Fig. 10a. The accessory network is composed of two layers of neurons – described with the superscript S – connected by randomly initialized weights different from those in the feedforward network (the orange feedback network in Fig. 10a). The accessory network takes as input the executed action determined by the Z-layer, and it carries the feedback signal needed for the weights update of the feedforward network (red arrows in Fig. 10). The neuron with the weakest inhibition is the only unit that provides the feedback signal that gates plasticity:

$$x_a^S = z_a = \delta_{aK}, \tag{25}$$

where δ_{aK} is the Kronecker-delta function and K is the selected action. The association layer activity in the accessory network can be the computed as:

$$y_j^S(t) = \sum_k w_{kj}^S x_k^S(t), \qquad y_m^S(t) = \sum_k w_{km}^S x_k^S(t), \tag{26}$$

where we distinguish between the units that carry feedback to the regular association units j and those that carry feedback to the memory units m. For the accessory network, we adopt a linear transfer function for the neural activation. Although using a linear activation function might be justified for small signals, this is not a strict requirement for CT-AuGMEnT to learn, other saturating functions can be used as well [61]. This network can be used to compute the weights updates; however, due to the transmission delay Δ, the feedback signal can potentially be associated with an earlier stimulus than the one currently processed by the feedforward network (see Fig. 10b).

Both forward and backward weights are modified through learning. We use the same update of the equivalent feedforward neurons as shown in [18], thus Eqs. (20) and (24) for the tags and for the weights update respectively:

$$\begin{aligned} Tag_{kJ}^S(t+dt) &= \left(1 - \tfrac{dt}{\phi}\right) Tag_{kJ}^S(t) + dt[y_J(t)x_k^S(t)] \\ w_{kJ}^S(t+dt) &= w_{kJ}^S(t) + dt[\beta\delta(t) Tag_{kJ}^S(t)], \end{aligned} \tag{27}$$

where the index J denotes either regular or memory units in the association layer.

Effect of transmission delays. We examined how this continuous feedback architecture with an accessory network can learn the previously introduced tasks in the presence of transmission delays.

The same meta–parameters and network architectures are used as before, but now the weights in the feedforward and feedback networks are randomly and differently initialized. We introduce delays between the layers of the feedforward and feedback networks: from the sensory layer to the association layer, from the association layer to the Q-layer of the feedforward network, and between the layers of the accessory network (see Fig. 10, Δ). The simulation step-size was set to $dt = 0.01$ (corresponding to 10 ms per time step) for all the tasks. We evaluate the architecture with various delays $\Delta = [0, 3, 9, 25, 50]$ time-steps with 10 ms per step: the maximum duration of a "round-trip" of the activity experienced by the network is then 1.5 s. Note that there is no delay between the Q-layer and the action selection model: the action selection has its own dynamics, which we fix for all the simulations ($\tau_\rho = 0.5$), except for the MDT task where we used $\tau_\rho = 0.1$. We report results based on runs with 100 randomly initialized networks. For the TM and SaS tasks, the same convergence criterion is used as for the standard implementations; every network is given 2.5×10^4 trials to learn the task; for the DMC and MDT tasks we allowed networks a maximum of 5×10^4 trials to reach the convergence criterium (see Table 3).

We find that CT-AuGMEnT is still able to learn all tasks when an accessory network is used to carry feedback activity of the output layer without delays or when small delays are introduced in between the network's layers: Fig. 11 plots the percentage of converging networks for the three tested tasks. A reduction in convergence rate is evident after about 100 ms of total delay. This is likely due to the mismatch between the forward processing and the feed back information provided by the accessory network during learning. The development feedback activity during learning is detailed in Appendix E.

5. Discussion

We presented CT-AuGMEnT, a biologically plausible framework for continuous-time neural SARSA reinforcement learning with working memory, formulated in the limit of small time-steps in a discrete-time model. For the weight update, we exploited the same principles as in the original time-step formulation of AuGMEnT: a combination of a neuromodulatory signal coding for the temporal difference and an 'attentional' feedback signal which tags those synapses that contributed to action selection [7] – for a detailed review of the neurobiological plausibility and the tagging mechanism and its relation to the "synaptic tagging and capture" theory [62,63] see also [8]. In the final layer, CT-AuGMEnT includes an action-selection system that implements a winner-take-all mechanisms in continuous-time, based on [15]. This action-selection system is a simplified neural architecture modelled after the basal ganglia. Several studies have suggested a role of the basal ganglia in action-selection and in reward-based learning [64–66,15]. Here, the dynamics of the action-selection system, expressed by the action-time constant, help stabilise the action execution by avoiding rapid switches between actions. The time-scale of the action dynamics should depend on the environment of the agent, like

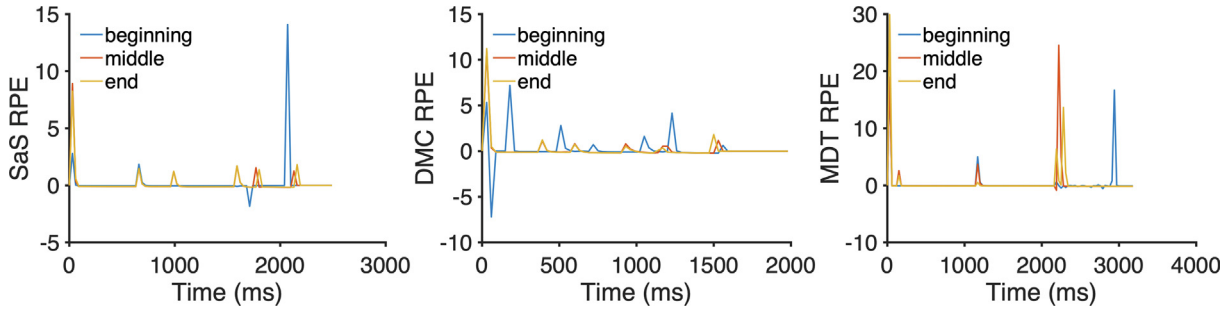

Fig. 9. Reward Prediction Error (RPE) for the three presented tasks (SaS, DMC, MDT, see main text for tasks details) at different moments of the training (begin, halfway, end). The three tasks all show the typical trend of RPE in RL: the RPE peak shifts from the time the agent receives the first unexpected reward to the beginning of the trial. For all tasks, only correct trials for one condition were considered. For SaS, among the 25 k training epochs, the antisaccade-right condition succeeded the task 5134 times. For DMC, we considered training for 25 k epochs and plotted the same category and same angle of motion (45 deg), obtaining 177 correct choices. For MDT, we trained for 20 k epochs and we considered the two choice task were the correct motion was at 135 deg with max coherence, this condition was correct 343 times.

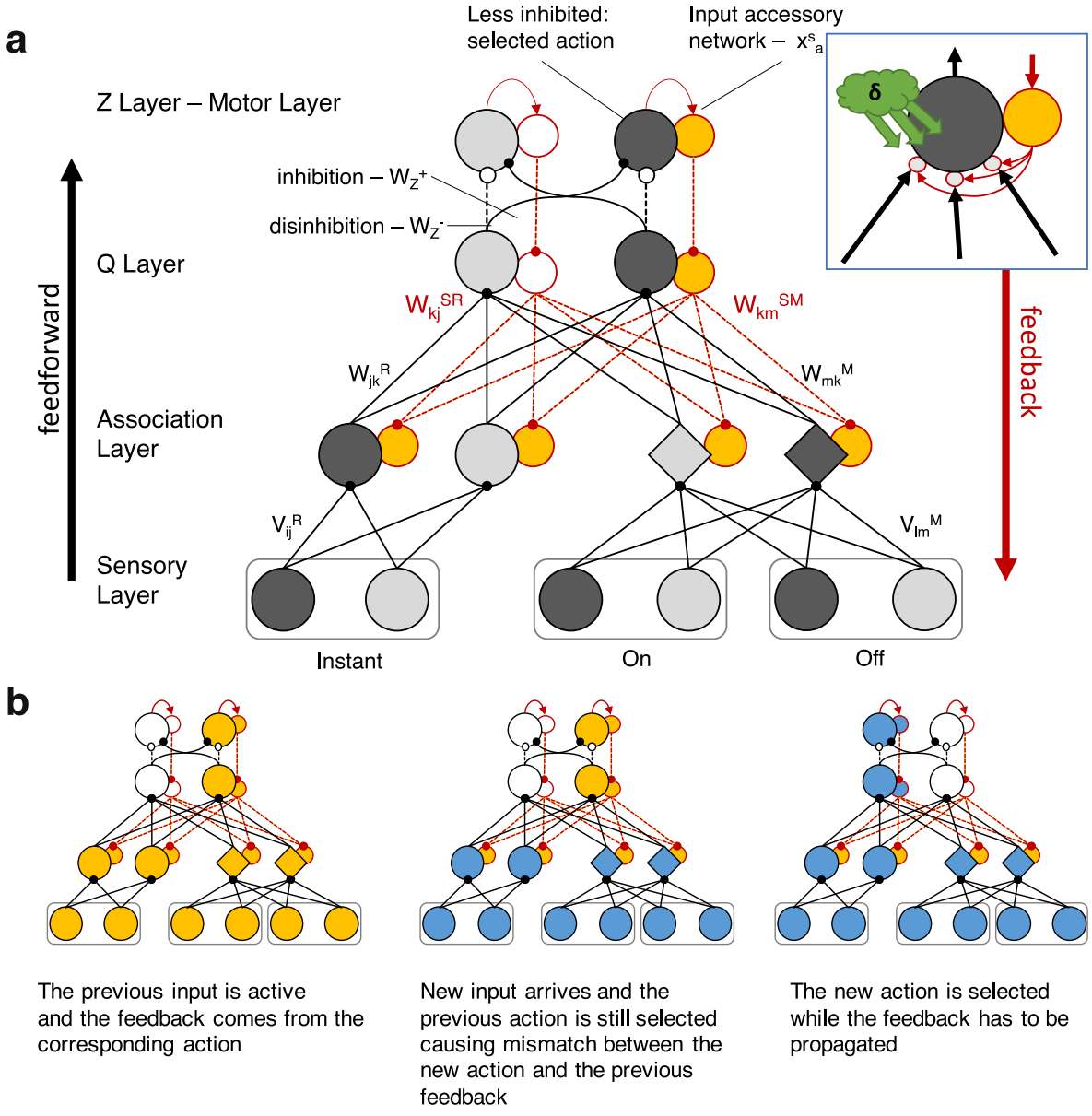

Fig. 10. **a** the feedforward (grey units: darker for more activation) and the accessory (yellow units) networks. The arrow direction signals the information flow. In the inset: the feedforward synaptic weights are modified according to the global TD error signal and the activation of the accessory units. **b** the mismatch between the feedforward and feedback activities due to the transmission delays Δ: a new input (cyan) is propagated in the feedforward direction while feedback from the previous selection (yellow) is used to update the synaptic weights.

Table 3
Summary of convergence results when we varied the transmission delay, measured as the number of trials required for learning the task. Delay between layers is measured in units of the timesteps of $dt = 0.01$ (10 ms).

Task	Total Roundtrip Duration Δ (ms)				
	0Δ	3Δ(90ms)	9Δ(270ms)	25Δ(750ms)	50Δ(1500ms)
SaS	4176 ± 251	4469 ± 295	5061 ± 286	4261 ± 165	3625 ± 78
DMC	3287 ± 301	3706 ± 422	4088 ± 458	3722 ± 331	5847 ± 0
MDT	20345 ± 3144	21301 ± 3467	21021 ± 3902	25783 ± 4049	22791 ± 3305

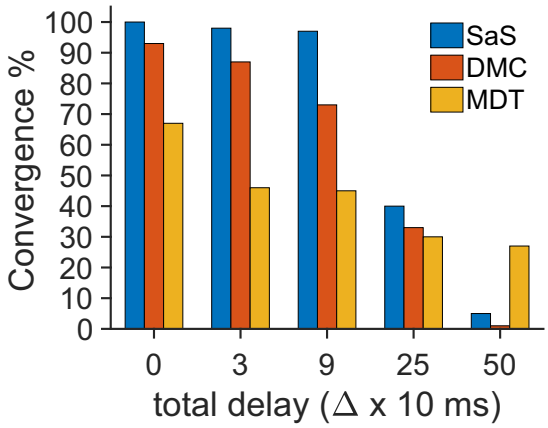

Fig. 11. Percentage of networks that converge versus transmission delays for the Saccade/Anti-Saccade task (blue bars), DMC task (red bars), and MDT task (yellow bars): for increasing delays, the agent increasingly fails to learn the task. Transmission delays Δ are measured as the number of 10 ms timestep.

the speed of muscle recruitment, and decouples these dynamics from the time-step in the network.

The action-selection system is endowed with a built-in exploration mechanism that is linked to the action dynamics. Exploration overrides the presently selected action by providing an additional input to the explorative action, related to what has been reported in humans [51]; the exploration duration depends on the action time-constant, allowing more exploration for longer-duration actions. This in turn helps the learning algorithm by ensuring that, during the weight updates, the correct amount of credit is assigned to the responsible weights in the neural network.

CT-AuGMEnT correctly learns the various cognitive tasks we presented when the time resolution of the simulation increases. We compared CT-AuGMEnT to AuGMEnT on three working-memory tasks that probe various aspects of task difficulty and decision making. Our results demonstrate that CT-AuGMEnT reaches the convergence criterion for every time-step duration we tested, while, whereas AuGMEnT usually did not reach convergence: CT-AuGMEnT needed a constant or moderately increasing number of trials to reach convergence when the time resolution increased.

We further demonstrated the ability of CT-AuGMEnT to train networks to learn to make a decision. The CT-AuGMEnT architecture can be seen as a simplified version of the principal structures involved in the decision-making process in the brain – cortex and basal ganglia. Since we are using the basal ganglia model of [15], this results in a (slightly) sub-optimal decision making model compared to the model for optimal multihypothesis sequential probability ratio test (MSPRT) derived in [42]; we found empirically in the tasks we studied that using this MSPRT model led to slightly but insignificantly worse performance. We speculate this may be related to the model of action-dynamics we use, or, alternatively, the type of tasks we study are just not very sensitive to the difference.

Importantly, CT-AuGMEnT is a plausible explanation of how working memory units *learn* what to accumulate during reinforcement learning, which is necessary to learn decision-making tasks. Moreover, different from the approach in [65], CT-AuGMEnT learns in continuous-time, which is a necessary feature when modelling time-dependent variables such as RTs, and enables CT-AuGMEnT to model RT patterns in decision-making problems with multiple alternatives. The dynamics of the action-selection system also allows for the prediction of RTs without need to set an arbitrary threshold as in the standard race models typically used to study this behaviour [67,65,42].

Our results show a good match between the accuracy and reaction times of monkeys and the network's performance. In this context, the working memory units learn to act as integrators for the perceptual evidence. Importantly, we did not modify the structure of the network, which was able to learn what to accumulate using only the reward signal that was only given at the end of the trial, highlighting the efficiency of the credit assignment process.

We studied the same tasks when a separate accessory network was used to carry the feedback signals [17,30]. We demonstrated that CT-AuGMEnT correctly learns all tasks when the feedback accessory network is used, even when the forward and feedback networks were randomly initialized to different values. Using a fixed random network, as in [30] results in a drop of performance for hard tasks [68]. Our formulation requires the same update in corresponding forward and backward connections, which can be implemented with biological networks [68,10]. We introduced transmission delays between the layers of the two networks to understand the limits of feedback during continuous-time RL problems in biological settings, where neural transmission and response dynamics introduce such delays. Our results show that CT-AuGMEnT is still able to learn the tasks when small delays are introduced – from 90 to 270 ms total delay, compatible with biological delays [69]; for larger delays, the mismatch between the feedforward and the feedback signals affects the network's performance.

The present study focused on the learning process and we did not specifically model the neuronal interactions that are responsible for maintaining a scalar value in working memory, which has been addressed in previous work [70]. One of these models by [58] designed a mechanism that allows the same network to store the memory for a sensory stimulus during a delay and to commit to a decision at a later point in time. The present study goes beyond these previous findings by demonstrating that CT-AuGMEnT can discover a similar mechanism during trial-and-error learning.

Still, our model does not fully capture the dynamics of the working memory cells, as done, for example, in the attractor dynamic model proposed by [58]. In their work, working memory and the decision are represented by one single variable, while here, they are represented by two different layers of neurons. This implies that, with respect to the biological cells, our working memory units do not reproduce the commitment to the decision, although the decision indirectly affected their value through the learning rule. It would be interesting for further studies to combine the attractor dynamics model with the CT-AuGMEnT learning rule to fully explain this behaviour. Finally, in the present study, working memory units are not explicitly controlled and they have to be

reset at the end of each episode. Other works have addressed this problem [24,25] and a similar mechanism can be implemented in CT-AuGMENT as suggested in [55]. Our model only implicitly deals with perceptual uncertainty, unlike explicit approaches like Bayesian networks [45]. An extension of our model toward Bayesian inference could possibly be implemented using dropout sampling [71,72] or related sampling methods [73].

The model as presented is to the best of our knowledge the first example of a plausible end-to-end neural network model for learning decision making tasks using SARSA. Such a formulation makes it possible to consider implementations based on spiking neurons and compare such spike-based models directly to measurements of the time-course of neural activity.

CRediT authorship contribution statement

Davide Zambrano: Conceptualization, Methodology, Software, Writing - original draft. **Pieter R. Roelfsema:** Writing - review & editing, Supervision. **Sander Bohte:** Conceptualization, Writing - review & editing, Supervision, Project administration, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

DZ is supported by NWO (Ideas grant 656.000.005, "DEVIS"), and PRR was supported by NWO (Ideas grant 656–000-002 "REASON"), and the EuropeanUnion (grant agreement 7202070 "Human Brain Project", and ERC grant agreement 339490 "Cortic_al_gorithms").

Appendix A. Condition for optimality in continuous-time

Following [33], we consider the continuous-time system,

$$\dot{\mathbf{s}}(t) = f(\mathbf{s}, \mathbf{a}(t)), \tag{28}$$

where $\mathbf{s}$ is the state at time t, and $\mathbf{a}$ is the selected action. Here we are interested in the time derivative of the Q-values, thus, since both $\mathbf{s}$ and $\mathbf{a}$ depend on time, we make a variable change as $\mathbf{x}(t) = [\mathbf{s}(t), \mathbf{a}(t)]$. According to the principle of optimality, the integral in Eq. (3), can be divided in two parts $[t, t+dt]$ and $[t+dt, \infty]$ as:

$$Q^*(\mathbf{x}(t)) = \max_{\mathbf{a}[t,t+dt]} \left[\int_t^{t+dt} \mathrm{e}^{-\frac{\zeta - t}{\tau}} r(\mathbf{x}(\zeta)) d\zeta + \mathrm{e}^{-\frac{dt}{\tau}} Q^*(\mathbf{x}(t+dt)) \right]. \tag{29}$$

For a small dt, the first term is approximated as:

$$r(\mathbf{x}(t))dt + o(dt), \tag{30}$$

while, by expanding through Taylor and applying the chain rule, the second term is:

$$\begin{aligned} Q^*(\mathbf{x}(t+dt)) &= Q^*(\mathbf{x}(t)) + \frac{\partial Q^*}{\partial t} dt + o(dt) \\ &= Q^*(\mathbf{x}(t)) + \frac{\partial Q^*}{\partial \mathbf{x}(t)} f(\mathbf{x}(t)) dt + o(dt). \end{aligned} \tag{31}$$

By substituting them into (29) and collecting Q^* on the left hand side, we have an optimality condition for $[t, t+dt]$ as:

$$\left(1 - \mathrm{e}^{-\frac{dt}{\tau}}\right) Q^*(\mathbf{x}(t)) = \max_{\mu} \left[r(\mathbf{x}(t))dt + \mathrm{e}^{-\frac{dt}{\tau}} \frac{\partial Q^*}{\partial \mathbf{x}(t)} f(\mathbf{x}(t))dt + o(dt) \right]. \tag{32}$$

By dividing both sides by dt and taking dt to zero, and by reapplying the variable change, we have the condition for the optimal action-value function as:

$$\frac{1}{\tau} Q^*(\mathbf{s}(t), \mathbf{a}(t)) = \max_{\mu} \left[r(\mathbf{s}(t), \mathbf{a}(t)) + \frac{\partial Q^*(\mathbf{s}(t), \mathbf{a}(t))}{\partial \mathbf{s}(t), \mathbf{a}(t)} f(\mathbf{s}(t), \mathbf{a}(t)) \right]. \tag{33}$$

Appendix B. Continuous-Time AuGMEnT stochastically minimises the reward-prediction error (RPE)

We can show that the continuous-time formulation of AuGMEnT outlined above reduces the reward-prediction error (RPE) as the original AuGMEnT formulation. The objective function is defined as:

$$E(t) = \frac{1}{2} \left| \delta(t)^2 \right|. \tag{34}$$

Given (18) and (7), the gradient of the objective function with respect to the weights w^R_{ja} becomes:

$$\begin{aligned} \dot{w}^R_{ja} &= -\beta \tfrac{\partial E(t)}{\partial w^R_{ja}} = \\ &= -\beta \delta(t) \tfrac{\partial}{\partial w^R_{ja}} \left[r(t) - \tfrac{1}{\tau} q_{a'}(t) + \dot{q}_a(t) \right] \\ &= \beta \delta(t) \left[+ \tfrac{1}{\tau} \tfrac{\partial q_{a'}(t)}{\partial w^R_{ja}} - \tfrac{\partial \dot{q}_a(t)}{\partial w^R_{ja}} \right] \\ &= \beta \delta(t) \left[+ \tfrac{1}{\tau} \tfrac{\partial q_{a'}(t)}{\partial w^R_{ja}} - \tfrac{1}{dt} \tfrac{\partial (q_{a'}(t) - q_a(t-dt))}{\partial w^R_{ja}} \right] \\ &= \beta \delta(t) \tfrac{1}{dt} \left[-(1 - \tfrac{dt}{\tau}) \tfrac{\partial q_{a'}(t)}{\partial w^R_{ja}} + \tfrac{\partial (q_a(t-dt))}{\partial w^R_{ja}} \right], \end{aligned} \tag{35}$$

where $\beta$is the learning rate.

Since the boundary condition for the Q-function, defined in (3), is given at $t \to \infty$, it is more appropriate to update the past estimates without affecting the future estimates as in [33]. Thus, recalling (8) and discretizing (35), a reduction of the gradient is guaranteed if:

$$\begin{aligned} \dot{w}^R_{ja} &= \beta \delta(t) \tfrac{\partial q_a(t-dt)}{\partial w^R_{ja}} \\ &= \beta \delta(t) y^R_j(t-dt). \end{aligned} \tag{36}$$

which is consistent with the trace update in (20) for $a = 1$. The same can be shown for the synapses between memory units M and Q-values:

$$\dot{w}^M_{ma} = -\beta \frac{\partial E(t)}{\partial w^M_{ma}} = \beta \delta(t) y^M_m(t-dt). \tag{37}$$

Note that in the latter equations the update of the synapses has to be consistent with the neuron activity at the previous dt, which is stored in the Tags (see (20) and (21)). Thus, Tags have to be updated after the weights. Gradient decent for the weights v^R_{ij} is similarly computed:

$$\begin{aligned} \dot{v}^R_{ij} &= -\beta \tfrac{\partial E(t)}{\partial v^R_{ij}} \\ &= \beta \delta(t) \tfrac{\partial q_a(t-dt)}{\partial v^R_{ij}} \\ &= \beta \delta(t) \tfrac{\partial q_a(t-dt)}{\partial y^R_j(t-dt)} \tfrac{\partial y^R_j(t-dt)}{\partial inp^R_j(t-dt)} \tfrac{\partial inp^R_j(t-dt)}{\partial v^R_{ij}} \\ &= \beta \delta(t) w'^R_{aj} \sigma'(inp^R_j(t-dt)) x_i(t-dt) \end{aligned} \tag{38}$$

and for v^M_{lm}:

$$\begin{aligned} \dot{v}^R_{ij} &= -\beta \frac{\partial E(t)}{\partial v^M_{lm}} \\ &= \beta\delta(t) \frac{\partial q_a(t-dt)}{\partial v^M_{lm}} \\ &= \beta\delta(t) \frac{\partial q_a(t-dt)}{\partial y^M_m(t-dt)} \frac{\partial y^M_m(t-dt)}{\partial inp^M_m(t-dt)} \frac{\partial inp^M_m(t-dt)}{\partial v^M_{lm}} \\ &= \beta\delta(t) w'^M_{am} \sigma'(inp^M_m(t-dt)) sTrace_{lm}(t-dt), \end{aligned} \tag{39}$$

where we assume for simplicity that the strength of the feedback from the motor layer back to the association layer w^R_{aj} is equal to w^R_{ja} and, analogously, $w^M_{am} = w^M_{ma}$.

Appendix C. Additional tasks

The Motion-or-Colour task (MoC) has been used to train monkeys to study decision-making under two different contexts [57]. Here, the agent has to attend one of two features of the same random-dot stimulus, either the colour or the motion direction, based on context indicated by the shape of the fixation mark. This task combines the continuous collection of evidence with various degree of motion coherence (as for the 2&4 choice task, with a non linear mapping between inputs and outputs, as in the SaS task. The task phases are shown in Fig. A3.1. The fixation mark – indicating the context – is shown for 300 ms, then the two targets are presented for 350 ms. While the agent has to maintain fixation on the fixation mark (for which it receives $r_{fix} = 0.2$), the stimulus is presented for 750 ms. The dots have a particular colour coherence $[-50, -16, -6, 6, 16, 50]$ and motion coherence $[-50, -16, -6, 6, 16, 50]$; where a colour of -50 denotes a clearly distinguishable red and 50 a clearly distinguishable green and values closer to zero have less coherence. Similarly, -50 and 50 denote a strong motion signal to the left and right, respectively. After the stimulus presentation, a delay phase of 1s follows. The disappearance of the fixation mark indicates the "Go" signal, requiring the agent to make one of two responses. In this task, the network uses 10 input units: 2 fixation marks, 2 indicating the colour stimulus, 2 for the motion stimulus, and 4 targets. The network also contains 5 regular units, 5 memory units and 3 output units. We pre-trained the model in the colour and motion tasks separately, with full coherence $[-50, 50]$. The pre-training stopped when the agent performed correctly on 90% of the last 50 trials for each condition. The full task is stopped when the agent reaches 85% of accuracy on the 50 trials in the conditions with a coherence of $[-50, -16, 16, 50]$ (see Table A3.1).

The results are obtained for the motion-or-colour task also reproduced many aspects of the monkeys' data (see Fig. A3.2). After learning, the agent correctly discriminates between the two relative features (colour or motion) depending on the current context. Fig. A3.2 shows that the non-relevant features do not affect the performance, whereas the coherence of the relevant feature determines the agent's accuracy.

The T-Maze (TM) task is a working memory RL task adapted from [12,55], where an agent has to reach a goal position at the end of a corridor and the location of the goal depends on the location of the road-sign observed at the start of the task (Fig. A3.3a). Hence, the agent has to learn to keep information in memory for multiple time-steps and where the difficulty of the task can be adapted by changing the corridor length.

The agent's position is defined by a two-dimensional coordinate system $[x, y]$. The agent starts from position $[0, 0]$, and it can select

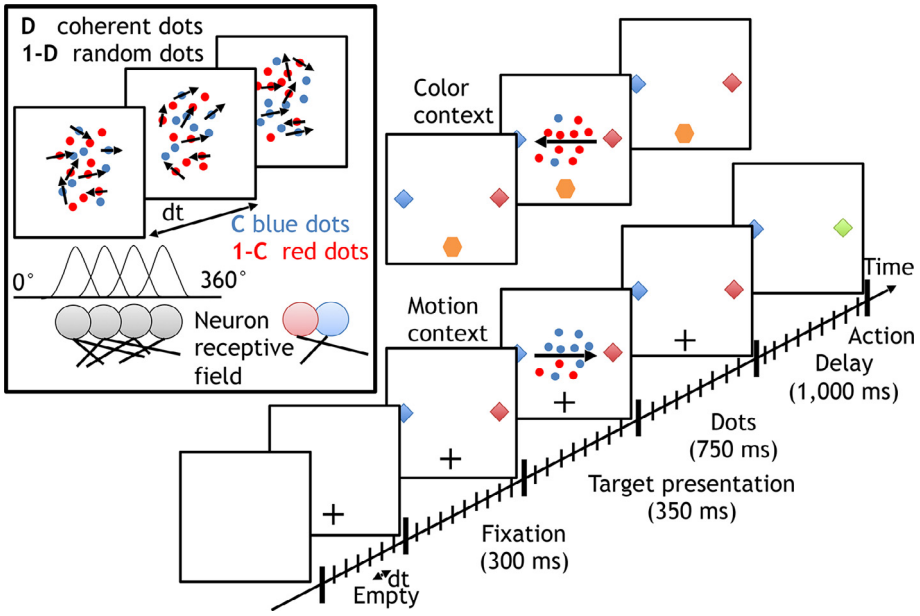

Fig. A3.1. Motion-or-colour task. The agent has to discriminate the colour or the motion of the presented dots, depending on the context cue – a cross for motion and a hexagon for colour – by making an eye movement to one of the two targets (blue or red diamond). In the inset: the dots have a motion direction and a colour.

Table A3.1
Summary of the MoC task configuration.

Task	Architecture	dt	τ_ρ	Rewards	Trained Nets	Convergence Condition	Pre-training Condition
MoC	10 In (2 Contexts, 2 Colour, 2 Motion, 4 targets), 4 R, 5 M, 3 Out (Fixate, Left, Right)	0.1, 0.03	0.5	$r_{fix} = 0.2$, $r_{fin} = 1.5$	100	85% in the last 50 trials	90% in the last 50 trials

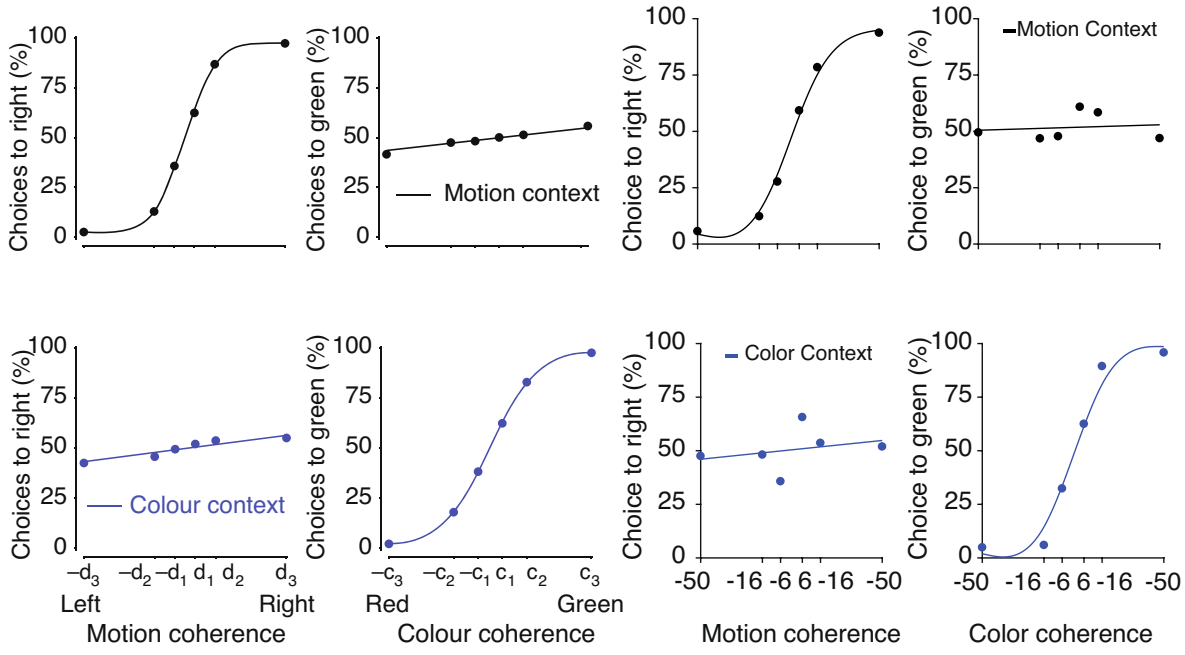

Fig. A3.2. Comparison between original data [57] (left two columns) and model results (right two columns) for the motion-or-colour. Similar to the DMC results, the accuracy increases with incr.eased coherence.

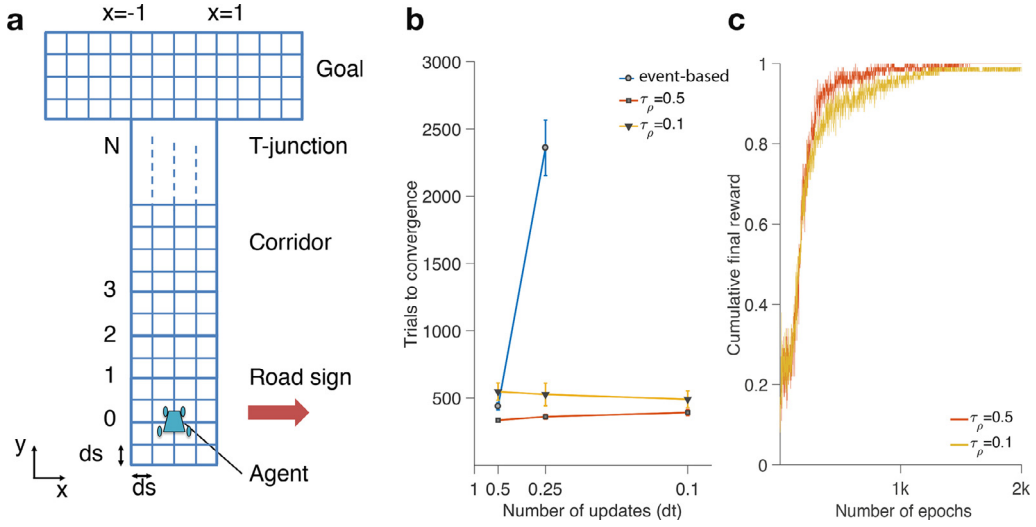

Fig. A3.3. **a** T-Maze. The agent has to reach the Goal location while remembering the Road sign observed during the first second of the trial. The agent moves with small steps of size $ds = dt$. **b** Comparing the event-based version of AuGMEnT (blue line, event-based) and CT-AuGMEnT (red line $\tau_\rho = 0.5$ and yellow line $\tau_\rho = 0.1$). Plotted is the number of trials needed to reach convergence for the task averaged for those networks that converged; the abscissa denotes the effective size of dt used for the simulations. **c** Comparing the learning curves for the two action time constants (red line $\tau_\rho = 0.5$ and yellow line $\tau_\rho = 0.1$) at $dt = 0.1$. The learning curve is computed as the ratio between cumulative sum of the final reward and the total number of trials, averaged over all the trained networks.

among four actions *Up, Down, Left* or *Right*; the agent cannot turn. During a trial, the agent's position is updated according to the currently selected action as: AgentPosition+ $= ds \cdot$ [Up$-$Down, Right $-$ Left], where the position is increased by a step-size, ds, proportional to dt as $ds = dt$ (e.g. with $dt = 0.1$ it takes 10 steps to move 1 cell). To ensure a consistent comparison between the event-based version of AuGMEnT and CT-AuGMEnT, the task is adapted to be identical for both algorithms, where the length of the corridor was fixed at $N = 10$. Walls are hit when the x position is $\geqslant 1$ or $\leqslant -1$. The agent has 3 sensory inputs where 1 represents a wall and 0 an empty space: in the corridor it thus sees $[1, 0, 1]$. For the first second, the agent observes $[2, 0, 1]$ or $[1, 0, 2]$, where a 2 denotes the road sign. A attempted move through the wall returns a negative reward of r_w: to avoid excessive collection of negative rewards when dt decreases, movement into the wall returns one r_w per second, i.e. this punishment is proportional to the time spent moving into the wall. At the T-junction, the agent is rewarded with $r_g = 4$ if it moves in the same direction

Table A3.2
Summary of the TM task configuration.

Task	Architecture	dt	τ_ρ	Rewards	Trained Nets	Convergence Condition	Pre-training Condition
TM	In = 3, R = 3, M = 4, Out = 4 (Up, Down, Left, Right)	0.5, 0.25, 0.1	0.5, 0.1	$r_w, r_b = -0.1, r_g = 4$	150	90% in the last 50 trials	None

Table A3.3
Summary of the results for the comparison between AuGMEnT and CT-AuGMEnT for TM.

Task	dt	AuGMEnT		CT-AuGMEnT $\tau_\rho = 0.5$		CT-AuGMEnT $\tau_\rho = 0.1$	
		Accuracy (%)	Trials	Accuracy (%)	Trials	Accuracy (%)	Trials
TM	0.5	99	439 ± 25	100	335 ± 13	98	548 ± 64
	0.25	94	2359 ± 206	99	361 ± 17	99	528 ± 82
	0.1	4	n.c.	96	392 ± 22	96	491 ± 62

Table A3.4
Results for the T-Maze comparison.

Task	dt	softmax CT-AuGMEnT		CT-LSTM		CT-AuGMEnT $\tau_\rho = 0.5$	
		Accuracy (%)	Trials	Accuracy (%)	Trials	Accuracy (%)	Trials
TM	0.5	81	1208 ± 421	100	79900 ± 9397	100	335 ± 13
	0.25	81	1897 ± 514	30	226900 ± 57997	99	361 ± 17
	0.1	76	1129 ± 382	0	n.c.	96	392 ± 22

as the road-sign was posted. An incorrect choice returns $r_b = -0.1$. The agent's x position determines its choice: as soon as it crosses $+1$ or -1 its decision is evaluated. We imposed a time-restriction condition proportional to the task difficulty N, which was $1.5N + 2$ time-steps; if the network did not reach the correct corridor within this time, the trial was aborted and no reward was obtained (see Table A3.2 for details).

For the T-Maze task, the number of trials needed to converge for both AuGMEnT and the CT-AuGMEnT is plotted in Fig. A3.3b. The number of trials needed to reach convergence rapidly increases for event-based AuGMEnT with increasing time resolution, and AuGMEnT quickly fails to converge at all (see Table A3.3). This illustrates the problem with learning action-values noted by [35] already: as the time resolution increases, the effect of a single infinitesimal action on the total reinforcement becomes undetectable. However, CT-AuGMEnT successfully learns the tasks for every dt and for two different values of the action time-constant τ_ρ. When reducing dt, for CT-AuGMEnT the number of trials needed to reach convergence remains constant and convergence remains near 100%. Fig. A3.3c plots the learning curves for the T-Maze task for different action time-constants: the learning curves are highly similar, with a slight advantage for the longer action time-constant. Effectively, these results show that CT-AuGMEnT is independent of the size of dt.

We also trained CT-AuGMEnT with a strict softmax action selection policy [51]: we observe that then CT-AuGMEnT has lower convergence rate and needs more trials, on average, to reach the convergence criteria (see also Appendix D). Since the network needs to learn proper action-selection from randomly initialized weights, a more conservative exploration strategy like Max-Boltzmann seems beneficial. To compare our results to a non-biologically-inspired algorithm, we trained an LSTM-based network that uses with Advantage Learning [12] in the continuous-time RL setting. We find that such an CT-LSTM does not converge

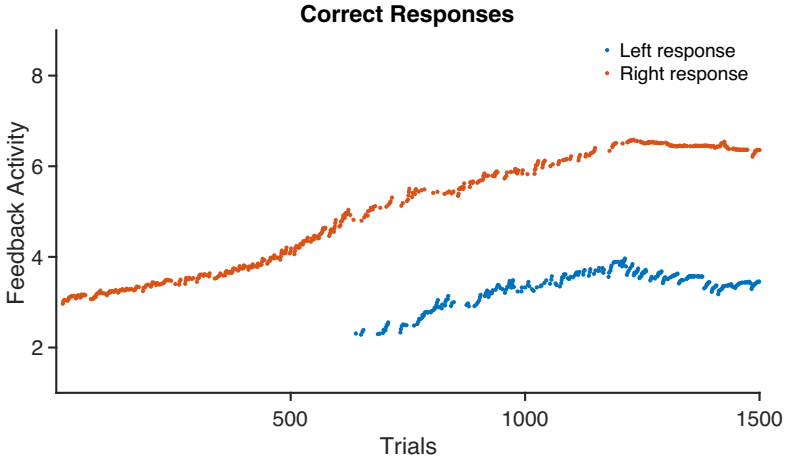

Fig. A4.1. Learning shapes attentional feedback from the final response selection stage. Summed attentional feedback arriving at memory units encoding information about in correct trials for the left responses (pro-left and anti-left conditions, blue) versus correct trials for right responses (pro-right and anti-right, red) throughout learning.

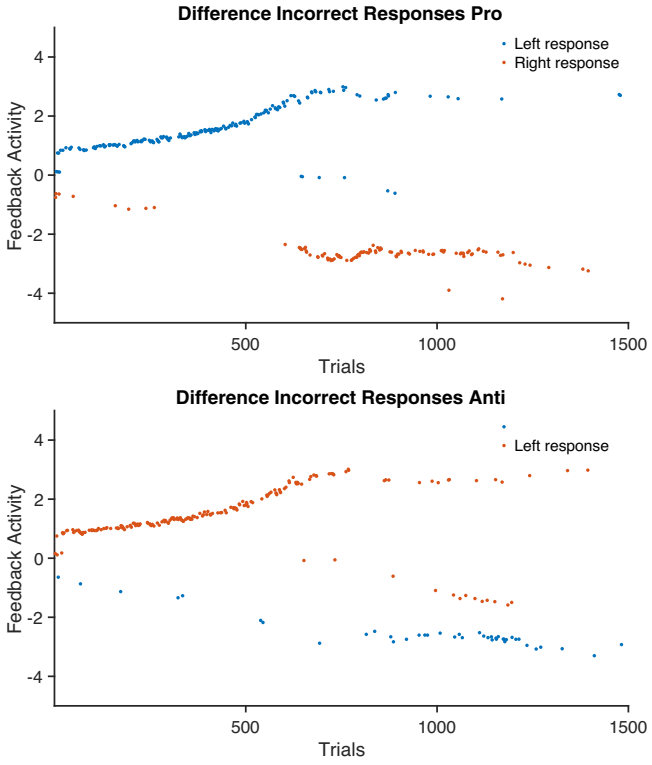

Fig. A4.2. The difference in average feedback activty between the incorrect and correct responses for the pro-saccade condition (top) and anti-saccade condition (bottom).

for small *dt* without any action-duration in the learning algorithm (details Appendix D). This reinforces our finding of the higher number of trials that CT-AuGMEnT needs for $\tau_\rho = 0.1$: the fast switching among actions increases the complexity of the temporal credit assignment problem.

Validation experiments on TM. Here we give the results for additional experiments on the T-Maze task to investigate the effect of our action selection policy and to compare CT-AuGMEnT to continuous-time LSTM. To examine the effect of the action selection policy, we replaced the Max-Bolzmann selection rule by a softmax rule. We trained 100 networks with Eq. (13) changed to:

$$P_B(\mathbf{a}) = \frac{\exp(q_a)/Temp}{\sum_k \exp(q_k)/Temp},$$

with the temperature parameter, *Temp* set to $5e10^{-}3$. We found, however, that this softmax rule prevents the algorithm from converging. Indeed, at the beginning of the task, the network needs to learn the input representation and a large amount of exploration – due to very similar q-values – is counterproductive. Therefore, for the first 150 trials we used the Max-Boltzmann action selection policy and then switched to softmax. All other settings remained unchanged. The results are shown in Table A3.4, left column. The softmax CT-AuGMEnT exhibits a lower convergence rate and requires more trials, on average, than CT-AuGMEnT to reach the convergence criteria.

The second set of experiments uses LSTM with Advantage Learning as reported in [12]. Since Advantage Learning can approximate continuous time, these experiments become an effective validation for our algorithm. Following [12], we used 12 standard units and 3 LSTM units, the learning rate $\alpha = 0.0002, \gamma = 0.98$, $\lambda = 0.8$, $\kappa = 0.1$ and we trained 10 networks for 500*k* trials. Moreover, we scaled the Eligibility Traces with a similar approach used for our Tags (see Eq. (19)) (not scaling the Traces yielded similar negative results). As shown in Table A3.4, despite the large number of trials, CT-LSTM does not converge for small *dt*, likely due to the lack of extended action-duration in the learning algorithm.

Appendix D

To visualize how the accessory network develops during training, we show the activity of the neurons in the association layer (see Eq. (26)), for the Saccade-Anti-Saccade Task, similar to [74]. Fig. A4.1 shows the sum of the feedback activity for the units $y^S_m(t)$ that carry the feedback from the selected action in the Q-layer to the memory units. This activity is collected at the end of every trial, and shown for one network. We see that the feedback

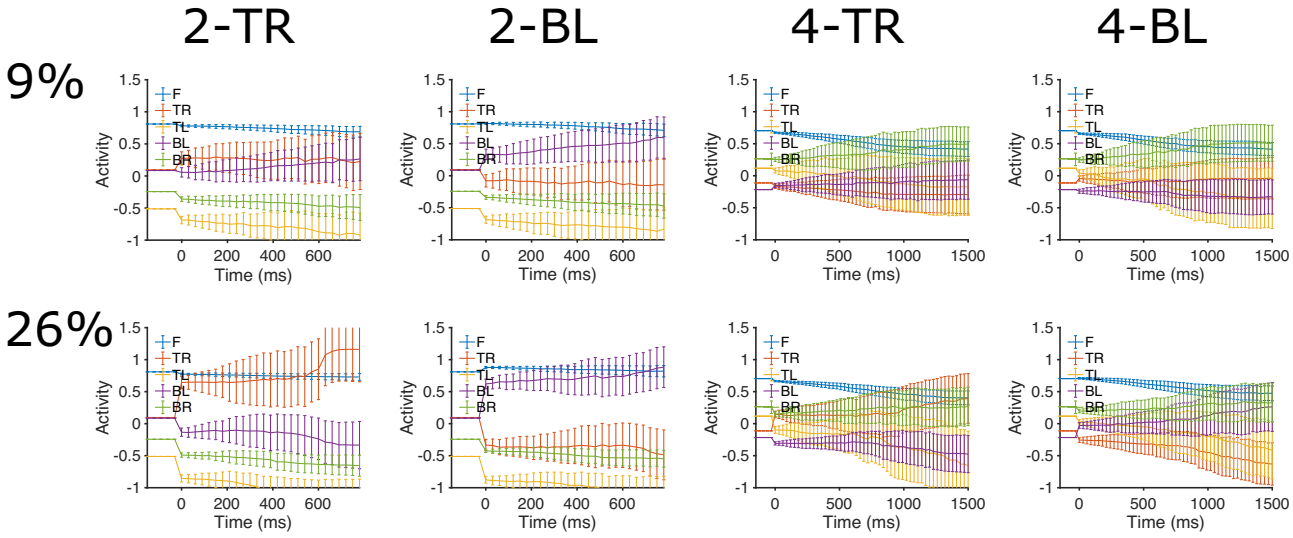

Fig. A5.1. Uncertainty during MDT for two motion coherences, **9%** top row, and **26%** bottom row. The column names represent the conditions: two or four choices, **2, 4**, and the corresponding correct action, top or bottom **T,B**, left or right **L,R**. The plot shows average and standard deviation of 5 Q-values during time (zero is the motion onset) across 10 k test trials. In the four choices task, the Q-values are much closer to each other representing higher uncertainty. It also corresponds generally to longer decision times. For that, a longer time domain is reported showing the building up of the correct choice through time.

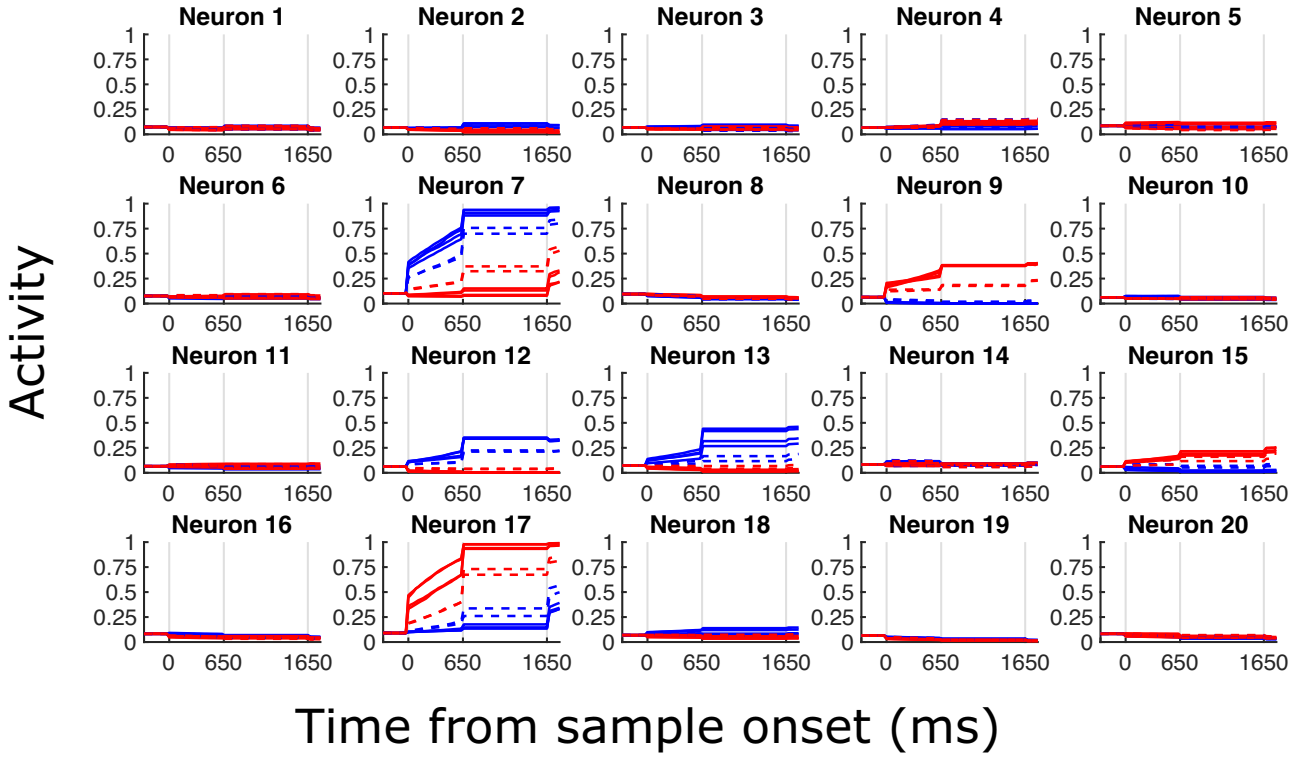

Fig. A5.2. DMC with 20 memory units. When increasing the number of memory units, a few units still show category preference and contribute to the task. Two neurons are strongly active when a specific category is shown (7 and 17), four others have lower activity (9, 12, 13 and 15). Note that the order is preserved in all of them, showing more uncertainty for motion near the category boundary (dashed lines).

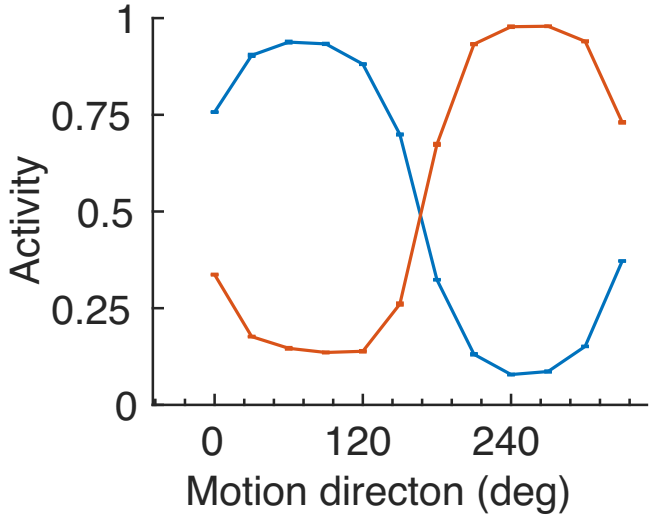

Fig. A5.3. activity of two memory cells in the CT-AuGMEnT network as a function of the 12 motion directions presented.

activity increases during training and then stabilizes. The graph demonstrates that learning changes the attentional feedback during training. Similarly, Fig. A4.2 shows that the *difference* between incorrect responses and the correct choice also grows during training, as is to be expected since the average feedforward response for correct choices also increases relative to incorrect choices. The difference is computed between the actual feedback (incorrect) and what should have been the correct one (left or right) in case of pro-saccade (top) and anti-saccade (bottom) conditions. As expected, the number of incorrect responses decreases during training. Note that an incorrect response could also be a fixation or even the correct one selected in an incorrect time: too early, breaking the fixation or too late, when the time-out condition is applied. In this case, the difference is approximately zero. Overall these Figures demonstrate that the network learns the attentional feedback to use during the task. (see Fig. A5.1)

Appendix E. Extended analysis on the DMC and MDT tasks

This appendix adds some more insights on the DMC and MDT tasks.

References

[1] W. Schultz, P. Dayan, P.R. Montague, A neural substrate of prediction and reward, Science 275 (5306) (1997) 1593–1599.
[2] W. Schultz, Getting formal with dopamine and reward, Neuron 36 (2) (2002) 241–263.
[3] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, 1998.
[4] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, Human-level control through deep reinforcement learning, Nature..
[5] D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, Mastering the game of Go with deep neural networks and tree search, Nature 529 (7587) (2016) 484–489.
[6] J. Rombouts, S.M. Bohte, P.R. Roelfsema, Neurally plausible reinforcement learning of working memory tasks, Advances in Neural Information Processing Systems 25 (2012) 1880–1888.
[7] J.O. Rombouts, S.M. Bohte, P.R. Roelfsema, How Attention Can Create Synaptic Tags for the Learning of Working Memories in Sequential Tasks, PLoS Computational Biology 11 (3)..
[8] P.R. Roelfsema, A. Holtmaat, Control of synaptic plasticity in deep cortical networks, Nature Reviews Neuroscience 19 (3) (2018) 166.

[9] B.A. Richards, T.P. Lillicrap, P. Beaudoin, Y. Bengio, R. Bogacz, A. Christensen, C. Clopath, R.P. Costa, A. de Berker, S. Ganguli, et al., A deep learning framework for neuroscience, Nature Neuroscience 22 (11) (2019) 1761–1770.
[10] I. Pozzi, S. Bohté, P. Roelfsema, A biologically plausible learning rule for deep learning in the brain, arXiv preprint arXiv:1811.01768..
[11] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (8) (1997) 1735–1780.
[12] B. Bakker, Reinforcement learning with long short-term memory, in: T. Dietterich, S. Becker, Z. Ghahramani (Eds.), Advances in Neural Information Processing Systems 14, MIT Press, 2002, pp. 1475–1482.
[13] G. Bellec, D. Salaj, A. Subramoney, R. Legenstein, W. Maass, Long short-term memory and learning-to-learn in networks of spiking neurons, Advances in Neural Information Processing Systems (2018) 787–797.
[14] D. Zambrano, P.R. Roelfsema, S.M. Bohte, Continuous-time on-policy neural reinforcement learning of working memory tasks, IJCNN 2015 (2015).
[15] K.N. Gurney, T.J. Prescott, P. Redgrave, A computational model of action selection in the basal ganglia. I.A new functional anatomy, Biological Cybernetics..
[16] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nature 323 (6088) (1986) 533–536.
[17] D. Zipser, D.E. Rumelhart, The neurobiological significance of the new learning models, in: Computational neuroscience, MIT Press, 1993, pp. 192–200..
[18] P.R. Roelfsema, A. van Ooyen, Attention-gated reinforcement learning of internal representations for classification, Neural Computation 17 (10) (2005) 2176–2214.
[19] S. Bartunov, A. Santoro, B. Richards, L. Marris, G.E. Hinton, T. Lillicrap, Assessing the scalability of biologically-motivated deep learning algorithms and architectures, Advances in Neural Information Processing Systems (2018) 9368–9378.
[20] C. Padoa-Schioppa, J.A. Assad, Neurons in the orbitofrontal cortex encode economic value, Nature 441 (7090) (2006) 223–226.
[21] G. Morris, A. Nevet, D. Arkadir, E. Vaadia, H. Bergman, Midbrain dopamine neurons encode decisions for future action, Nature Neuroscience 9 (8) (2006) 1057–1063.
[22] Y. Niv, N.D. Daw, P. Dayan, Choice values, Nature Neuroscience 9 (8) (2006) 987–988.
[23] A.G.E. Collins, M.J. Frank, How much of reinforcement learning is working memory, not reinforcement learning? A behavioral, computational, and neurogenetic analysis, The European Journal of Neuroscience 35 (7) (2012) 1024–1035.
[24] M.T. Todd, Y. Niv, J.D. Cohen, Learning to Use Working Memory in Partially Observable Environments through Dopaminergic Reinforcement, in: D. Koller, D. Schuurmans, Y. Bengio, L. Bottou (Eds.), Advances in Neural Information Processing Systems 21, 2009, pp. 1689–1696..
[25] K. Lloyd, N. Becker, M.W. Jones, R. Bogacz, Learning to use working memory: a reinforcement learning gating model of rule acquisition in rats, Frontiers in Computational Neuroscience 6 (2011) 87.
[26] H.F. Song, G.R. Yang, X.-J. Wang, Reward-based training of recurrent neural networks for cognitive and value-based tasks, Elife 6 (2017) e21492.
[27] R.J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Machine Learning 8 (3–4) (1992) 229–256.
[28] D. Wierstra, A. Förster, J. Peters, J. Schmidhuber, Recurrent policy gradients, Logic Journal of the IGPL 18 (5) (2010) 620–634.
[29] J.M. Murray, Local online learning in recurrent networks with random feedback, eLife 8 (2019) e43299.
[30] T.P. Lillicrap, D. Cownden, D.B. Tweed, C.J. Akerman, Random feedback weights support learning in deep neural networks..
[31] S.J. Bradtke, Reinforcement learning applied to linear quadratic regulation, Advances in neural information processing systems..
[32] S.J. Bradtke, M.O. Duff, Reinforcement learning methods for continuous-time markov decision problems, in: G. Tesauro, D.S. Touretzky, T.K. Leen (Eds.), Advances in Neural Information Processing Systems 7, MIT Press, 1995, pp. 393–400.
[33] K. Doya, Reinforcement learning in continuous time and space, Neural Computation 12 (1) (2000) 219–245.
[34] C.J.C.H. Watkins, Learning from delayed rewards, Ph.D. thesis, King's College, Cambridge (1989)..
[35] L.C. Baird, III, Advantage updating..
[36] M.E. Harmon, L.C. Baird III, Multi-player residual advantage learning with general function approximation, Wright Laboratory..
[37] G.A. Rummery, M. Niranjan, On-line Q-learning Using Connectionist Systems (1994).
[38] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, arXiv preprint arXiv:1707.06347..
[39] E. Vasilaki, N. Frémaux, R. Urbanczik, W. Senn, W. Gerstner, Spike-based reinforcement learning in continuous state and action space: when policy gradient methods fail., PLoS Computational Biology 5 (12) (2009) e1000586..
[40] G. Chevalier, S. Vacher, J. Deniau, M. Desban, Disinhibition as a basic process in the expression of striatal functions. i. the striato-nigral influence on tecto-spinal/tecto-diencephalic neurons, Brain Research 334 (2) (1985) 215–226.
[41] J. Deniau, G. Chevalier, Disinhibition as a basic process in the expression of striatal functions. ii. the striato-nigral influence on thalamocortical cells of the ventromedial thalamic nucleus, Brain Research 334 (2) (1985) 227–233.
[42] R. Bogacz, K. Gurney, The basal ganglia and cortex implement optimal decision making between alternative actions, Neural Computation 19 (2) (2007) 442–477.
[43] C.W. Baum, V.V. Veeravalli, A sequential procedure for multihypothesis testing, IEEE Transactions on Information Theory 40 (6)..
[44] K. Samejima, K. Doya, Multiple Representations of Belief States and Action Values in Corticobasal Ganglia Loops, Annals of the New York Academy of Sciences 1104 (1) (2007) 213–228.
[45] R.P. Rao, Decision making under uncertainty: a neural model based on partially observable markov decision processes, Frontiers in Computational Neuroscience 4 (2010) 146.
[46] J.J. Nassi, E.M. Callaway, Parallel processing strategies of the primate visual system, Nature Reviews Neuroscience 10 (5) (2009) 360–372.
[47] S. Funahashi, C.J. Bruce, Mnemonic coding of visual space in the monkey's dorsolateral prefrontal cortex, Journal of Neurophysiology..
[48] J. Gottlieb, M.E. Goldberg, Activity of neurons in the lateral intraparietal area of the monkey during an antisaccade task, Nature Neuroscience 2 (10) (1999) 906–912.
[49] B.W. Brunton, M.M. Botvinick, C.D. Brody, Rats and humans can optimally accumulate evidence for decision-making, Science 340 (6128) (2013) 95–98.
[50] M. Wiering, J. Schmidhuber, HQ-Learning, Adaptive Behavior 6 (2) (1997) 219–246.
[51] N.D. Daw, J.P. O'Doherty, P. Dayan, B. Seymour, R.J. Dolan, Cortical substrates for exploratory decisions in humans, Nature 441 (7095) (2006) 876–879.
[52] L. Baird et al., Residual algorithms: Reinforcement learning with function approximation, in: Proceedings of the Twelfth International Conference on Machine Learning, 1995, pp. 30–37.
[53] G.A. Rummery, M. Niranjan, On-line Q-Learning using Connectionist Systems, University of Cambridge, Department of Engineering, 1994.
[54] W. Maass, On the computational power of winner-take-all, Neural Computation 12 (11) (2000) 2519–2535.
[55] J.O. Rombouts, P.R. Roelfsema, S.M. Bohte, Learning resets of neural working memory, in: Proceedings of the European Symposium on Neural Networks (ESANN 2014), 2014, pp. 111–116.
[56] D.J. Freedman, J.A. Assad, Experience-dependent representation of visual categories in parietal cortex, Nature..
[57] V. Mante, D. Sussillo, K.V. Shenoy, W.T. Newsome, Context-dependent computation by recurrent dynamics in prefrontal cortex, Nature 503 (7474) (2013) 78–84.
[58] C.K. Machens, R. Romo, C.D. Brody, Flexible control of mutual inhibition: a neural model of two-interval discrimination, Science 307 (5712) (2005) 1121–1124.
[59] A.K. Churchland, R. Kiani, M.N. Shadlen, Decision-making with multiple alternatives, Nature Neuroscience 11 (6) (2008) 693–702.
[60] A. Hernández, E. Salinas, R. García, R. Romo, Discrimination in the sense of flutter: new psychophysical measurements in monkeys., The Journal of Neuroscience: The Official Journal of the Society for Neuroscience 17 (16) (1997) 6391–6400..
[61] M. Karamanis, D. Zambrano, S. Bohté, Continuous-time spike-based reinforcement learning for working memory tasks, International Conference on Artificial Neural Networks, Springer (2018) 250–262.
[62] U. Frey, R.G. Morris, et al., Synaptic tagging and long-term potentiation, Nature 385 (6616) (1997) 533–536.
[63] R.L. Redondo, R.G. Morris, Making memories last: the synaptic tagging and capture hypothesis, Nature Reviews Neuroscience 12 (1) (2011) 17.
[64] M. Ito, K. Doya, Distinct neural representation in the dorsolateral, dorsomedial, and ventral parts of the striatum during fixed- and free-choice tasks, The Journal of Neuroscience: The Official Journal of the Society for Neuroscience 35 (8) (2015) 3499–3514.
[65] R. Bogacz, T. Larsen, Integration of reinforcement learning and optimal decision-making theories of the basal ganglia, Neural Computation 23 (4) (2011) 817–851.
[66] K. Doya, Modulators of decision making, Nature Neuroscience 11 (4) (2008) 410–416.
[67] R. Bogacz, E. Brown, J. Moehlis, P. Holmes, J.D. Cohen, The physics of optimal decision making: a formal analysis of models of performance in two-alternative forced-choice tasks, Psychological Review 113 (4) (2006) 700–765.
[68] M. Akrout, C. Wilson, P. Humphreys, T. Lillicrap, D.B. Tweed, Deep learning without weight transport, in: Advances in Neural Information Processing Systems, 2019, pp. 976–984..
[69] T.R. Stanford, S. Shankar, D.P. Massoglia, M.G. Costello, E. Salinas, Perceptual decision making in less than 30 milliseconds, Nature Neuroscience 13 (3) (2010) 379–385.
[70] A.A. Koulakov, S. Raghavachari, A. Kepecs, J.E. Lisman, Model for a robust neural integrator, Nature Neuroscience 5 (8) (2002) 775.
[71] Y. Gal, Z. Ghahramani, Dropout as a bayesian approximation: Representing model uncertainty in deep learning, in: International Conference on Machine Learning, 2016, pp. 1050–1059.
[72] N. Wycoff, P. Balaprakash, F. Xia, Neuromorphic acceleration for approximate bayesian inference on neural networks via permanent dropout, in: Proceedings of the International Conference on Neuromorphic Systems, 2019, pp. 1–4.
[73] L. Buesing, J. Bill, B. Nessler, W. Maass, Neural dynamics as sampling: a model for stochastic computation in recurrent networks of spiking neurons, PLoS Computational Biology 7 (11) (2011) e1002211.

[74] J.O. Rombouts, S.M. Bohte, J. Martinez-Trujillo, P.R. Roelfsema, A learning rule that explains how rewards teach attention, Visual Cognition 23 (1–2) (2015) 179–205.

Dr. Davide Zambrano (M) is a senior post-doc in the Laboratory of Intelligent Systems of the Ecole polytechnique fèdérale de Lausanne (EPFL). He received his M.Sc. degree (cum laude) in biomedical engineering and his Ph.D. degree in health technologies from the University of Pisa, Pisa, Italy, in 2008 and 2012, respectively. He worked as a Ph.D. and post-doc at The BioRobotics Institute of The Scuola Superiore Sant Anna, Pisa, Italy. From 2014 to 2018 he has worked as post-doc at the CWI Machine Learning group with Prof. Sander Bohte on spiking neural networks, biological plausible reinforcement learning models.

Prof. Pieter Roelfsema (M) is director of the Netherlands Institute for Neuroscience and he also heads the lab "Vision & Cognition" at this institute. Additionally, he is a part-time professor at the University of Amsterdam and also at the Free University Amsterdam. He investigates how neurons in different brain areas work together during visual cognition and he proposed the influential theory that the processing of visual stimuli occurs in different phases with different contributions of feedforward and feedback connections. Roelfsema has received many awards including the NWO VICI award and the EU ERC advanced grant.

Prof Dr Sander M. Bohté (M) is a senior researcher and PI in the CWI Machine Learning group, and also a part-time professor of Cognitive Computational Neuroscience at the University of Amsterdam and of Bio-Inspired Neural Networks at the Rijksuniversiteit Groningen, The Netherlands. He received his PhD in 2003 on the topic of "Spiking Neural Networks" and worked as a post-doc with Prof Dr Michael Mozer at the University of Colorado, Boulder, USA. Since 2016, he is part of the CWI Machine Learning group, where his research bridges the field of neuroscience and bio-inspired neural networks.