Biologically Plausible Reinforcement Learning of Deep Cognitive Processing

Alexandra R. van den Berg

Biologically Plausible Reinforcement Learning of Deep Cognitive Processing

Alexandra R. van den Berg

Biologically Plausible Reinforcement Learning of Deep Cognitive Processing

Alexandra R. van den Berg

Biologically Plausible Reinforcement Learning of Deep Cognitive Processing

CWI
Centrum Wiskunde & Informatica

NETHERLANDS
INSTITUTE
FOR NEUROSCIENCE
Master the mind

UNIVERSITEIT
VAN AMSTERDAM

NWO

HP Human Brain Project

This work was carried out at the Machine Learning group at the Centrum Wiskunde & Informatica (CWI), in collaboration with the Vision & Cognition group at the Netherlands Institute for Neuroscience (NIN).

The research in this thesis was financially supported by the Dutch Research Council (NWO, NWO-NWA grant NWA.1292.19.298), European Union's Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreement No. 945539 (Human Brain Project SGA3, Task 3.7), Horizon Europe (ERC advanced grant 101052963 "NUMEROUS"), "DBI2", a Gravitation program of the Dutch Ministry of Science, Education and Culture) and Agence Nationale de la Recherche (ANR) within Programme Investissements d'Avenir, Institut Hospitalo-Universitaire FOReSIGHT (ANR-18- 590 IAHU-0001). We also acknowledge the use of Fenix Infrastructure resources, which are partially funded from the European Union's Horizon 2020 research and innovation programme through the ICEI project under the grant agreement No. 800858.

Cover design by Flux 1.1 Pro.

Printed and bound by Ridderprint.

Biologically Plausible Reinforcement Learning of Deep Cognitive Processing

Academisch Proefschrift

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. P.P.C.C. Verbeek
ten overstaan van een door het College voor Promoties ingestelde
commissie, in het openbaar te verdedigen in de Agnietenkapel
op dinsdag 1 juli 2025, te 13.00 uur

door

Alexandra Roeline van den Berg

geboren te Amsterdam

Promotiecommisie

Promotores:	prof. dr. S.M. Bohte	Universiteit van Amsterdam
	prof. dr. P.R. Roelfsema	Universiteit van Amsterdam
Overige leden:	prof. dr. C.M.A. Pennartz	Universiteit van Amsterdam
	dr. H.S. Scholte	Universiteit van Amsterdam
	dr. I.I.A. Groen	Universiteit van Amsterdam
	prof. dr. M.A.J. van Gerven	Radboud University
	dr. D. George	Google Deepmind

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

Contents

Chapter 1
Introduction

Reinventing the wheel is an admirable, yet likely fruitless endeavour. Luckily for us, nature has – over the course of evolution – created effective solutions to a large number of problems that we can shamelessly draw inspiration from. One such example is flight. To develop airplanes, the Wright brothers studied the mechanism behind flying in birds, and concluded that artificial flight must rely on the gliding on air currents (Padfield & Lawrence, 2003). Other examples are plentiful. Many types of textiles find their basis in biology (Eadie & Ghosh, 2011), flying in bird-like formations improves the efficiency of airplanes (Ning et al., 2011), and the structural strength of the hexagon was rightfully acknowledged after observing it in honeycombs (Penick et al., 2022).

These examples demonstrate the power of bio-mimicry. It would only make sense to also draw inspiration from what is arguably one of the most complex and fascinating biological systems: the human brain. Yet, the brain remains shrouded in mystery. One of the primary facets contributing to its obscurity of course being that it is very much embedded in the body and rather vital for survival. As such, it unfortunately does not lend itself to being messed with if one has a good sense of (self-)preservation. Not to be defeated, researchers have developed various ways of investigating it despite these limitations. For instance, the invention of computed tomography (Schulz et al., 2021) and magnetic resonance imaging (Macovski, 2009) allowed people to non-invasively visualise the anatomical structure of human brains, and methods such as electroencephalography (Empson, 1986) and functional magnetic resonance imaging (Bandettini, 2012) made it possible to draw inferences about brain function. In recent years, it has even become possible with techniques such as deep brain stimulation (Schwalb & Hamani, 2008) or transcranial focused ultrasound (Fini & Tyler, 2017) to alter human brain activity without causing damage.

Animal research has also been pivotal in unravelling the mysteries of the brain. For example, early studies using the giant squid axon contributed immensely to our understanding of how electrical signals are transmitted through the nervous system (Hodgkin & Huxley, 1939). Moreover, as technology advances we can now gather information about the activity of not only one isolated nerve fibre, but also that of large populations of neurons across different regions of the brain – while animals are awake and are performing tasks (Jun et al., 2017). Furthermore, we can use specially designed pharmacological agents (Urban & Roth, 2015) and photopigments (Dugué et al., 2012) to temporarily increase or decrease the activity of specific neurons and study their effect on behaviour.

However, some important limitations remain using these methods. Namely, studies on humans are, naturally, subject to severe constraints since they should not cause any harm. Therefore, inferences can only be drawn indirectly based on the non-invasive measurement of relatively large groups of neurons (Kalin, 2021; Yen et al., 2023) and one cannot simply 'turn them off to see what happens'. On the other hand, more experimental methods are available for animal research. However, these studies are only useful to the extent that the structures or behaviours under study capture those seen in humans (Żakowski, 2020). This becomes especially challenging with higher cognitive functions and disease profiles such as depression (Hao et al., 2019), schizophrenia (Marcotte et al., 2001) or dementia (Neha et al., 2014), which are difficult to emulate in animal models. In addition, given the rapid progress in neuroscience in recent years, there is an increasing volume of experimental data being gathered, despite a relatively lack of theoretical frameworks to explain all this information (Levenstein et al., 2023). There is then only one natural conclusion: we must model the brain.

1.1 Why model the brain?

The previous discussion on bio-mimicry illustrated the utility of studying biological systems for inventing new technology. What could modelling contribute to our understanding of the brain?

In the wise words of Richard Feynman – and featured in virtually every university lecture, textbook, and PhD dissertation on the topic[1]:

> *"What I cannot create, I do not understand"*
>
> Feynman, 2019

.

As this quote illustrates, although one might believe to understand something, a deep comprehension can only be evidenced by recreating it. This might be familiar to anyone who has been humbled while building one of the more challenging IKEA sets after losing the instructions. Of course, the brain is vastly more complex than

[1] This quote being an excellent example of a wheel that need not reinventing.

a piece of furniture and replicating it entirely is therefore not only prohibitive, but also rather futile (for at that point, why not study a real brain?).

Instead, the true power of modelling lies in abstraction and simplification: what are the core components necessary for explaining a phenomenon and how do these components work together? For instance, when trying to understand memory, one might exclusively consider the interactions between various brain regions, while abstracting away cellular processes. Of course, the level of abstraction is highly dependent on the research question. Rather than looking at memory as a higher-level cognitive process, another researcher might prefer to investigate what molecular mechanisms underlie memory at the synaptic level. This brings us to our next famous modelling quote:

> *All models are wrong, but some are useful.*
>
> George E. P. Box, 1919-2013

What then makes a model useful? A useful model can be a tool to structure knowledge around. It can serve to integrate experimental findings and make sense of the world. It can be used to summarise and explain a phenomenon (Lens, 1987). Few people would realistically be interested in an exhaustive list of all experimental findings in a particular field. Rather, it would be vastly more useful to provide a more conceptual overview of what all these studies are trying to assess. Aside from theoretical understanding, there are also more practical applications of modelling. For instance, a model can guide scientific exploration by generating new hypotheses that can then be experimentally tested (Carley, 1999). This can form a feedback loop, where each of these new findings can be used to modify the model. This model can then generate new ideas that can once more be tested, therefore driving scientific progress over time. Finally, models can have various real-world practical applications, such as the aforementioned examples of bio-mimicry.

Similarly, models of the brain have a lot of practical potential. Not only is it fascinating to know more on a fundamental level about what makes us humans tick, a better understanding of the brain could for instance be tremendously helpful in healthcare. By discovering more about brain processes, we can gain new insights about what is going wrong in neurological diseases and use this information to develop new cures (Teufel & Fletcher, 2016). Additionally, the advent of brain-computer interfaces opens up possibilities for restoring brain functions such as sight (Chen et al., 2020) or audition (Shannon, 2012), and has even enabled people with spinal cord injuries to walk (Lorach et al., 2023).

1.2 How to model the brain: artificial neural networks

Provided the previous section convincingly demonstrated that computational neuroscience is a worthwhile pursuit, the next question arises: how to model the brain?

While there is a large variety of modelling approaches, ranging from conceptual models describing how brain processes might operate in abstract terms (e.g. (Baddeley, 1992; Wagemans et al., 2012)), to statistical models that aim to parameterise behaviour (e.g. (Pedersen et al., 2017; Rescorla & Wagner, 1972)), and even those that represent brains of specific individuals for healthcare purposes (Wang et al., 2024), we will here focus on a different subclass of models: artificial neural networks (ANNs).

ANNs were developed by using inspiration from neuroscience to build an artificial system (Hassabis et al., 2017). The main inspiration regards how brains are structured, with layers of neurons that are connected by synapses through which they communicate information. Some of the more simple and powerful ANNs have only three layers: an input layer, a hidden layer and an output layer (Kruse et al., 2022). Information from the environment (e.g. the image of an apple, see Figure 1.1) is first presented to the input layer. Units – or neurons – in this layer process this input by multiplying it with a set of synaptic weights (Kruse et al., 2022). For instance, some units may not care much for apple-related information (e.g. how red and round they are, whether they are packaged with friends, and so forth), resulting in very low activity in these neurons. Other neurons, in contrast, could be very interested in apples and become highly active. All these activations then pass through a so-called 'activation function', which transforms these weighted activations in a non-linear way, essentially enabling these networks to develop more powerful and complex representations (Kruse et al., 2022; Sharma et al., 2020). Subsequently, these activations are projected towards the hidden layer, which now performs these computations based on the end-product of the input layer. Consequently, information can be integrated (LeCun et al., 2015). For example, if 'apple' input neurons are activated at the same time as units coding for 'orange' and 'banana', a hidden unit encoding for the concept 'fruit' could become active. Finally, the resulting hidden unit activations traverse through the output layer, which is where (after the same initial processing as in the previous layers), decisions can be made. Usually, the unit that is overall the most highly active forms the output of the network. In the case of this example, if hidden units for the concepts 'fruit' and 'tree' were represented, a potential output of the network could be 'orchard'.

The above process describes the forward information flow through an ANN. However, the synaptic weights that determine each neuron's response to input is not predetermined: it is learned (LeCun et al., 2015). Typically, ANNs learn

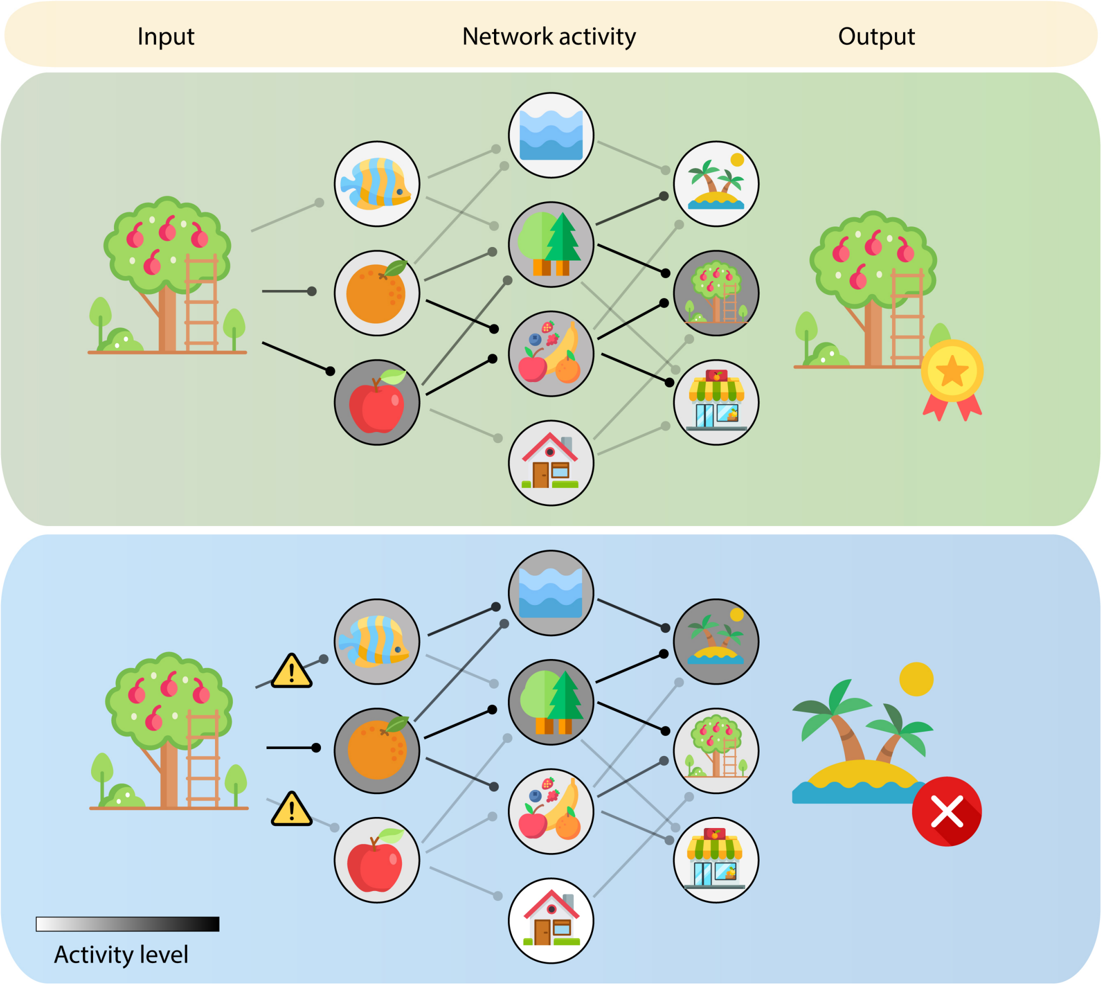

Figure 1.1: Feedforward processing in ANNs. In the green panel, a picture of an orchard is presented to a network. Units in the input layer encoding for apples are highly active, ones for oranges are moderately active and those for fish are inactive. The orange and apple input neurons activate hidden units for the concepts fruit and trees, while those for houses and the sea remain inactive. The tree and fruit hidden neurons in turn activate the output units for grocery store and orchard, but not island. The orchard output unit becomes the most active unit since it receives both activation from the tree and fruit units, compared to grocery store that only receives activation from the active fruit unit. Therefore, the network correctly selects orchard. In the blue panel, a different network chooses incorrectly after it observes the same image. Rather than the apple input unit being highly active upon seeing the image of an orchard, the fish input neuron is active together with the orange unit. This results in the hidden units for sea and trees becoming highly active, whereas the fruit unit is less active due to the absence of input from the apple unit. This leads to the island output unit incorrectly becoming the most active neuron in the final layer.

by receiving a specific target label that describes for each input what the correct response should have been. For instance, if a picture of an orchard was fed to the network and the network gave ‘island’ as output, this discrepancy triggers learning and the network will be altered in a way that makes it more likely to respond with ‘orchard’ when seeing similar images in the future (see Figure 1.1).

Although the hitherto mentioned ANN structure is relatively simple, more hidden layers could be added to make deep neural networks that are capable of highly complex processing (LeCun et al., 2015). A large variety of different model types have been developed in recent years and have achieved great success, even surpassing human-level performance on several games, as well as finding various applications in translation and image generation (Chang et al., 2024; Hassabis et al., 2017; OpenAI et al., 2024; Thirunavukarasu et al., 2023).

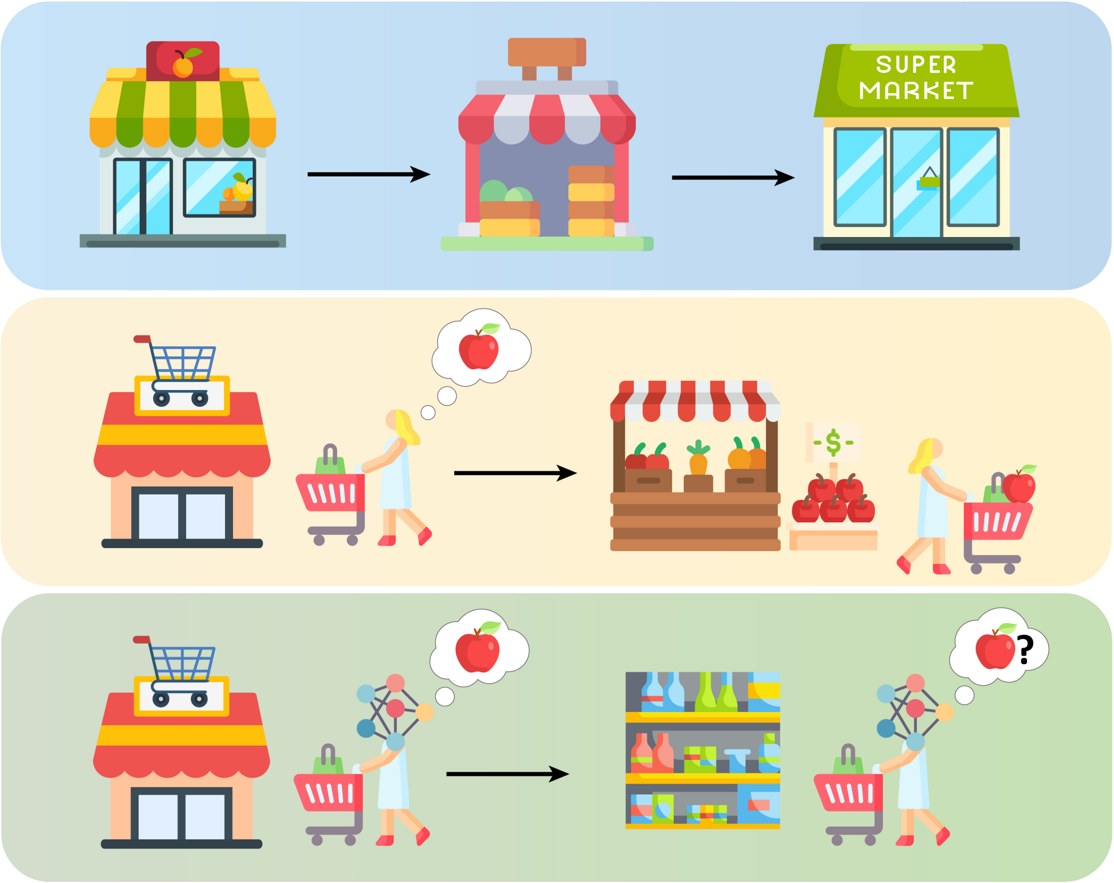

Figure 1.2: We can learn-to-learn over repeated experiences. For example, when visiting many different grocery stores (blue), we can learn their general lay-out. This allows us to quickly locate relevant aisles, even when we have never entered a particular supermarket before (yellow). This is since we can infer that apples belong to the fresh produce section, which is typically found near the entrance. In contrast, ANNs forget their past experiences (green). As a result, they would have to explore the supermarket at random in the hope of eventually stumbling upon apples.

1.3 How to make artificial neural networks more brain-like?

Although ANNs are very useful, there is one major caveat: they do not resemble the brain as much as their name would suggest. A well-known example of how ANNs and biological networks operate differently despite seemingly relying on similar mechanics for visual processing (Alsallakh et al., 2017; LeCun et al., 2015; Lee et al., 2009) concerns adversarial images. These images illustrate a failure of ANNs to recognise image classes they have been trained upon (e.g. dogs) due to added noise almost imperceptible to humans (Dujmović et al., 2020; Szegedy et al., 2013). Another very important way in which they diverge is in how they learn from experience. Imagine, for example, that you enter a grocery store you have never been in before in the search for apples. This may seem a rather trivial task since you know apples belong in the fresh produce section, which is generally located near the entrance (see Figure 1.2). However, an ANN may not find this equally trivial. This is because ANNs struggle with learning from experience and instead forget past learning entirely when confronted with new problems (Carpenter & Grossberg, 1987; French, 1999), essentially making every grocery store be the first one they have ever entered[2]. Then, grocery shopping may prove a very daunting endeavour indeed: with the large number of locations, together with the many products of different shapes, colours and sizes, such a model would have to aimlessly stumble around the store until – at last – apples are found. This phenomenon, and how to model it, is a central topic in the second chapter of this thesis.

As a consequence of these differences between artificial and biological systems, caution should be used while drawing inferences from them on the brain[3]. That brings us to the next question: how then can we make ANNs resemble the brain better?

The answer to this question is highly contingent on the topic of interest. As aforementioned, a model ought to be simple enough to be useful, while also being complex enough to be theoretically interesting. Biological realism can then be implemented in a variety of manners. One type of models focuses on network architecture by implementing several 'modules' that represent brain areas, and study how these modules interact to produce behaviours (O'Reilly & Frank, 2006). Spiking neural networks emulate more biologically realistic neural firing behaviour by incorporating factors such as membrane potentials in synaptic transmission (Paugam-Moisy & Bohte, 2012; Tavanaei et al., 2019). Another family of models reintroduces more complexity at the level of the synapse, by for instance modelling dendritic trees (Wu et al., 2018; Zheng et al., 2024), while different models consider also other cell types than neurons, such as astrocytes and other glial cells that

[2] Modelling dementia in ANNs turns out to be significantly easier than in animal models.

[3] Which is of course the goal here.

support neurons (Alvarellos-González et al., 2012; Porto-Pazos et al., 2011).

In this thesis, the focus is on how we can develop ANNs that learn information in a more biologically plausible manner. Namely, the algorithm (or 'learning rule') used to train ANNs called 'error-backpropagation' (Rojas, 1996) has proven to be very powerful, yet is problematic from a biological perspective (Whittington & Bogacz, 2019). One of the primary issues regards locality. The backpropagation algorithm assesses for each unit in all layers of the network what their exact contribution was to a particular network output (e.g. the label 'bee' upon seeing an image of a bee). If this network output was correct, it strengthens the connections that helped produce this response and otherwise weakens them (see Figure 1.3). However, in order to know precisely how each unit contributed to an output and thus how much it should change, it needs to consider how each unit influences the other units in the network. This is viable for neurons that share a synaptic connection, but becomes unrealistic for deeper networks. This is otherwise referred to as the credit assignment problem (Whittington & Bogacz, 2019). Credit assignment becomes even more tricky when considering learning over time (Lillicrap & Santoro, 2019), because then each unit would not only account for their influence on other units, but also how their responses have shaped network activity over time. This is referred to as the temporal credit assignment problem (Lillicrap & Santoro, 2019). Instead, what is considered 'biologically plausible' is that these network changes are exclusively based on information that is available at the level of the synapse, i.e. 'local' (Whittington & Bogacz, 2019). In other words, if a specific neuron does not have access to information about other neurons (for instance a neuron in the input layer that is not connected to one of the deeper hidden layers), this should not affect its plasticity (see Figure 1.3).

Another distinction between how ANNs are typically trained compared to humans and animals, is the type of feedback they receive from their environment. ANNs are often trained in a 'supervised' manner (see Figure 1.4), which involves informing the network not only that it is incorrect when it makes a mistake, in addition to detailing what the correct response should have been (e.g. 'flower' when a network provides 'crocodile' as output) (Cunningham et al., 2008). Conversely, a large amount of biological learning tends to not be as explicit, and often proceeds by trial-and-error (Roelfsema & Holtmaat, 2018). This can be seen, for instance, in animal training: if a dog performs a desired action (e.g. sitting down upon the word 'sit'), it could receive a food reward, which incentivises it to perform this action once more when the command is given in the future. However, if it performs the wrong action (e.g. barking), it only learns through the absence of a reward that it was misguided, but does not receive information about what the correct action ought to have been. This type of learning is also referred to as reinforcement learning (Sutton & Barto, 2018).

Therefore, the main research question central to this thesis is the following: how can we train neural networks in a way that mimics learning in the brain? For this, we consider 'biologically plausible' to mean learning that is local and based

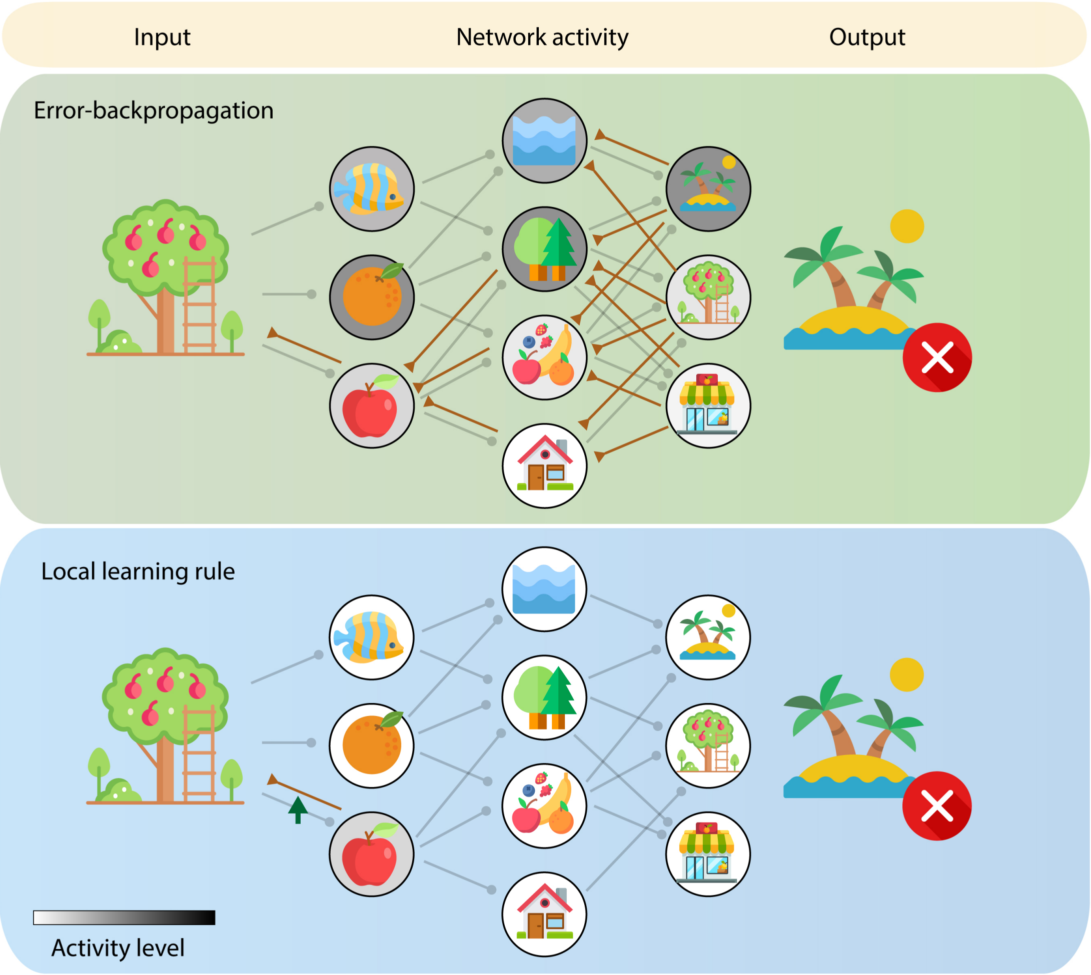

Figure 1.3: Learning in ANNs. The green panel depicts the mechanism behind error-backpropagation. This network erroneously gave ‘island’ as output, rather than the correct response of ‘orchard’, partially due to a lack of activity of the apple input unit. To make sure this unit becomes more active when observing orchard images in the future, it has to strengthen the connection to the apple unit. To do this, error-backpropagation uses non-local information about activity levels of hidden units and output units. For instance, it would use the orange feedback connections to consider how much the apple unit activated the tree unit and how much that tree unit activated the incorrect island unit, or the orchard unit that should have been chosen. Additionally, it would calculate how much the apple unit activated the fruit unit and how much the fruit unit in turn activated the island unit and orchard unit. This is information that the synapse to the apple unit does not have direct access to and should therefore not be used when developing a biologically plausible learning rule. The blue panel shows a local learning rule that only relies on information at the synapse to the apple unit: namely, the activity of the relevant apple unit, the input to this unit (the orchard image), and the strength of the connection between these. No information is used that impacted the incorrect decision by the network, but that this synapse does not have access to, such as the activity level of the grocery store output unit or that of the sea hidden unit.

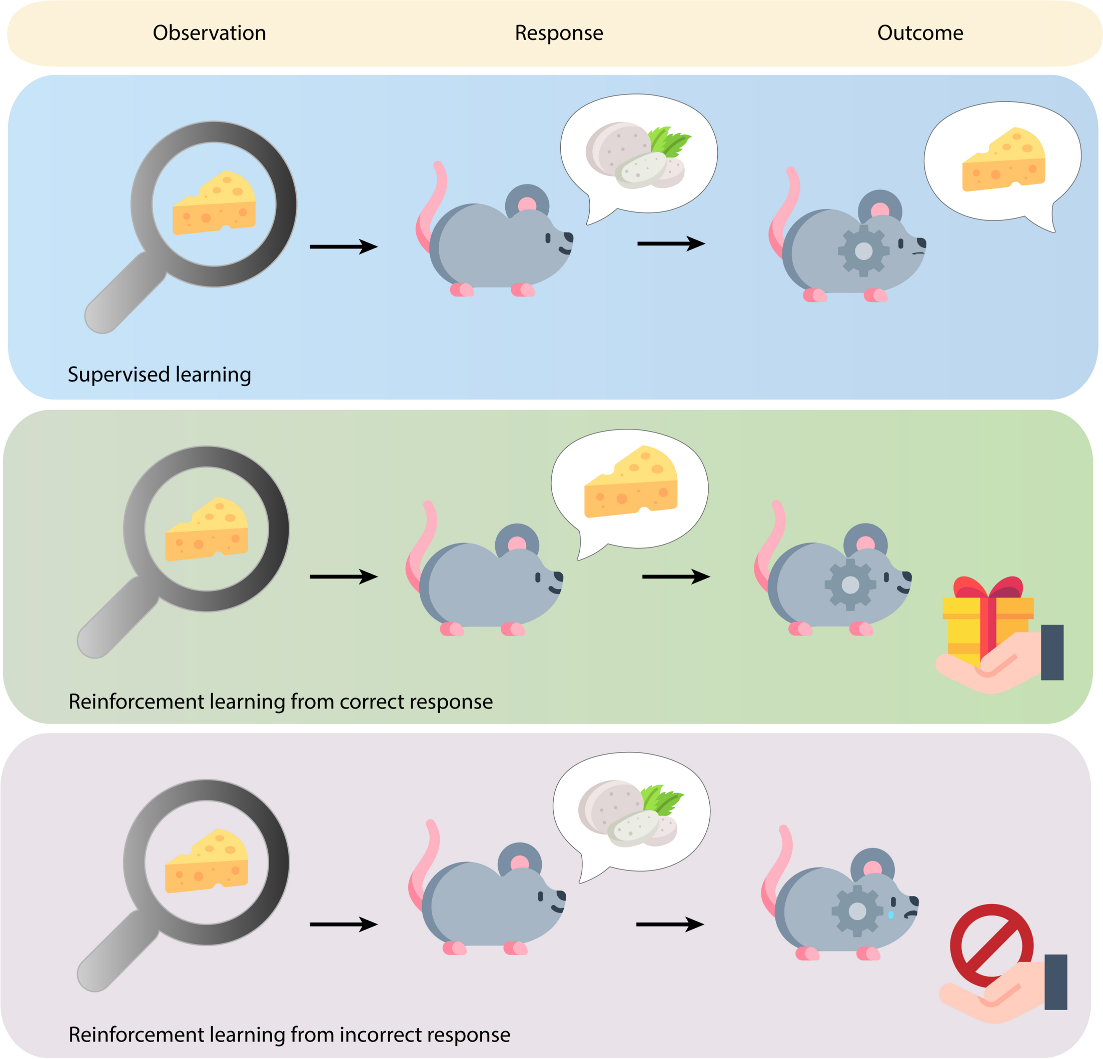
Observation
Response
Outcome
Supervised learning
Reinforcement learning from correct response
Reinforcement learning from incorrect response

Figure 1.4: Different types of learning. In supervised learning (blue panel), a mouse observes a type of cheese and is asked what cheese it is. If it incorrectly classifies Gouda cheese as mozzarella, it receives information about what its response should have been. In reinforcement learning, the mouse is only given a reward upon correct responses (green panel) and is not told what the correct response should have been if it was mistaken (purple panel).

on trial-and-error.

1.4 Chapter overview

Here, I will outline the different chapters in this thesis.

Chapter II: Can we develop a biologically plausible network that can learn-to-learn?

The second chapter introduces a fundamental difference between biological and artificial systems: how we learn from experience, or how we 'learn-to-learn'. I will discuss how this is believed to occur in the human brain and how a biologically plausible implementation of this mechanism is currently lacking. I introduce a new model called RECOLLECT that builds upon previous biologically-inspired learning rules, and extend it to simplified models of memory so that we can investigate carefully how these models learn over time. We demonstrate that RECOLLECT can learn to flexibly memorise and forget information depending on task demands, and can use this capacity to learn-to-learn in ways that resemble animal physiology and behaviour.

This paper has been published in *PLoS ONE 19*(12): e0316453.

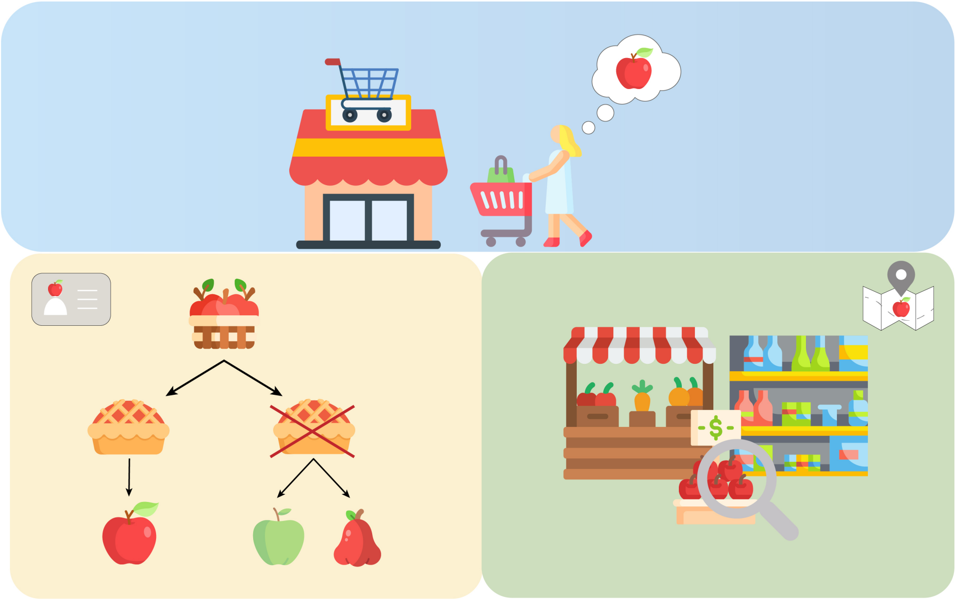

Figure 1.5: Humans can flexibly use working memory to obtain their goals. For instance, when buying an apple at the grocery store, one might need to recall which type of apples would be tasty for an apple pie, but also remember where these apples can be found within the store. The more complex the goal and the sensory environment are, the more helpful a deeper memory architecture might be to solve it.

Chapter III: Can we develop a biologically plausible deep gated memory network for working memory?

In the third chapter, I generalise the RECOLLECT model to deeper architectures to create the Stackollect model. As a deeper model, Stackollect can learn more complex tasks with richer sensory input (see Figure 1.5). Moreover, since Stackollect has multiple layers of memory units, this opens up the possibility to see how task-relevant features are encoded in memory across these different layers. To this end, I developed a novel working memory paradigm and show that Stackollect can successfully learn to memorise information over a delay to solve this task.

Chapter IV: Can we develop a curriculum to train networks on complex datasets with biologically plausible learning rules?

In the fourth chapter, I emphasise the importance of not only developing models and learning rules that resemble the brain, but also of carefully structuring the way we train them. Drawing inspiration from how we teach children and animals, I develop a curriculum that allows the biologically inspired BrainProp model (Pozzi et al., 2020) to learn on large-scale, complex datasets that hitherto proved too challenging to solve (see Figure 1.6).

Chapter V: Discussion
In the fifth and final chapter, I discuss and position the results from these three chapters in a broader context, outlining links to related models, current limitations and future opportunities.

Overall, I contribute to bridging the gap between biological and artificial intelligence by developing new biologically plausible models of learning-to-learn and working memory, as well as providing a foundation for how we can structure and improve their learning process in more biologically relevant ways.

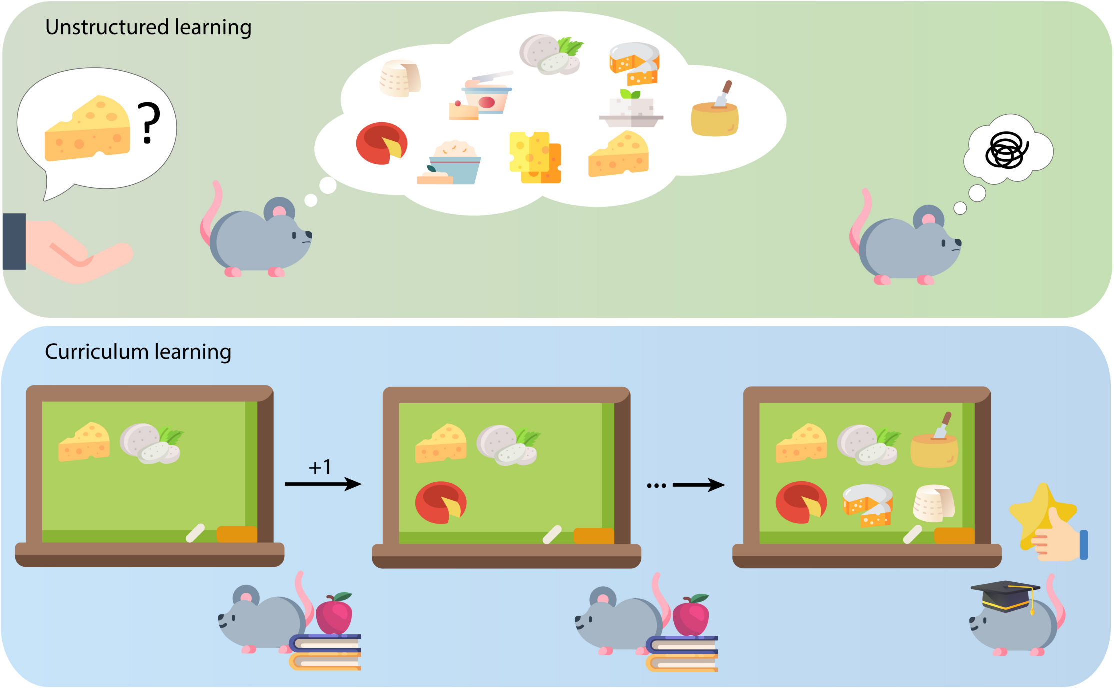

Figure 1.6: When learning a large number of classes, it can be very helpful to structure training with a curriculum. When this is not done, it can become very difficult to train a mouse with reinforcement learning to classify different types of cheeses (green panel), since it cannot be told what the correct cheese should have been if it was wrong. It would have to try out many types of cheese before it learns what the correct cheese should have been. A curriculum can accelerate this process (blue panel). By first teaching a mouse to distinguish between two types of cheese (Gouda and mozzarella), the problem is simplified. Once the mouse successfully distinguishes the two cheeses, another cheese can be introduced. This process is repeated until all cheeses have been learned by the mouse and it can successfully classify all the different cheeses in the dataset.

Chapter 2

Biologically plausible gated recurrent neural networks for working memory and learning-to-learn

Alexandra R. van den Berg[1,2], Pieter R. Roelfsema[2,3,4,5], Sander M. Bohté[1,6]

[1]*Machine Learning Group, Centrum Wiskunde & Informatica, Amsterdam, the Netherlands*

[2]*Department of Vision & Cognition, Netherlands Institute for Neuroscience, Amsterdam, the Netherlands*

[3]*Department of Integrative Neurophysiology, Center for Neurogenomics and Cognitive Research, Vrije Universiteit Amsterdam, Amsterdam, the Netherlands*

[4]*Department of Neurosurgery, Academic Medical Center, Amsterdam, the Netherlands*

[5]*Laboratory of Visual Brain Therapy, Sorbonne Université, Institut National de la Santé et de la Recherche Médicale, Centre National de la Recherche Scientifique, Institut de la Vision, Paris, France*

[6]*Swammerdam Institute of Life Sciences, University of Amsterdam, Amsterdam, the Netherlands*

As published in PLoS ONE 19(12): e0316453.

Abstract The acquisition of knowledge and skills does not occur in isolation but learning experiences amalgamate within and across domains. The process through which learning can accelerate over time is referred to as learning-to-learn or meta-learning. While meta-learning can be implemented in recurrent neural networks, these networks tend to be trained with architectures that are not easily interpretable or mappable to the brain and with learning rules that are biologically implausible. Specifically, these rules have often employed backpropagation-through-time, which relies on information that is unavailable at synapses that are undergoing plasticity in the brain. Previous studies that exclusively used local information for their weight updates had a limited capacity to integrate information over long timespans and could not easily learn-to-learn. Here, we propose a novel gated memory network named RECOLLECT, which can flexibly retain or forget information by means of a single memory gate and is trained with a biologically plausible trial-and-error-learning that requires only local information. We demonstrate that RECOLLECT successfully learns to represent task-relevant information over increasingly long memory delays in a pro-/anti-saccade task, and that it learns to flush its memory at the end of a trial. Moreover, we show that RECOLLECT can learn-to-learn an effective policy on a reversal bandit task. Finally, we show that the solutions acquired by RECOLLECT resemble how animals learn similar tasks.

2.1 Introduction

A hallmark of human intelligence is the capacity to accumulate knowledge across learning experiences. This capacity not only accelerates learning within one domain, but can also facilitate learning in related domains, a phenomenon referred to as learning-to-learn (Harlow, 1949; Thrun & Pratt, 1998). Standard neural network models lack this ability and quickly and catastrophically forget previously acquired knowledge when they are trained on a new task (Carpenter & Grossberg, 1987; French, 1999). This is particularly problematic in the case of reversal learning (Izquierdo et al., 2017), where stimuli are initially associated with a certain reward probability, e.g. stimulus A with a 75% chance of reward and stimulus B with a 25% chance of reward. When the stimulus-reward associations are reversed, i.e. stimulus A is now rewarded with 25% probability and stimulus B with 75% probability, the network has to fully change its weight structure to adjust to the new reward probabilities. To overcome this limitation, researchers have developed meta-learning models that acquire a set of weights over the course of several similar tasks that facilitate generalisation to novel tasks if they bear similarities to previously learned tasks.

Meta-learning can be achieved using various approaches (Huisman et al., 2021; Sutton, 2022; Wang, 2021). An approach that is plausible from a biological perspective uses recurrent neural networks that are trained with reinforcement learning (Duan et al., 2016; Wang et al., 2018). These networks are trained on a distribution of tasks and learn to rely on information about previous stimuli, actions and rewards to represent the appropriate task context. Subsequently, they can carry out new tasks even if the weights of the network are fixed, provided meta-learning was successful. In this framework, the network learns to accumulate information about the new task in its working memory by observing the reward structure. A previous study by Wang et al. (2018) suggested that learning-to-learn could rely on interactions between the prefrontal cortex, the basal ganglia and the thalamus for the build-up of working memory representations that support learning-to-learn. Task switching can happen within one or a few trials by adapting the activity pattern in working memory as opposed to going through the elaborate process of retraining the network connectivity.

Even though the behaviour of these meta-learning models is similar to that of animals, the architectures and learning rules have limited biological plausibility for at least two reasons. Firstly, some of the previous studies on meta-learning relied on complex units, such as the long short-term memory (LSTM) unit (Hochreiter & Schmidhuber, 1997). The LSTM unit has three multiplicative gates that control its activity, which is unnecessary for some tasks (Dey & Salem, 2017; Ravanelli et al., 2018), can be difficult to interpret and may not be found in biological neurons. Simplifications of LSTM units have been proposed, such as the gated recurrent unit (GRU), which has two gates (Cho et al., 2014), and more recently, the light-gated recurrent unit (Light-GRU) with a single gate (Ravanelli et al., 2018). Models with these simpler units have yielded good or even superior performance on some tasks compared to architectures with LSTM units (Cho et al., 2014; Ravanelli et al., 2018).

Secondly, previous models were trained with non-biological learning rules, such as backpropagation-through-time (BPTT). Updates in BPTT rely on information that is not available locally at synapses (i.e. it is non-local in time; Lillicrap & Santoro, 2019). An example of an algorithm that is biologically plausible is AuGMEnT, because synapses trained with this learning rule have access to the necessary information (Rombouts et al., 2015). AUGMEnT includes units with persistent activity for working memory and uses synaptic traces, local signals that are stored within synapses to influence plasticity (information about AuGMEnT can be found in Methods). These traces determine which synapses should be strengthened and which ones should be weakened and help to solve a spatial and a temporal credit assignment problem. The spatial credit assignment problem is related to identifying the synapses in the network that are responsible for the outcome of an action. AuGMEnT solves the spatial credit assignment problem with an attentional feedback signal originating from the selected action that highlights the synapses that are responsible for it and are therefore eligible for plasticity. The temporal credit assignment problem is to identify actions that are associated with rewards that only come after a delay and that may be contingent on later actions. AuGMEnT solves the temporal credit-assignment problem by computing a reward-prediction error and by including memory units, which can maintain information about previous sensory inputs. However, AuGMEnT lacks mechanisms for forgetting and the memory therefore needs to be reset after each trial. The inability to integrate information across trials hinders its ability to learn-to-learn. A related biologically inspired learning rule is e-prop (Bellec et al., 2020), which also approximates BPTT by using synaptic traces.

In this study, we propose RECOLLECT, a learning rule based on Light-GRUs that modifies synapses based exclusively on information that is both local in space and time, making it biologically plausible. RECOLLECT adapts the synaptic tags and traces from AuGMEnT (Rombouts et al., 2015) to implement a learning rule that closely approximates BPTT but that can also forget information that is no longer relevant and solves the spatial credit-assignment signal for deeper networks. We show that RECOLLECT can flexibly use its working memory to perform a pro-/anti-saccade task and that it learns-to-learn on a reversal bandit task. Finally, we illustrate similarities between the training of networks with RECOLLECT and how animals acquire these tasks.

2.2 Results

2.2.1 Architecture

Feedforward processing

Our aim is to develop a biologically plausible architecture that can learn to memorise and forget. Specifically, we strived for a brain-like architecture and a learning rule in which all the information necessary for a weight change is available locally, at the synapse.

The novel model is called "REinforCement learning of wOrking memory with bioLogically pLausible rECurrent uniTs" - RECOLLECT (Fig 2.1). RECOLLECT draws inspiration from two models: the light-gated recurrent unit (Light-GRU; Ravanelli et al., 2018) and AuGMEnT (Rombouts et al., 2015; see 'AuGMEnT model' in Methods). The network's goal is to learn action-values (known as Q-values; Sutton & Barto, 2018), which correspond to the amount of reward that is predicted for a particular action when executed in a particular state of the world. If the outcome deviates from the reward-prediction, a neuromodulatory signal that encodes the global reward-prediction error (RPE) gates synaptic plasticity to change the Q-value, in accordance with experimental findings (Dayan & Balleine, 2002; Montague et al., 2004; Morris et al., 2006; Schultz, 2007). RECOLLECT uses a variant of Light-GRU units to learn tasks that require memorisation and forgetting, so that the network can integrate feedback from the environment across trials and determine if it is time to switch to another stimulus-response mapping.

The Light-GRU (Ravanelli et al., 2018) is a recurrent network that combines incoming sensory information with a memory of the state of the environment of the previous timestep. The maintenance of information in working memory is regulated by a learnable 'gate' that determines the influence of the memory and new sensory inputs. This ability enables the network to maintain memories when needed, but also to erase them and focus on new input when memories lose relevance or when the environment changes. Light-GRU units might correspond to a circuit with several neurons in the brain, for example, the neurons of the so-called direct and indirect pathways, which form a loop from cortex to basal ganglia, thalamus and then back to cortex (see Discussion).

RECOLLECT consists of an input layer, a memory layer with GRUs and an output layer. As in Light-GRU (Ravanelli et al., 2018), the memory layer contains three types of units: candidate memory cells (C_j), gating units (k_j) and memory cells (M_j), which might be part of the same cortical column or part of a loop involving the cortex, basal ganglia and thalamus. Incoming sensory information ($x_i(t)$) is processed by the candidate memory cells and available to update the activity of the memory cell:

$$C_j(t) = \sigma(\sum_i W_{ij}^C x_i(t) + b_j^C). \tag{2.1}$$

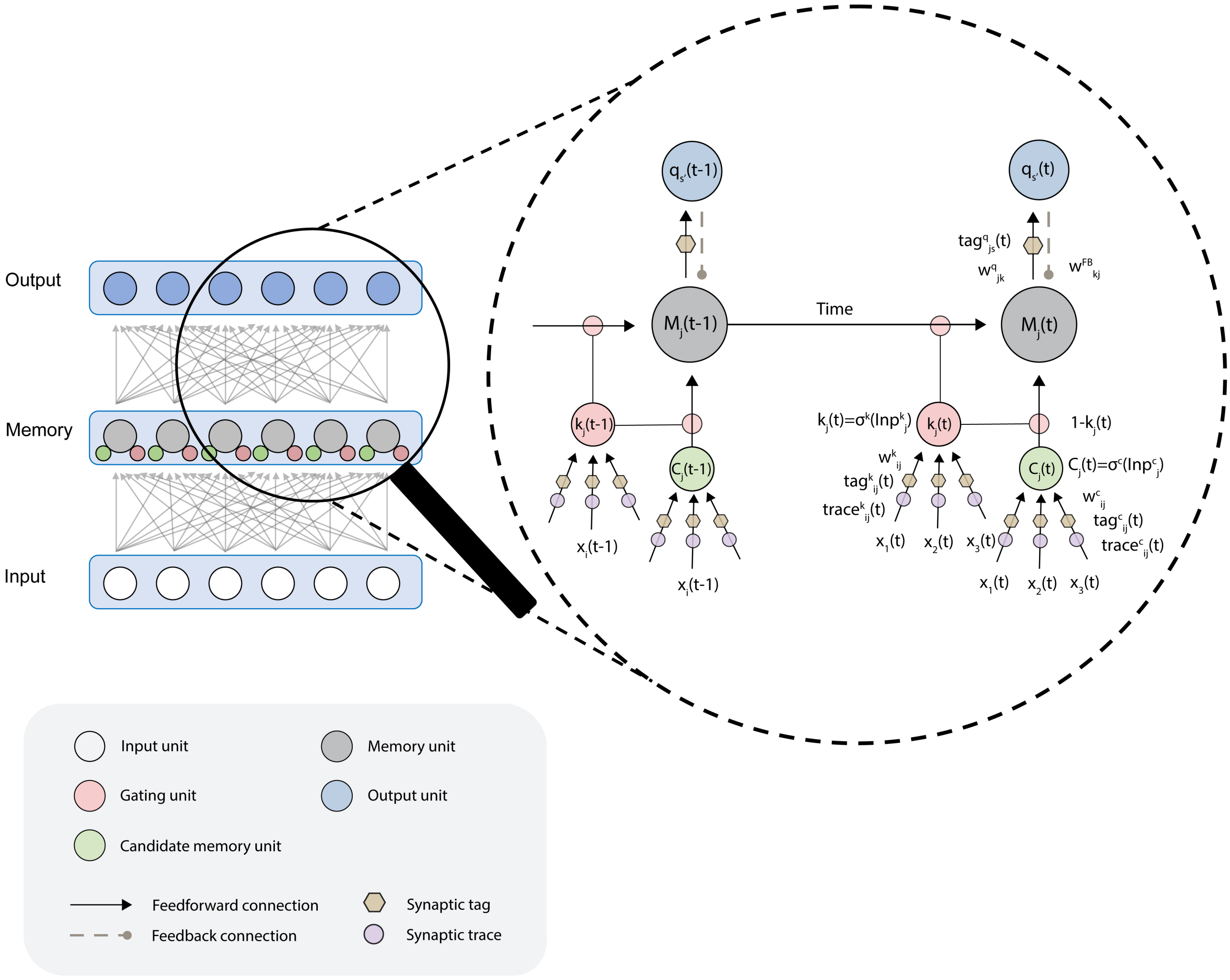

Figure 2.1: RECOLLECT Architecture. Gating units (red circles, k_j) balance between memory and updating by novel information from candidate memory cells (green circles, C_j). Memory units (gray circles, M_j) activate output units that estimate the Q-values of actions (blue circles). Synaptic tags (yellow hexagons) and traces (purple circles) store information that is necessary for the synaptic updates. Traces measure the influence of a connection on the activity of the memory unit and tags the influence of a connection on the selected Q-value unit (see Fig 2.2A for a more detailed explanation). Dashed grey lines, feedback connections from output units to memory units (W^{FB}_{kj}).

Here, C_j represents the activity of the candidate memory units, W^C_{ij} denotes the synaptic weights between sensory unit i and candidate memory unit j, b^C_j the bias and $\sigma(\cdot)$ is the sigmoidal activation function used to constrain the output between 0 and 1 (see equation M1 in the Methods):

The gating units k_j determine the degree to which the memories are maintained or overwritten by new sensory input. The activity of the gating units k_j depends on the input through weights W^k_{ij}:

$$k_j(t) = \sigma(\sum_i W^k_{ij} x_i(t) + b^k_j). \tag{2.2}$$

The gating units determine the updating of the activity of memory units M_j as follows:

$$M_j(t) = k_j(t) \odot M_j(t-1) + (1 - k_j(t)) \odot C_j(t), \tag{2.3}$$

where $\odot$ refers to element-wise multiplication. If gating units are active, the candidate memory cells do not have much influence on the memory unit and the previous memory $M_j(t-1)$ is retained. In contrast, if the gating units are only weakly active, the memory units make a large step in the direction of the activity level C_j of the candidate memory cells. We therefore refer to this gate as a memory gate. The process by which RECOLLECT uses memory gates to balance memorisation and forgetting is depicted in Fig 2.2B.

One important difference between the Light-GRU units in RECOLLECT and those originally formulated by (Ravanelli et al., 2018) is the exclusion of recurrent weights that allow previous memory states to affect the updating of the gate and candidate memory (equations 1 and 2). As we discuss in the next section ('learning rule'), this allowed us to derive an exact alternative for BPTT, rather than an approximation thereof. Another advantage was the additional simplicity provided to the model. Other differences include a different activation function (sigmoid, rather than rectified linear units) and the exclusion of batch normalisation.

The activity of the memory units is propagated to the output units:

$$q_k(t) = \sigma(\sum_j W^q_{jk} M_j(t) + b^q_k). \tag{2.4}$$

The output units estimate the Q-value q_k, the expected (discounted) reward of each action k that can be taken by the network. Once these values have been computed, an epsilon-greedy strategy selects the winning action s, where the action with the highest Q-value is chosen with probability 1-ϵ, and a random action is selected with probability ϵ.

Finally, there are feedback connections extending (W^{FB}_{sj}) from the output units back to the memory units. As we will discuss in the next section, these feedback connections influence plasticity of connections from input units to gating- and candidate memory units.

Learning rule

Reinforcement learning

RECOLLECT defines a learning rule for the Light-GRUs that is based on synaptic tags and traces and relies exclusively on information local to the synapse. This rule is equivalent to BPTT when the model does not use recurrent connections (as in the model described in the previous section). In this section, we explain the equations that determine learning in RECOLLECT.

As is common in models of reinforcement learning that use Q-learning, RECOLLECT selects an action s, and it may or may not receive a reward. If this reward differs from the expected reward based on the Q-value of the chosen action, this discrepancy gives rise to a reward prediction error (RPE) δ:

$$\delta(t) = r(t) + \gamma q_s(t) - q_a(t-1). \tag{2.5}$$

The SARSA temporal difference learning rule compares the predicted outcome of the previous action $q_a(t-1)$ to the sum of the observed reward $r(t)$ and the discounted Q-value of the winning unit $q_s(t)$. The reward discount factor γ, which ranges between 0 and 1, controls the discounting of future rewards, which are considered less valuable than immediate rewards. A negative RPE indicates that the outcome was worse than anticipated, whereas a positive RPE signals that a higher reward was received than was estimated at the previous time step. The RPE is presented to the network in the form of a global neuromodulator, hence it is a signal that is accessible for all synapses in the network.

Tags and traces

When synapses are exposed to the neuromodulator that reflects the RPE, plasticity can occur. As in AuGMEnT (Rombouts et al., 2015), plasticity is regulated using tags and traces. It is important to distinguish between the role of these components. Tags are formed on all synapses that contributed towards action selection and they register how much a synapse contributed to the selected action (Houk et al., 1995; Sutton & Barto, 2018). Tags also form on the synapses from the input layer to the memory layer, based on feedback connections from the selected action to the memory layer. After their formation, the tags interact with the global neuromodulator that provides information about the RPE. Consequently, only those synapses that were tagged will become plastic. Because the plasticity rule for feedback connections from the output units to the memory units is the same as that of feedforward connections from the memory units to the output units, these connections become proportional in strength as learning progresses.

Unlike tags, the synaptic traces are only maintained on connections from input units to the candidate memory units and gating units. The synaptic traces measure the influence of a synapse on the activity level of a memory unit, taking the history of memory activity into account. Specifically, if input unit i contributed to the activation of a memory unit j, then the $trace_{ij}$ keeps track of how much of this input is still visible in the memory activity, even if this input occurred in the past.

The tags and traces ensure that all the information that is required for network updates is locally available (see Fig 2.2A for a schematic illustration of the learning rule). The following equations define the updates for the tags, traces and weights for each of the units in RECOLLECT.

For the output units, the tags are formed in the presence of both presynaptic activity (M_j) and postsynaptic activity after an action s is selected. The Tag^q_{jk} only increases if the output unit k is selected, i.e. if $k = s$, in which case the presynaptic activity M_j of memory unit j is added to the tag:

$$Tag^q_{js}(t) = \lambda\gamma Tag^q_{js}(t-1) + M_j(t). \tag{2.6}$$

$$Tag^q_{jk}(t) = \lambda\gamma Tag^q_{jk}(t-1); k \neq s. \tag{2.7}$$

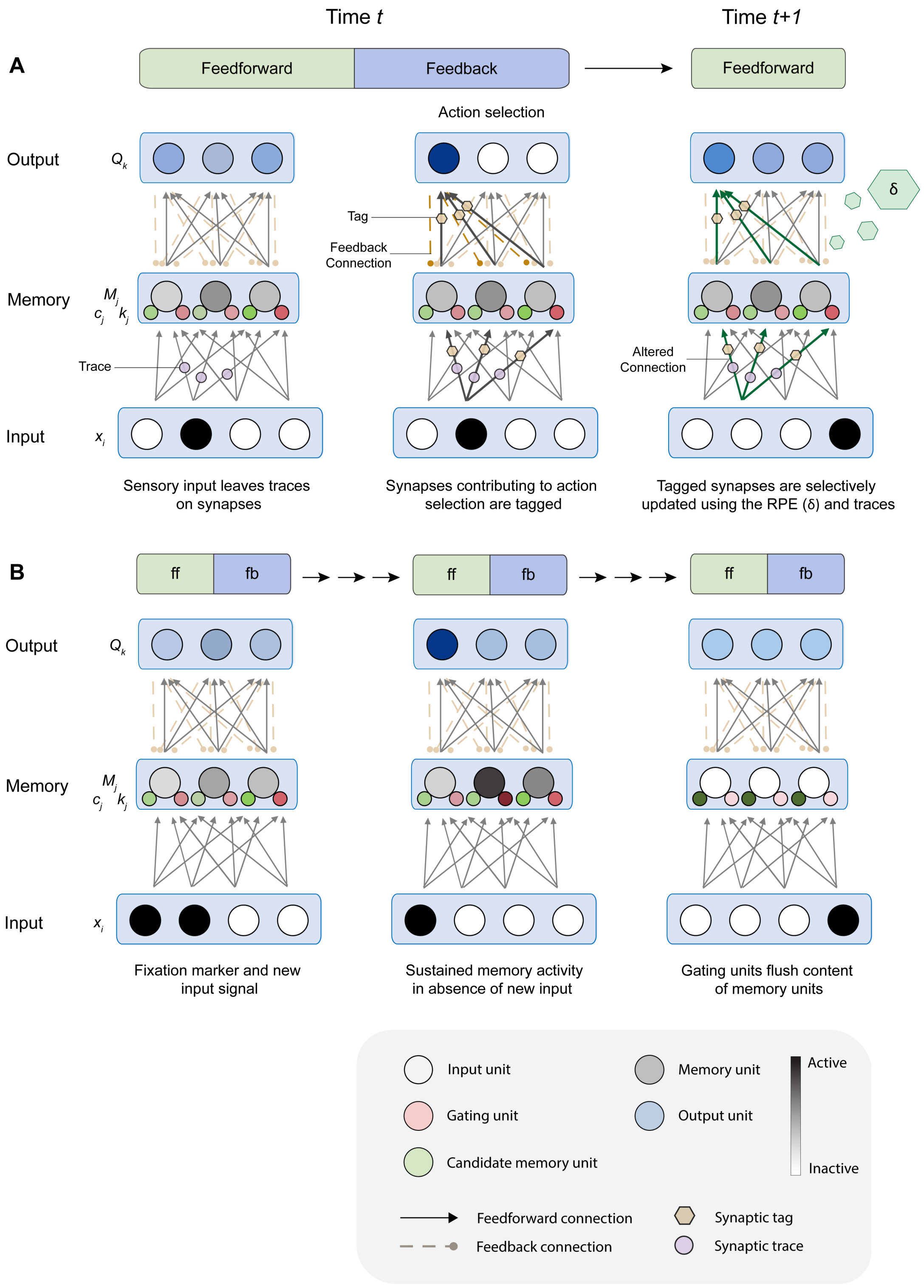
Time t
Time t+1
A
Feedforward
Feedback
Feedforward
Action selection
Output
Q_k
Tag
Feedback Connection
δ
Memory
M_j
c_j
k_j
Trace
Altered Connection
Input
x_i
Sensory input leaves traces on synapses
Synapses contributing to action selection are tagged
Tagged synapses are selectively updated using the RPE (δ) and traces
B
ff
fb
ff
fb
ff
fb
Output
Q_k
Memory
M_j
c_j
k_j
Input
x_i
Fixation marker and new input signal
Sustained memory activity in absence of new input
Gating units flush content of memory units
Input unit
Memory unit
Active
Gating unit
Output unit
Candidate memory unit
Inactive
Feedforward connection
Synaptic tag
Feedback connection
Synaptic trace

Figure 2.2: The process of learning and remembering in RECOLLECT. A) Formation of synaptic tags and traces. The activation of input units during feedforward processing creates synaptic traces (purple circles) on the connections to gating and candidate memory units. Upon action selection, relevant synapses contributing to the selected actions are tagged (yellow hexagons) by feedback connections. The RPE is released in the form of a global neuromodulator (green hexagons) when the expected reward based on the Q-value of the selected action is different from the actual reward that is received. The tagged synapses are either potentiated or depressed depending on the sign of the RPE. If the reward is higher or lower than expected, the tagged connections are potentiated or depressed, respectively. B) RECOLLECT flexibly remembers or forgets across multiple time steps, each with a feedforward and feedback phase (as shown in A). Memory units increase their activity when new sensory information is acquired. This activity can be sustained over a memory delay if the gating units (small red circles) are active (dark red colour). When a relevant sensory stimulus is shown at the beginning of the trial it can therefore be memorized. Signals that demarcate the end of a trial can decrease the activity of gating units, causing forgetting (light red colour). Dashed lines indicate feedback connections from output units to memory units.

Once a tag is formed, it decays according to two hyper-parameters: the tag decay rate (λ) and the aforementioned reward discount factor (γ; this parameter is identical to the one used for calculating the RPE in equation 5). As a result, synapses contributing to previous actions can still be affected by network updates in subsequent timesteps, but to a smaller extent as time progresses. This aspect of the learning scheme corresponds to the temporal difference TD(λ) algorithm (Sutton, 1988).

Weight update for output units

The weight update ΔW^q_{jk} depends on the tag, the RPE δ and the learning rate (β):

$$\Delta W^q_{jk} = \beta \delta Tag^q_{jk}(t). \tag{2.8}$$

Weight update for candidate memory units

The update of the synapses W^C_{ij} from the sensory inputs to the candidate memory cells, providing new input to the memory units, also depend on the degree to which these input cells contributed to Q-value q_s of the selected action s. Their influence is indirect, through the memory unit j. Plasticity therefore depends on (i) how much memory unit j contributed to the Q-value of the selected action and (ii) the contribution of this synapse on the memory unit j's activity level on the current and previous time steps.

The first of these components is reflected by the feedback connection from the selected action, since feedforward and feedback connections between memory and output units are proportional in strength.

The second component is provided by a synaptic trace. Namely, RECOLLECT (as in AugMEnT; Rombouts et al., 2015) uses a 'trace' to keep track of the synapse's influence on the activity level of memory unit j. We will first describe the properties of the trace before explaining how it combines with the feedback signal from the selected action to create the tag, which together with the RPE determines the synaptic changes.

The trace measures the influence of an input unit on the activity of a candidate memory cell. It is initialised at a value of 0:

$$Trace_{ij}^{C}(0) = 0. \tag{2.9}$$

The influence of the synapse W_{ij}^{C} on the activity of memory unit j depends on the slope of the activation function $\sigma'(Inp_j^C(t))$ $(Inp_j^C(t))$ is defined in equation 1) of the C_j unit at time t, the activity of the input unit x_i, and on the activity of the memory gate k_j, which together define the second term in this equation:

$$Trace_{ij}^{C}(t) = k_j(t)Trace_{ij}^{C}(t-1) + [1 - k_j(t)]x_i(t)\sigma'(Inp_j^C(t)). \tag{2.10}$$

The first term represents a trace of the influence of the synapse on the activity of memory unit j on previous time steps $Trace_{ij}^{C}(t-1)$. The trace of previous influences quickly declines if $k_j(t)$ is small, i.e. if the memory gate is open for new sensory input. If the gate activity is close to 1, the memory is maintained and the same holds for the trace. Note that the trace can be computed locally at the synapse and is used to update the tag at the same synapse.

We can now determine the influence of the trace on the tag, which measures the influence of synapse on the current Q-value estimate q_s, as follows:

$$Tag_{ij}^{C}(t) = \lambda\gamma Tag_{ij}^{C}(t-1) + Trace_{ij}^{C}(t)W_{sj}^{FB}. \tag{2.11}$$

Note that the second term includes W_{sj}^{FB}, which equals the feedback that arrives at the memory unit j through the feedback connection from the winning output unit s. This attentional feedback signal is proportional to the contribution of unit j to the Q-value of the selected action. The first term implements TD(λ) in case λ is larger than 0, just as was described above for the weights between the memory units and the output layer.

The Tag_{ij}^{C} interacts with globally released neuromodulator that signals the RPE δ to determine the weight update, as was also described above:

$$\Delta W_{ij}^{C} = \beta\delta(t)Tag_{ij}^{C}. \tag{2.12}$$

Hence, all signals necessary for this weight update are available locally at the synapse.

Weight update for gating units

We will now consider the plasticity of the connections of the gating units, which are updated equivalently, using tags and traces. The trace is initialised at time 0:

$$Trace_{ij}^{k}(0) = 0. \tag{2.13}$$

The contribution of the synapse W_{ij}^{k} to the activity of the memory unit j depends on the slope of the activation function $\sigma(Inp_j^k(t))$ (with $Inp_j^k(t)$ as defined in equation 2), the activity of the input unit x_i, as well as the difference between the activity of the memory unit at the previous time step $M_j(t-1)$ and the new input to the memory unit C_j, because the activity of the memory gate is irrelevant if the activity of the candidate memory unit is equal to that of the memory unit on the previous time step (as reflected in the second term in equation 14 below). The first term in the equation below represents the influence of the synapse on the activity of the memory unit on previous time steps, $Trace_{ij}^{k}(t-1)$.

$$Trace_{ij}^{k}(t) = k_j(t)Trace_{ij}^{k}(t-1) + [M_j(t-1) - C_j(t)]x_i(t)\sigma'(Inp_j^k(t)). \tag{2.14}$$

The equations for the tag and the weight update are equivalent to those of the connections to the candidate memory units (equations 11 and 12):

$$Tag_{ij}^{k}(t) = \lambda\gamma Tag_{ij}^{k}(t-1) + Trace_{ij}^{k}(t)W_{sj}^{FB}. \tag{2.15}$$

$$\Delta W_{ij}^{k} = \beta\delta(t)Tag_{ij}^{k}. \tag{2.16}$$

Biological plausibility

RECOLLECT uses only local information in its learning rule and has various other properties that were inspired by neurobiology. For instance, the output units in RECOLLECT encode for the Q-value of actions. Neurons coding for action values have been observed in several regions, including the midbrain (Morris et al., 2006), basal ganglia (Hikosaka et al., 2014; Ito & Doya, 2009) and frontal cortex (Cai & Padoa-Schioppa, 2014; Padoa-Schioppa & Assad, 2006; Rushworth et al., 2011).

Moreover, to shape plasticity RECOLLECT makes use of a global neuromodulatory signal that conveys the RPE. Such prediction errors are believed to be generated by midbrain dopamine neurons and support decision-making and learning (Schultz, 2016). Another relevant signal is the sensory prediction error (Keller & Mrsic-Flogel, 2018). Equation 14 includes a comparison between the memory unit activity and the current candidate memory unit $[M_j(t-1) - C_j(t)]$, representing such a sensory prediction error. Other biological features include the tags (also known as eligibility traces), which are used to demarcate synapses that contribute to the winning unit (Gerstner et al., 2018; Yamaguchi et al., 2022). The tag/tracing mechanism is based on neurophysiological findings, such as the influence of neuromodulators and feedback connections on plasticity (reviewed by Roelfsema & Holtmaat, 2018). The learning rule represents a form of Hebbian plasticity (Magee & Grienberger, 2020) that depends on both presynaptic and postsynaptic activity, in combination with the RPE.

In conclusion, RECOLLECT is a biologically inspired model that is equipped with a gated memory that allows for selective forgetting and integration of information over longer timespans. In the Methods section we demonstrate that RECOLLECT closely approximates BPTT, while exclusively using information that is locally available at the synapse.

2.2.2 RECOLLECT selectively gates relevant information into working memory

Our goal was to develop a model that can learn to memorise and forget using a local, biologically plausible learning rule. To investigate how RECOLLECT gates information into its working memory and how it sustains these memory representations over time, the model was trained on the pro-/anti-saccade task from Gottlieb and Goldberg (1999) (Fig 2.3A). This task was previously used to train AuGMEnT (Rombouts et al., 2015), which also used a biologically plausible learning rule but could not forget. Hence, the task is useful to illustrate differences between these models. The task consists of 50% pro-saccade trials in which the model should make a saccadic eye movement to a cued location after a memory delay and 50% anti-saccade trials in which the eye movement must be made in the direction opposite to where the cue appeared.

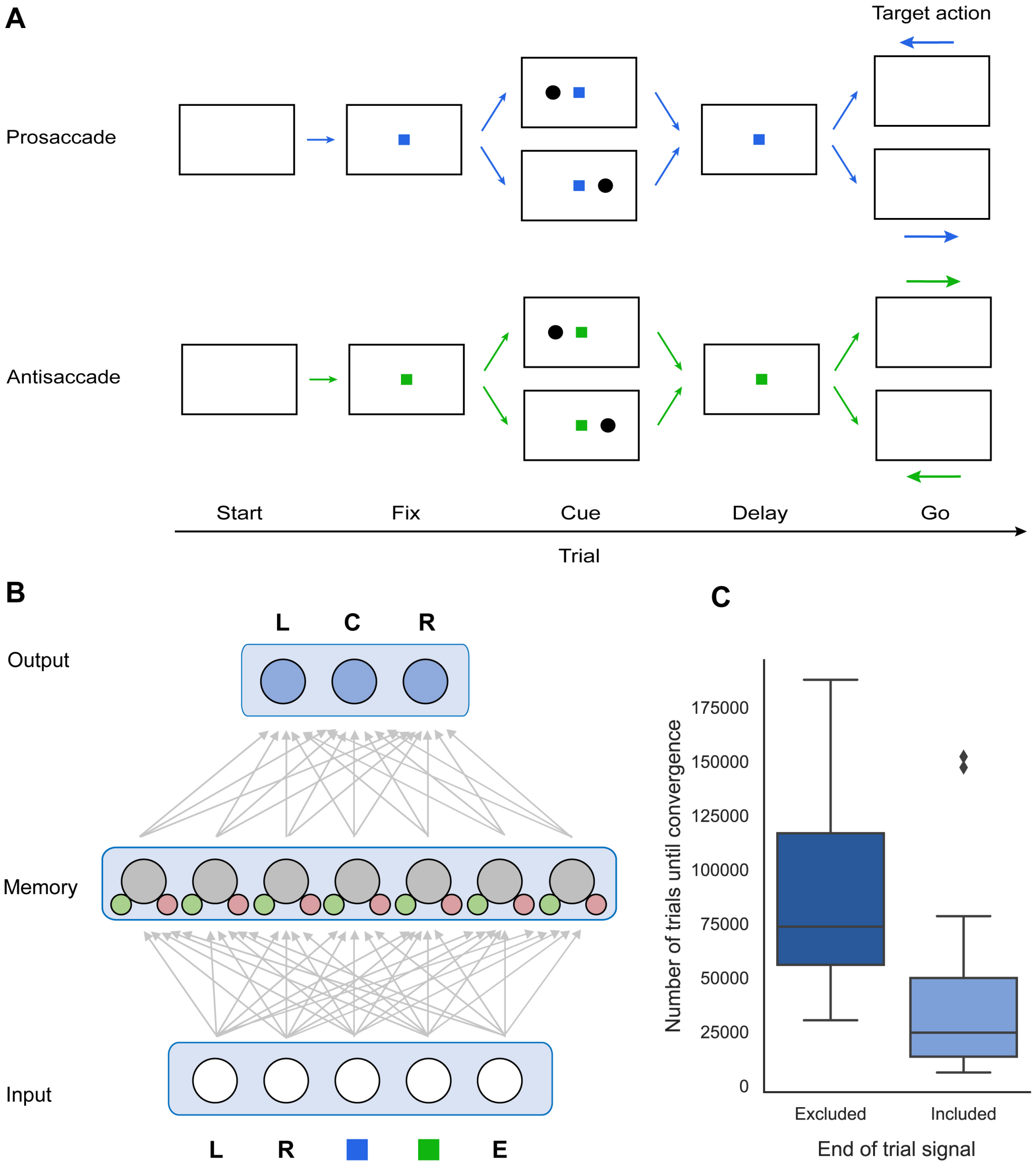
A
Target action
Prosaccade
Antisaccade
Start
Fix
Cue
Delay
Go
Trial
B
L
C
R
Output
Memory
Input
L
R
🟦
🟩
E
C
Number of trials until convergence
175000
150000
125000
100000
75000
50000
25000
0
Excluded
Included
End of trial signal

Figure 2.3: Structure of and performance of RECOLLECT on the pro-/anti-saccade task. A) Structure of the pro-/anti-saccade task. The fixation colour indicates whether a pro-saccade (blue) or anti-saccade (green) to a cue on the left or right side of the fixation mark has to be performed after a memory delay. B) Schematic representation of network architecture. The input layer in RECOLLECT receives information about the colour of the fixation marker (blue and green squares) and the position of the cue (L = left, R = right). An optional fifth input unit encoded the case end-of-episode signal (E). The output layer encodes the three actions that can be taken: gaze directed to the left (L), centre (C) or right (R). C) Number of trials before convergence without an end of episode signal (left) or when it is included in the input (right). Boxes represent the first and third quartiles, with the middle line indicating the median. The whiskers range from the first quartile minus 1.5 times the interquartile range to the third quartile plus 1.5 the interquartile range. Outliers are indicated with diamonds.

The model could direct gaze to the centre of the screen or to a position on the left or the right of the screen, by activating a corresponding unit in the output layer (Fig 2.3B). The task started with an empty visual display, after which either a blue or green fixation marker appeared in the centre of the screen (Fig 2.3A). A blue fixation marker signalled that a pro-saccade would be required and a green fixation marker an anti-saccade. If gaze was not directed to the centre position within 10 timesteps upon presentation of the central cue, the trial was terminated without reward. Otherwise, the model received a reward of 0.2 arbitrary units and was presented with a cue on either the left or the right side of the fixation marker during a single timestep. Once the cue disappeared, a memory delay of 2 timesteps commenced. If gaze fixation was broken before the end of this delay, the trial was aborted without additional reward. If the model kept fixating, the central fixation marker disappeared and the model had to make the appropriate saccade within 8 timesteps to receive a reward of 1.5 arbitrary units. There was an inter-trial interval of one timestep before the next trial started.

Hence, correct performance depended on the saccade direction which was determined by a non-linear combination of the colour of the fixation point and the cue location, which had to be memorised, requiring the maintenance of information until the ‘go’ cue. To prevent interference, the model should forget the cue location before the memory epoch of the successive trial.

There were two input units coding for the possible colours of the fixation marker (one-hot encoding) and two input units for the left or right cue (Fig 2.3B). The network was trained for a maximum of 1.000.000 trials or until convergence. Convergence was established if 1) the model had reached criterion performance (85% correct trials) on the last 100 trials of the four trial types (i.e. pro-saccade left, pro-saccade right, anti-saccade left and anti-saccade right), and 2) when it could perfectly complete all four trial types with its weights fixed and without exploration (i.e. learning was disabled).

We trained 20 networks with 4 input units, 7 memory units and 3 output units (Fig 2.3B) and randomly initialised, fully-connected weights. All networks reached the convergence criterion, indicating that RECOLLECT indeed successfully utilised its working memory. However, more training was required before convergence than in the previous AuGMEnT model although the network size was comparable (see Methods). Specifically, the median number of trials required was 73,614 for RECOLLECT, but only 4,100 for AuGMEnT. We note, however, that there are important differences between RECOLLECT and AuGMEnT. Memory units of AuGMEnT are perfect integrators and their activity is reset at the end of every trial. In contrast, RECOLLECT needs to learn to maintain information during a trial by the appropriate setting of the memory gates, and to later forget before the memory epoch of the successive trial. Hence, RECOLLECT learns about the structure of the environment, how it is composed of trials, as well as when and what to memorise. The comparison with AuGMEnT reveals that its versatile gating mechanism requires additional training time.

We hypothesised that learning with RECOLLECT could accelerate if we would add an explicit cue indicating the termination of a trial, since the network might learn to flush its memory upon receiving this signal, improving the learning process. Indeed, the inclusion of this end-of-trial signal reduced the median number of trials before convergence from 73,614 to 24,657 trials (Z = -2.99, p = 0.003, Wilcoxon signed-ranks test, for 20 randomly initialised networks with and without reset signal) (Fig 2.3C).

To investigate how RECOLLECT solves the pro-/anti-saccade task, we examined the activity profile and tuning of the units. In this analysis, we first increased the memory delay to five timesteps and the intertrial interval to three timesteps, using a curriculum (Materials & Methods).

Units developed selectivity for the type of saccade (pro- or anti-saccade), the location of the visual cue (left or right), and to combine these two types of information to select the appropriate saccade. To investigate how this information can be combined across units to solve the task, we plotted the activity of example units in one of the networks (Fig 2.4G) for the four trial types (Figs 2.4A-F). For instance, the gating unit illustrated in Fig 2.4A responded to left cues and was slightly more active on pro-saccade trials. In general, gating units often showed high activity for a particular feature (e.g. the blue marker, cueing pro-saccades) to facilitate memory while causing forgetting for the opposite feature (e.g. the green marker, cueing anti-saccades). Some units were selective for only one of the four trial types, such as the candidate memory unit in Fig 2.4B, which was most active for anti-saccade trials with a cue on the right. Several memory units developed selectivity for the required saccade direction, coding for the appropriate eye movement during the memory delay on both pro- and anti-saccadic trials. For instance, the memory unit in Fig 2.4C displayed a selectivity for leftward eye movements. As required by the task, the output unit with the highest Q-value was the one coding for the required action. Small differences between the Q-values suffice for convergence, because the network usually selects the action with the highest Q-value. The Q-values for the erroneous actions should eventually evolve to zero if training would continue. Finally, several units coded for the end-of-trial signal (Figs 2.4D-F) so that the network flushed the memories to prevent interference on subsequent trials.

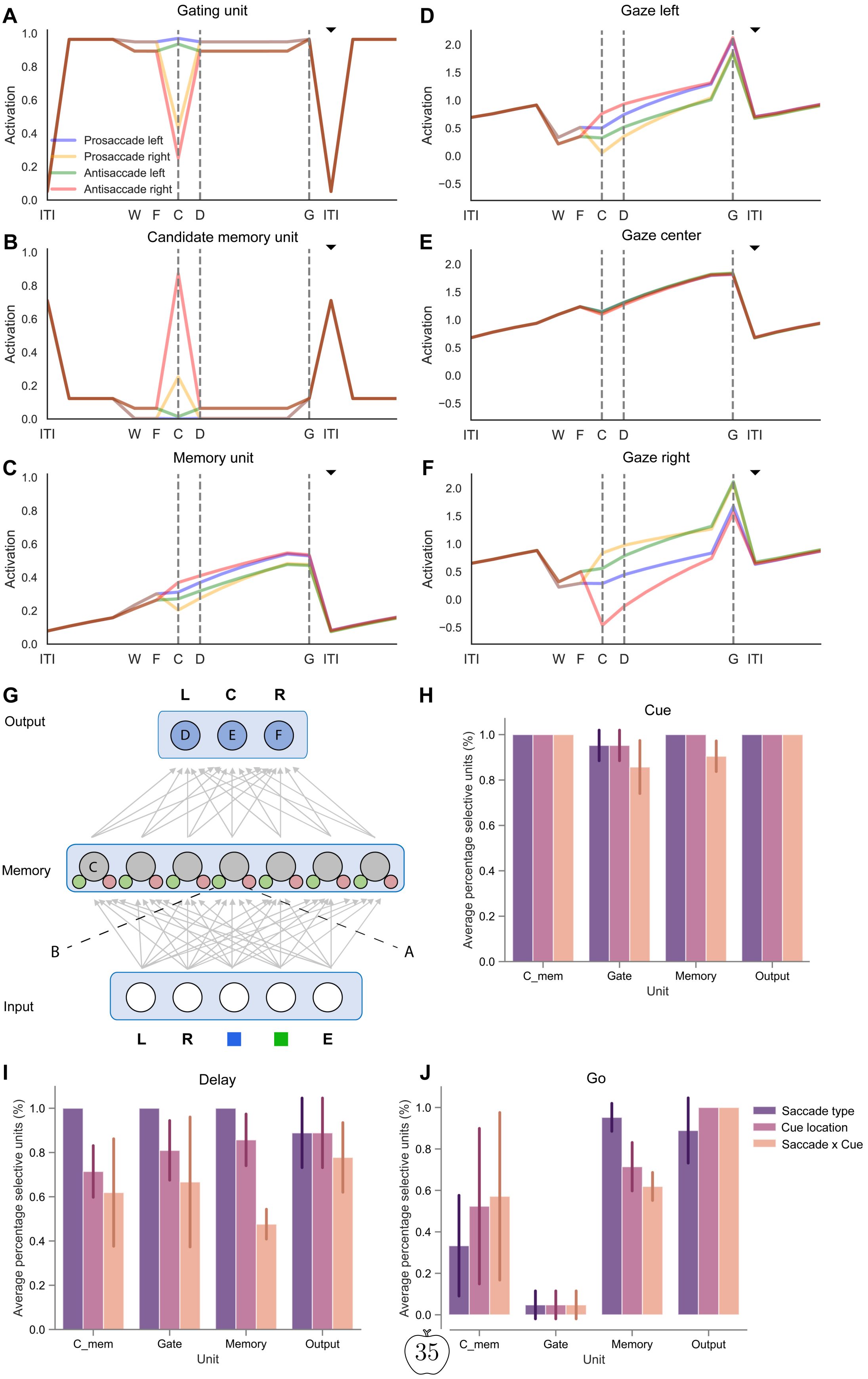
A
Gating unit
Activation
1.0
0.8
0.6
0.4
0.2
0.0
Prosaccade left
Prosaccade right
Antisaccade left
Antisaccade right
ITI W F C D G ITI
B
Candidate memory unit
Activation
ITI W F C D G ITI
C
Memory unit
Activation
ITI W F C D G ITI
D
Gaze left
Activation
2.0
1.5
1.0
0.5
0.0
−0.5
ITI W F C D G ITI
E
Gaze center
Activation
ITI W F C D G ITI
F
Gaze right
Activation
ITI W F C D G ITI
G
L C R
Output
D E F
Memory
C
B
A
Input
L R E
H
Cue
Average percentage selective units (%)
C_mem
Gate
Memory
Output
Unit
I
Delay
Average percentage selective units (%)
C_mem
Gate
Memory
Output
Unit
J
Go
Average percentage selective units (%)
C_mem
Gate
Memory
Output
Unit
Saccade type
Cue location
Saccade x Cue

Figure 2.4: The selectivity and activity of units in networks trained on the pro-/anti-saccade task. A-F) The activity of example units on pro-/anti-saccade trials with a left or right cue are shown in different colours. Pro-saccade trials with a cue on the left (right) are shown in blue (yellow) and anti-saccade trials with a cue on the left (right) in green (red). The black triangle indicates the time step when the end-of-trial signal was given. A) Example of a gating unit that was sensitive for cues on the left side, with strong activity on pro-saccade trials. Note the weak activity during the end-of-trial signal, which causes forgetting. B) A candidate memory unit that responded to right cues on anti-saccade trials. C) A memory unit that prefers trials with leftward saccades. D-F) The output units estimated the Q-value of a leftward saccade (D), fixation (E) and a rightward saccade (F). G) Architecture of RECOLLECT models trained on the pro-/anti-saccade task with labels referring to the example units from one network plotted in panels E-J to illustrate how RECOLLECT solves the pro-/anti-saccade task. H-J) Average percentage (± s.d.) of units selective for saccade type (pro- or anti-saccade), cue location (left or right), and their interaction, across three initialisations of the network. During cue presentation (H), nearly all units are selective for multiple features. During the delay (I), most units are selective for saccade type and a majority is also selective for cue location and the interaction between these factors. During the 'go' epoch (J), only few gating units exhibit selectivity. The selectivity of candidate memory units varies, whereas most memory and output units are selective for both features and their interaction. Labels: ITI = intertrial interval, W = waiting period until fixation is acquired, F = time of fixation, C = cue presentation, D = memory delay period onset, G = go-signal, i.e. the disappearance of the fixation point.

Figs 2.4H-J shows the percentage of units that exhibited significant selectivity for these features and their interaction, across three initialisations of the network shown in Fig 2.4G. As can be observed, most units – irrespective of unit type – were significantly selective for all task features during cue presentation (Fig 2.4H). More variability could be seen during the memory delay period (Fig 2.4I), but in general most units coded for saccade type, with a large number of units also showing selectivity for the cue location, as well as the interaction between cue location and saccade type. Diverging selectivity profiles between unit types primarily emerged during the 'go' phase (Fig 2.4J), wherein gating units exhibited nearly no tuning to task features and only a relatively small number of candidate memory units being selective for saccade type, cue location and their interaction. However, nearly all output units and the majority of memory units were selective for all task features during this phase.

Gottlieb and Goldberg (1999) and Zhang and Barash (2000, 2004) studied the selectivity of neurons in the lateral intraparietal area (LIP) in monkeys during a pro-/anti-saccade task. Gottlieb and Goldberg (1999) found that many neurons in a no-delay version of the task responded to one of the cues and did not show selectivity upon saccade onset (Fig 2.5A), whereas a smaller number of LIP neurons coded for the saccade direction. Zhang and Barash (2004) used a memory delay, and reported a subset of neurons representing the memory of the cue location by firing persistently during the delay (Fig 2.5B). Yet other LIP neurons encoded the required motor response, or a non-linear combination of the stimulus position and the required eye movement. Units of networks trained with RECOLLECT expressed all these activity profiles (Figs 2.5C-D).

Other neurophysiological studies demonstrated that the duration of the persistent activity depends on the length of the period that the stimulus needs to be remembered. When the memory delay is extended the memory activity of LIP neurons persists longer (Gnadt & Andersen, 1988) (Fig 2.6A). To investigate whether RECOLLECT displays a similar behaviour, we trained a network with varying memory delays (from one to five timesteps) (Fig 2.6B). The duration of persistent activity depended on the length of the delay, after which it declined upon the end-of-trial signal.

We conclude that RECOLLECT can train networks on the pro-/anti-saccade task. These networks learn to memorise and forget when necessary and use persistent activity to code for memories in a similar manner as neurons in the brain.

2.2.3 RECOLLECT exhibits learning-to-learn on a reversal bandit task

We next investigated whether RECOLLECT can be used to train networks to learn-to-learn on a reversal bandit task (see Fig 2.7A). This task has previously been used to assess meta-learning (e.g. Wang et al., 2018) because its overarching reward structure can be learned and exploited.

On each trial during the task, the model chooses between two levers, of which one has a high (75%) reward probability and the other has a low (25%) reward probability. The task consisted of two contexts because the reward probabilities could reverse. Episodes consisted of 100 lever pulls and after every episode the reward probabilities were either reversed (reversal bandit), or randomly reassigned (random reversal bandit). The network had to sample the levers to assess the context, i.e. determine which one yielded the higher reward and then harvest rewards by consistently pulling this lever until the end of an episode. The reversal bandit is easier than the random reversal bandit because the network can exploit the predictable reversal between successive episodes.

Successful meta-learning on this task implies that a trained model can quickly (i.e. within one or just a few trials) switch to the new context at the start of a new episode by associating each context with a memory state. The model should

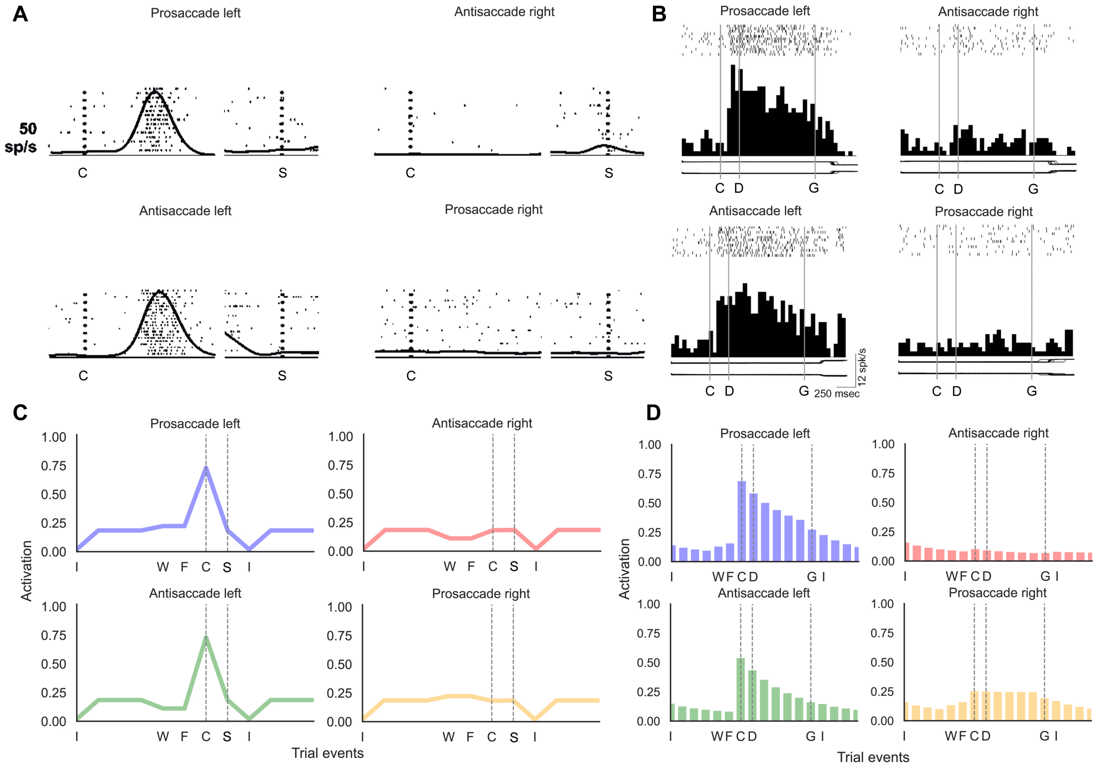

Figure 2.5: Comparison between neuronal data recorded in the parietal cortex of monkeys and RECOLLECT on the pro-/anti-saccade task. A) Example neuron in area LIP in the parietal cortex of a monkey coding for a visual cue on the left (adapted from Gottlieb & Goldberg, 1999). The left and right dashed lines indicate cue and saccade onset, respectively. B) The activity of an example LIP memory cell for coding for cue location (adapted from Zhang & Barash, 2004). Dashed lines signify cue onset, the memory delay period, and go-time (disappearance of the fixation cue, prompting saccade onset). C) Candidate memory unit in RECOLLECT coding for the left cue. D) Memory unit in RECOLLECT. Labels: I = intertrial interval, W = waiting period until fixation is acquired, F = time of fixation, C = cue presentation, D = memory delay period, G = go, S = saccade onset. Note that the conditions are ordered in the same way in panels C and D as the neurophysiological data in A and B, respectively.

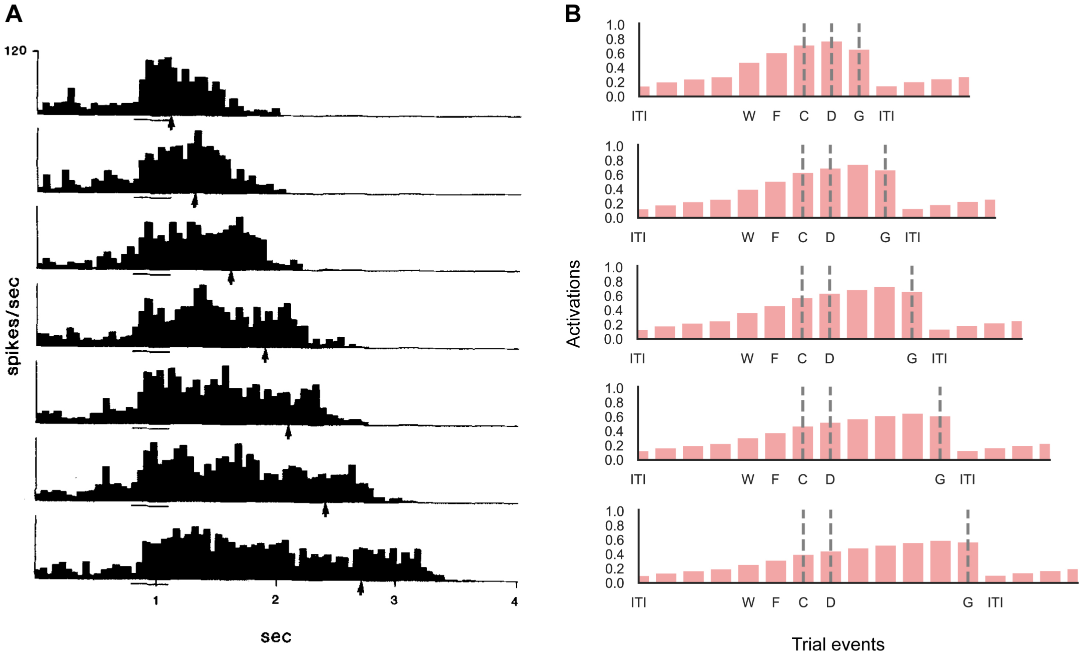

Figure 2.6: Sustained memory delay activity in the parietal cortex of monkeys and of RECOLLECT units on the pro-/anti-saccade task. A) Neurons in lateral interparietal cortex (LIP) in the parietal cortex of macaque monkeys persistently fire for the length of the memory delay of the pro-/anti-saccade task (Gnadt & Andersen, 1988). B) Memory units in RECOLLECT also exhibit persistent firing across increasingly long delays (1, 2, 3, 4 or 5 timesteps), which ceases when the memory epoch ends. Labels: ITI = intertrial interval, W = waiting period until fixation is acquired, F = time of fixation, C = cue presentation, D = memory delay period, G = go cue, which was cued by the disappearance of the central fixation point.

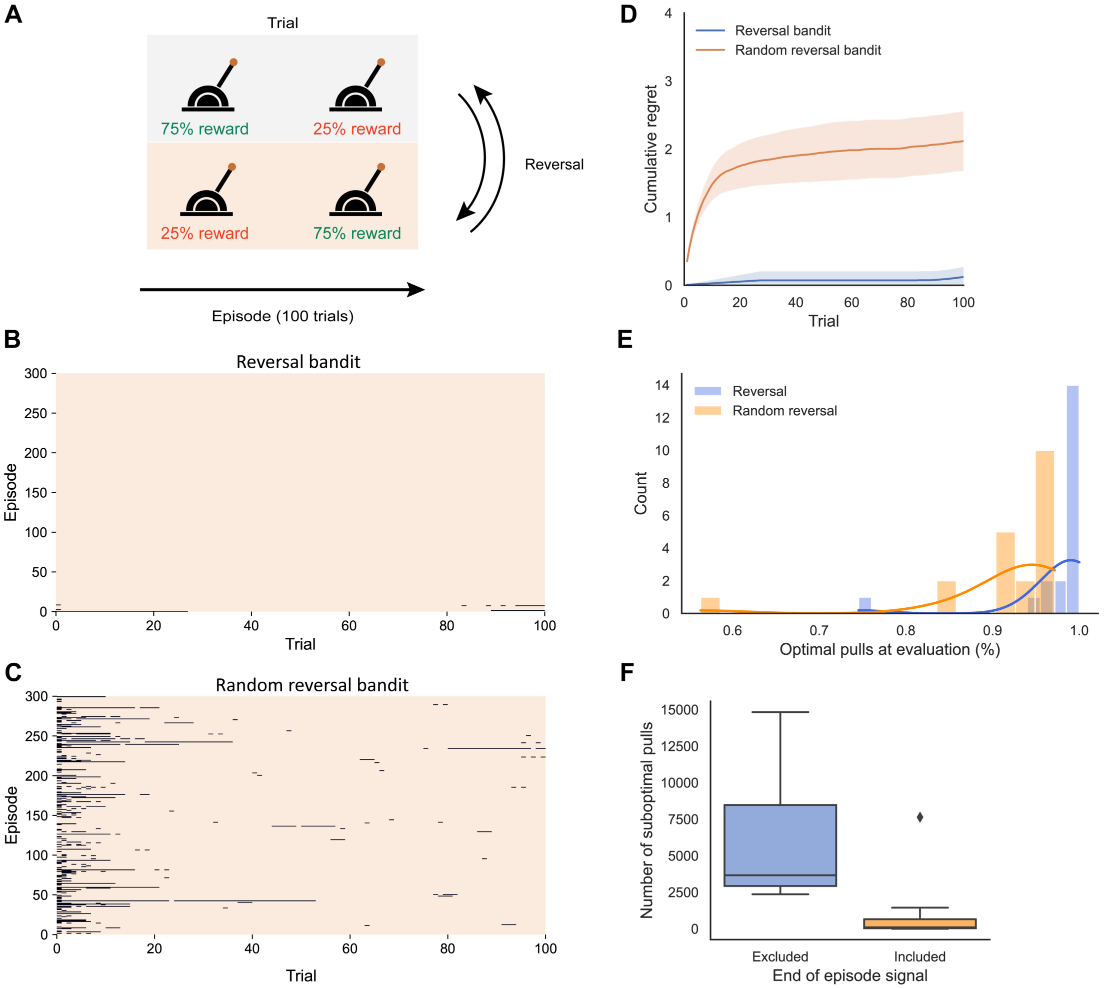

Figure 2.7: Structure of and performance of RECOLLECT on the reversal bandit task. A) Two-armed bandit reversal task. In the random version, we randomly assigned reward probabilities to the two levers when a new episode started. B,C) Performance on example networks after training on the reversal bandit (B) and random reversal bandit (C) at evaluation (99.8% and 97.2% optimal pulls, respectively). The networks were initialised with the same seeds. Orange and black regions denote optimal and suboptimal choices, respectively. Trials are shown on the x-axis, with 100 trials per episode, and successive episodes on the y-axis. D) Cumulative regret (±95% confidence interval) on the reversal bandit task (blue) and the random version (orange). E) Histogram of the percentage optimal pulls on evaluation trials of the reversal bandit and random reversal bandit for the same 20 random seeds. F) The number of suboptimal pulls (300,000 pulls in total) in the non-random reversal bandit task is lower when an end of episode signal is included, cueing the model that the reward contingencies reverse.

change strategy when the preferred lever starts giving less reward, but the model needs to integrate information across several trials in which reward is unexpectedly omitted, because the best choice is only rewarded on 75% of the trials. The model could learn to use its working memory to represent the context by integrating information about the reward probability of the levers, as opposed to the much slower solution of relearning its weight structure upon every switch in the context. To facilitate meta-learning, the network had access to the action that it took on the previous timestep and the reward it received, which is informative about the current context. We also provided a signal that an episode had ended.

We trained RECOLLECT with 4 input units, 4 gating, candidate memory and memory units each (5 for the random reversal bandit). The two output units represented the two lever actions. We presented 20,000 episodes of 100 trials each (as in Wang et al., 2018). Once the training phase was completed, learning and exploration were disabled and the model completed an additional 300 evaluation episodes. We evaluated performance as the number of choices of the low-rewarding (i.e. suboptimal) lever on 20 random initialisations of the network. For comparison with Wang et al. (2017, 2018), we also provide a measure of cumulative regret. Regret occurs when the action taken deviates from the optimal action (under hindsight) and a reward is not obtained. Cumulative regret refers to the cumulative loss of these expected rewards over time (Pepels et al., 2014).

Figs 2.7B-C illustrate the suboptimal pulls as black line segments during the evaluation phase of two networks that were trained on the reversal and random reversal bandit tasks, respectively. The example network trained on the reversal bandit task learned to select the correct lever upon episode reversals almost perfectly. Suboptimal pulls only occurred either at the beginning of an episode or just before the end. There were more suboptimal arm pulls on the random reversal task, which were concentrated at the beginning of episodes. While RECOLLECT tended to select the correct lever thereafter, there were also some episodes with errors at other time points. We predicted that these occurrences might occur due to the absence of an expected reward on several consecutive trials, thereby falsely suggesting a context switch. In accordance with this view, the average reward received on the previous three trials was 0.74 when a correct response was made but only 0.23 when incorrect choices were made.

Networks trained on the reversal bandit task (see Fig 2.7E) achieved a median accuracy of 99.7%, with some networks reaching 100% optimal pulls. As expected, the accuracy on the random reversal bandit was significantly lower at 94.9% (Wilcoxon Signed-Ranks test, z = -3.06, p = .002). Hence, RECOLLECT exploited the regularity of the reversal bandit task, in which the episodes always alternated and the network did not have the sample the new reward structure when a new episode started. The performance of RECOLLECT on the random reversal bandit (see Fig 2.7D) was only slightly below that of long-short term memory (LSTM)-based architectures trained in the same learning-to-learn setting, with an average cumulative regret of 2.1 for RECOLLECT (97.2% optimal pulls) versus 1.1 in

Wang et al. 2017; 98.5% optimal pulls). This is remarkable, given the reduced computational complexity of RECOLLECT and its use of a local, biologically plausible learning rule.

To investigate the effect of the end-of-episode signal, we trained 20 networks with and without this signal on the non-random reversal bandit (Fig 2.7F). At evaluation, the median number of suboptimal pulls of these networks was 99 (of a total of 300,000 pulls) in the presence of the end-of-episode signal, which was significantly lower than the median number of 3,661 suboptimal pulls without this signal (Wilcoxon signed-ranks test, Z = -3.47, p < .001). Hence, RECOLLECT capitalises on the end-of-episode signal to increase its performance.

We analysed a smaller network, with only two memory units, to gain insight into how it solves the reversal bandit task. We plotted the average activity ($\pm SEM$) across episodes of network units for left and right high-rewarding episodes before and after reversals for an example network (Fig 2.8). We will first discuss activity in the absence of an end of episode signal (Fig 2.8A). Before the reversal, the activity of the Q-value unit coding for the highly rewarded action was higher than that of the other Q-value unit. This pattern reversed slowly after the switch (t = 0) until the unit for the now appropriate action was more active (around 4 trials after the reversal). This strategy reflects the accumulation of evidence for a switch in context. Because the correct lever is only rewarded 75% of the time and the incorrect lever yields a reward on 25% of the trials, a single rewarded or unrewarded lever pull does not give reliable information about the context. Instead, RECOLLECT needs to integrate outcome information across a few trials until it can determine that the context changed. Note that the Q-values exceed the reward value the network can receive on a single trial. Instead, these values reflect the discounted reward expectation across a number of trials given that a particular action is chosen.

The activity of Q-value units depended on the activity of memory and gating units, which had comparable activity time courses. Interestingly, the activity of one of the gating units was close to one until the reversal, which indicates that the memory was maintained (Fig 2.8A). When the episode ended, the activity of the gating unit decreased, permitting an influence of the candidate memory units and the reversal of activity of the gating and memory units.

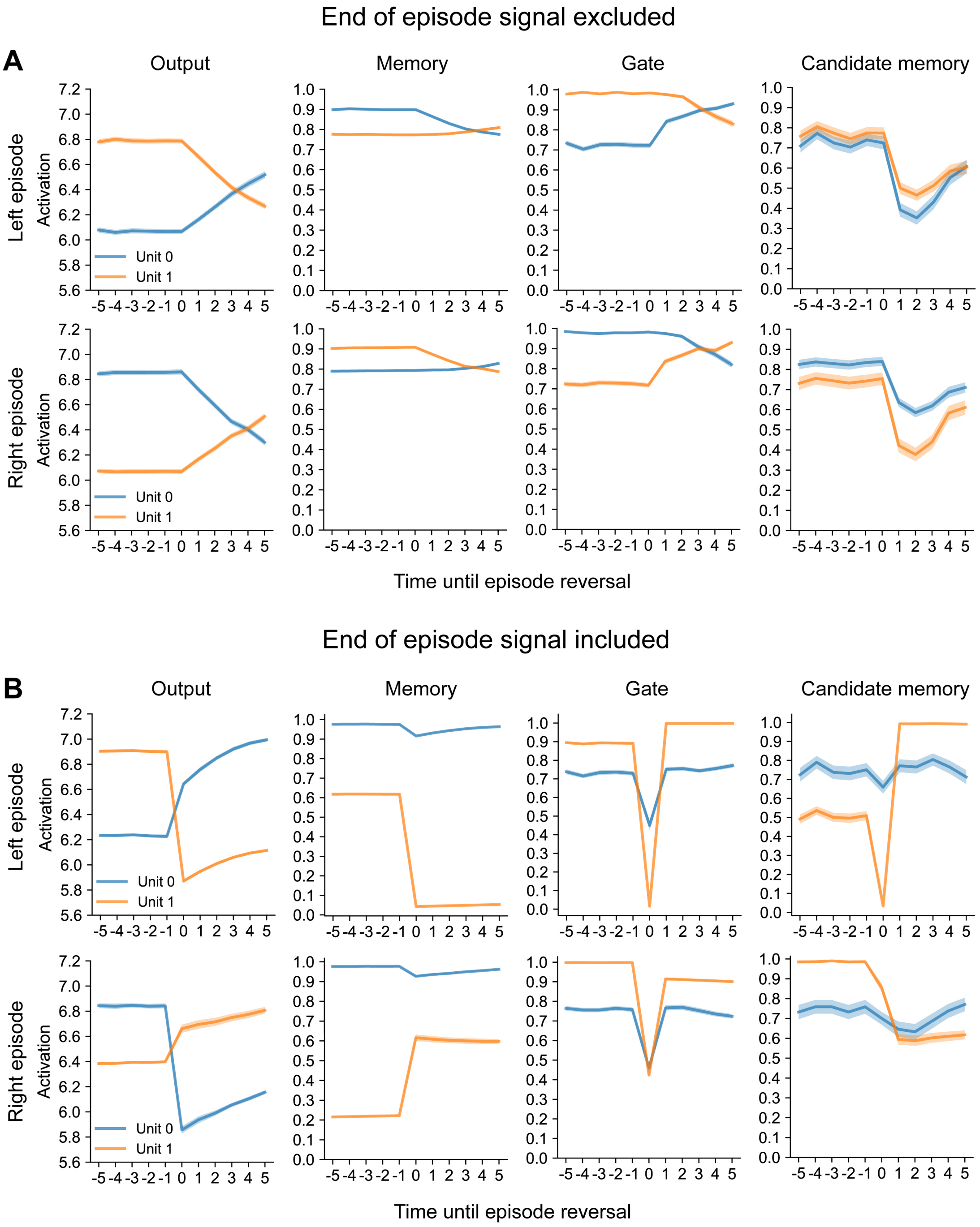
End of episode signal excluded
A
Output
Memory
Gate
Candidate memory
Left episode
Right episode
Activation
Unit 0
Unit 1
7.2
7.0
6.8
6.6
6.4
6.2
6.0
5.8
5.6
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-5 -4 -3 -2 -1 0 1 2 3 4 5
Time until episode reversal
End of episode signal included
B
Output
Memory
Gate
Candidate memory
Left episode
Right episode
Activation
Unit 0
Unit 1
Time until episode reversal

Figure 2.8: The average of example units in the reversal bandit task. (A) Example network trained in the absence of an end-of-episode signal. The activities of the two Q-value units in the output layer reverse after the episode. The network has to gather evidence across trials in its memory units (second column) based on the reward contingency that the context changed, because only 75% of the optimal choices are rewarded. The third and fourth columns show the activity of the gating and candidate memory cells, respectively. The shading shows $\pm SEM$. (B) The reversal of the state of the network is abrupt in the presence of the end-of-episode signal, which changes the memory state of the network within one trial.

The activities of the two candidate memory units indicated selectivity for the context (Fig 2.8A). Their activity decreased upon the absence of expected reward due to the change in context, followed by a slower recovery. Hence, the network learned to represent the task context in its working memory by integrating information across chosen actions and obtained rewards across a number of successive trials, in the absence of explicit reset signals.

The activity of the network that was trained with an end of episode signal was drastically different (Fig 2.8B). The switch in the activity of Q-value units occurred within a single trial, indicating that the network learned the significance of the end-of-episode signal and efficiently changed its working memory to select the correct, alternative lever in the successive episode. In the example network, one of the memory units exhibited sharp decreases and increases upon episode reversals for left and right episodes, respectively (orange in Fig 2.8B), driving the change in the Q-values in the output layer. Both gating units exhibited steep declines in activity in response to the end-of-trial signal. Finally, one of the candidate memory units (orange in Fig 2.8B) was very active during right high-rewarding episodes, and less during left high-rewarding episodes. In summary, RECOLLECT rapidly switched between memory states in the presence of an end of episode signal, which improved the efficiency on the reversal bandit task.

Finally, we compared the behaviour of networks trained with RECOLLECT on the non-random reversal bandit task to the choices made by rats trained on a similar task. Brunswik (Brunswik, 1939) trained rats on serial-reversal task on a T-maze, with two arms that were baited with different rewards. On the first 24 trials, one arm was always rewarded and the other arm was never rewarded. Rewards were reversed for the subsequent 16 trials. This was followed by several reversal episodes of 8 trials each, until the rats completed a total of 8 episodes. During the first episode the performance gradually increased (Fig 2.9A). The first reversal caused a sharp increase in errors, which then declined, a pattern that repeated for every reversal afterwards. Interestingly, the rats required fewer trials to accommodate the later switches, indicating that the rats learned-to-learn this task.

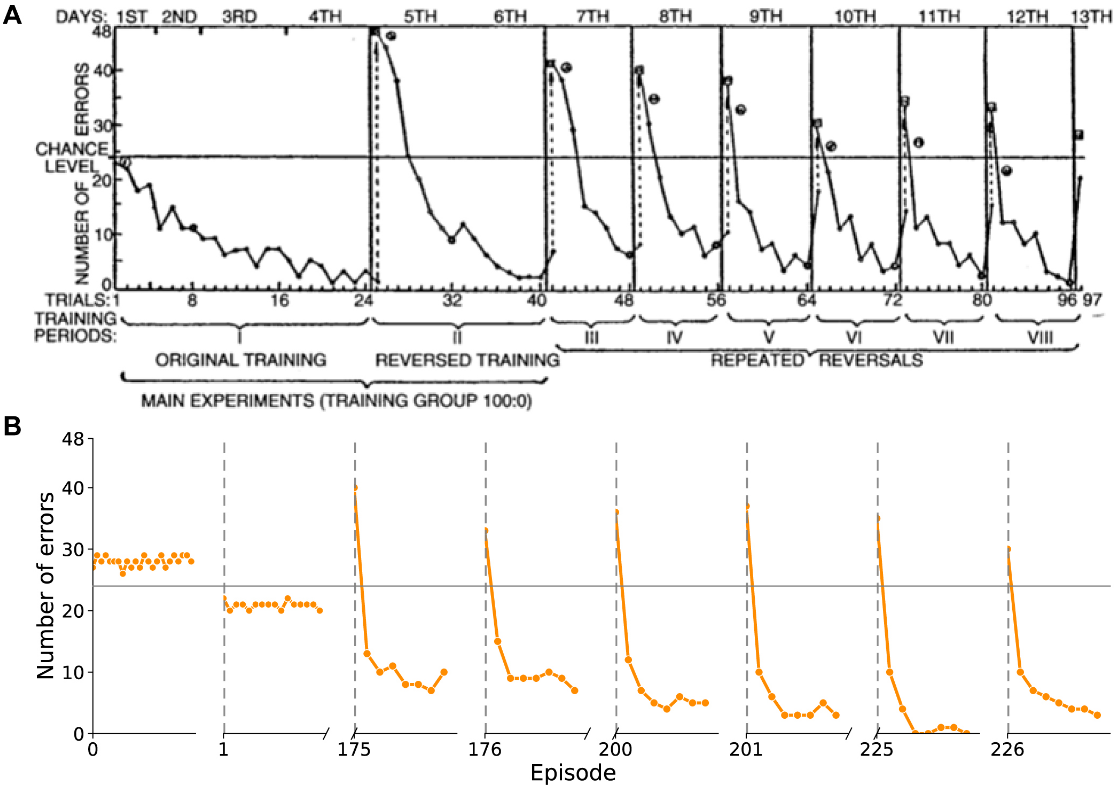

Figure 2.9: Performance in a reversal task of rats and of networks trained with RECOLLECT. (A) Learning decreases the number of errors that rats make on a reversal bandit task (data from Brunswik, 1939). Y-axis, number of rats (total 48) making an error. (B) Networks trained with RECOLLECT. Each data point represents the number of errors per trial in successive episodes, summed across 48 networks. We plotted the first 24 trials of the first episode, 16 trials after the first reversal and 8 trials of subsequent episodes.

We next analysed the appearance of switching behaviour for 48 networks trained with RECOLLECT (Fig 2.9B), baiting the highly rewarding lever on 100% of trials with a reward and the other lever on 0% of the trials. Learning in RECOLLECT is slower than that of rats, and we therefore plotted the number of errors in the first episode, the first reversal and the 175th, 200th and 225th episodes with the subsequent reversals (episode 176, 201 and 226). This difference in learning rate is presumably due to the fact that RECOLLECT is initialised *tabula rasa* at time of training, unlike the rats. The evolution of behaviour in RECOLLECT, however, was similar to that of the rats. Episodes started with many errors, after which the accuracy improved in later episodes, similar to what was observed by Brunswik et al. (1939).

In conclusion, RECOLLECT can successfully train networks on the reversal bandit task in a way that is comparable to non-biologically plausible models. Moreover, the progression of learning is qualitatively similar to the behaviour of rats in a reversal task.

2.3 Discussion

We developed a novel gated memory network that could memorise task-relevant information, forget it when appropriate and learned-to-learn in a biologically plausible manner. The model incorporated a version of the light-gated recurrent unit (Light-GRU; Ravanelli et al., 2018) and its learning rule was based on AuGMEnT (Rombouts et al., 2015) that uses a combination of attentional feedback and neuromodulators that code for the RPE. The result is a biologically plausible form of learning that is similar to backpropagation-through-time. In RECOLLECT, all information used to update the network is locally available at the synapses of the network. Specifically, candidate memory, gating and memory units could be considered part of the same cortical column or loop through subcortical structures and the attentional feedback signal as a locally available signal in that column. Indeed, neurons in the different layers of the cortex play specific roles in representing sensory input, attention and working memory (van Kerkoerle et al., 2017). Hence, RECOLLECT provides a biologically plausible learning rule for gated memory networks, which differentiates it from AuGMEnT, which required a reset of its memory after every trial.

The main advantage of the RECOLLECT architecture with memory gates is its flexibility. Whereas its predecessor AuGMEnT remembers by default and cannot learn to forget (Rombouts et al., 2015), RECOLLECT learns to strategically flush its memory when useful. The Light-GRU is one of the simplest memory units with this property (Ravanelli et al., 2018), making it a useful component of neuronally plausible models to study the mechanisms underlying memory and forgetting compared to larger and LSTM-based networks, which are more difficult to interpret (Wang et al., 2018). We note, however, that the precise mapping of the gating mechanisms onto the circuits underlying memory and forgetting in the brain remains to be elucidated. Previous neuroscientific studies revealed multiregional loops between the cortex, thalamus and striatum for working memory (Bolkan et al., 2017; Rusu & Pennartz, 2020; Schmitt et al., 2017; Wang et al., 2021). Recent evidence also points towards a role of the loop through the cerebellum in working memory (Brissenden et al., 2021; De Zeeuw et al., 2021; Gao et al., 2018; Schmahmann, 2019). These loops have also been implied in reversal learning (Parker et al., 2022; Tuite et al., 2022). More research is needed to fully comprehend how these circuits effectuate working memory and forgetting. The learning rule for the gating connections compared the memory and the new input (equation 14). This comparison plays a prominent role in theories of predictive coding (Keller & Mrsic-Flogel, 2018), thereby establishing a new link between theories of predictive coding and biologically plausible learning rules.

We tested RECOLLECT on a pro-/anti-saccade task, and found that the model flexibly selects which information to remember during a delay. Moreover, RECOLLECT learned to flush its memory at the end of a trial to prevent interference of the memories on subsequent trials, representing an improvement over the AuGMEnT model. A comparison of units in networks trained with RECOLLECT to neurophysiological data revealed many similarities. Units developed selectivity for the colour of the fixation marker and the position of the cue, as well as persistent firing coding for the relevant features, just as been observed in the visual and parietal cortex of monkeys (Gnadt & Andersen, 1988; Gottlieb & Goldberg, 1999; Zhang & Barash, 2004). Thus, RECOLLECT is not only biologically plausible given its reliance on neuromodulators and attentional feedback signals, but networks trained with RECOLLECT develop units that resemble neurons in the brains of animals that have learned the same tasks.

We used a reversal bandit task to test whether RECOLLECT learned-to-learn. Networks trained with RECOLLECT sampled the environment to gauge which of the two levers yielded the highest reward, and it then consistently chose this lever until the end of the episode. Moreover, the model's behaviour during learning was reminiscent of how rats learn the reversal bandit task (Brunswik, 1939). There was an initial increase in errors upon the start of a new episode that decreased over the course of the episode. These errors declined more quickly as training progressed, indicating a similar progression of learning-to-learn in the model and in rats.

An interesting observation pertained to the role of the end-of-trial signal in the pro-/anti-saccade task and the end-of-episode signal in the reversal bandit task. These signals enhanced performance by providing a signal that it is time to update the memory state; thereby simplifying the problem, because the network did not have to integrate information about the relation between stimulus, response and reward to detect a reversal. Likewise, cells in the prefrontal cortex have been shown to represent action sequence boundaries by increased firing rates following the end of the sequence (Fujii & Graybiel, 2003). We found that RECOLLECT networks took advantage of cues signalling a reversal, by rapidly switching to the new strategy. The network also learned to integrate information about the rewards across a number of trials, when the change in the reward contingencies was not signalled explicitly. Hence, RECOLLECT parallels aspects of animal learning such as the identification of sequence boundaries. The accumulation of evidence across trials for a switch of context resembles the activity of neurons in the anterior cingulate cortex of monkeys, which also accumulate evidence based on delivered rewards that the context might have changed (Kawai et al., 2015; Shima & Tanji, 1998).

We here only tested RECOLLECT with a single layer with memory units. Future work could expand RECOLLECT for more complex tasks with multiple memory layers for simple and more complex features. Furthermore, while RECOLLECT consistently converged on the pro-/anti-saccade task, learning was slower than with the previous AuGMEnT architecture (Rombouts et al., 2015), which remembers by default. Similarly, RECOLLECT performed slightly less well on the random reversal bandit than LSTM-based networks trained in the same learning-to-learn setting. These differences are partially explained by the extra information that was given to the previous models. For example, in the study on AuGMEnT (Rombouts et al., 2015) and in Wang et al. (Wang et al., 2018), the network state was reset at the end of each trial. In a variant of AuGMEnT that had to learn to reset its working memory itself, learning was slower than in standard AuGMEnT (Rombouts et al., 2014). RECOLLECT stands out because it learned the time structure of the task, what to remember and when to forget it. The network took advantage of end-of-trial signals, but learning was even possible when such a signal was not presented.

We implemented a few modifications to the Light-GRU units (Ravanelli et al., 2018). The main change is that we excluded recurrent weights from memory units to other memory and gating units. This modification allowed the correspondence to BPTT (see equations 9-16) in a simpler model. Such simplicity sometimes enhances performance (Dey & Salem, 2017; Jozefowicz et al., 2015) and RECOLLECT learned the tasks that we studied here without these additional connections. Nevertheless, the RECOLLECT learning rule is compatible with architectures in which these connections are present and future studies could include them, because they might benefit learning of more complex tasks.

There are other learning rules and models that approximate backpropagation-through-time (e.g. Bellec et al., 2020; Nicola & Clopath, 2017). RECOLLECT uses the same approximation as AuGMEnT and also e-prop (Bellec et al., 2020), which has been used to train long-short term memory models in reinforcement learning settings. There are a number of important differences between RECOLLECT and e-prop. Firstly, RECOLLECT incorporates synaptic tags that implement the faster TD(λ) algorithm, rather than the simpler TD(0) method (Seijen & Sutton, 2014). Secondly, e-prop requires each unit to be connected to an output unit to propagate the error signal. Hence e-prop cannot train the lower layers of deeper networks, effectively limiting the approach to shallow networks. In contrast, RECOLLECT can be extended to deeper networks, just like AuGMEnT (Rombouts et al., 2015) and BrainProp (Pozzi et al., 2020), and hence to more complex tasks. Thirdly, RECOLLECT uses the Light-GRU unit, which is much simpler than the long-short term memory units that were used by Bellec et al. (Bellec et al., 2020). There are also studies investigating learning-to-learn in spiking architectures (Bellec et al., 2018, 2019; Scherr et al., 2020; Schmidgall & Hays, 2023; Subramoney et al., 2021), but we note that these still rely on BPTT for training or are less straightforward to implement in the brain because they use second-order gradients in the outer loop training process (i.e. the overarching learning problem where knowledge is accumulated over multiple learning experiences rather than just in a single trial), rather than the more biologically plausible meta-reinforcement learning method formalised by (Duan et al., 2016; Wang et al., 2018). Finally, the previous 'WorkMATe' model (Kruijne et al., 2021) also used the AuGMEnT learning rule in a model for working memory. The mechanisms for memory and forgetting differ substantially between WorkMATe and RECOLLECT. WorkMATe relies on complex gated memory stores for sensory stimuli, which are updated in an all-or-nothing manner. A separate output module chooses whether new stimuli are encoded in one of the memory store blocks or forgotten. Hence, stored stimuli override previous memory content in WorkMATe, making memorising and forgetting less flexible than in RECOLLECT.

In conclusion, RECOLLECT is a novel gated neural network that only uses information that is available locally at the synapse to learn how to use its working memory flexibly and learn-to-learn in a manner that is reminiscent to animal learning. It presents a biologically plausible alternative to more traditional gated memory networks such as long-short term memory. RECOLLECT thereby contributes to our understanding of how working memory, forgetting and learning-to-learn are implemented by the brain.

2.4 Materials & Methods

2.4.1 Architecture details

Activation function

A sigmoid activation function determined the activity of gating units and candidate memory units:

$$\sigma(input_j(t)) = \frac{1}{(1 + exp(-(\rho \cdot input_j(t))))}, \tag{M.1}$$

where ρ represents the slope of the sigmoid. The value of ρ was set to 2 in all experiments.

Learning rate

The learning rate is shown in Table 1. We noticed that rapid plasticity of gating units decreased the stability of learning. We therefore set the learning rate of synapse onto gating units at a lower value than those of other connections.

Network parameters

During the initialisation, all biases (i.e. for the gating units, candidate memory units and output units) were set to one. For the other parameters, a grid search with a limited set of a priori chosen values was conducted for parameter optimisation. For this, the standard learning rate for all units (β), the learning rate specific to the gating units (β_{gate}) and the tag decay rate (λ) were particularly important. Learning benefitted from lower values for these hyper-parameters in the bandit paradigms (especially the random reversal bandit), because of the more conservative updates in times of uncertainty and preventing premature decisions for a lever before sufficient information has been gathered. Unless otherwise indicated, the parameters used for the experiments were as follows:

	Pro-/anti-saccade task	Reversal bandit	Random reversal bandit
Exploration rate (ϵ)	0.025	0.025	0.025
Number of input units (including end of trial/episode signal)	5	4	4
Number of Light-GRU units	7	4	5
Number of output units	3	2	2
Learning rate (β)	0.1	0.01	0.005
Learning rate of gating units (β_{gate})	0.006	0.006	0.0005
Discount factor (γ)	0.9	0.9	0.9
Tag decay rate (λ)	0.4	0.2	0.1

Table M.1: RECOLLECT hyperparameters for each task (variant).

Pro-/anti-saccade task

To facilitate comparison with Rombouts et al. (2015), simulations regarding performance (Fig 2.3) on the pro-/anti-saccade task were performed using an intertrial interval of 1 time step and a memory delay of 2 time steps. In further stimulations (except for Fig 2.6) we used an intertrial interval of 3 and memory delay of 5 timesteps so that the neural activations during memory delay and after the end of trial signal could be studied more closely. A curriculum was used to achieve these longer memory delays. Specifically, we started with a delay of 1 time step. After the model reached criterion performance (85% correct trials on the previous 100 trials of each trial type), the memory delay was set to 2 time steps and then to 4 time steps until the final memory delay of 5 time steps was reached. Networks contained 7 Light-GRU units, and each of them was composed of a gating, candidate memory and memory unit. However, for Figs 2.5C-D 12 Light-GRU units were included. AuGMEnT was trained with 3 regular hidden units, 4 memory hidden units, and special input units which were either following the input ($N = 4$, instantaneous input units) or responded to the on- and offset of stimuli ($N = 8$ transient input units (Rombouts et al., 2015). The total number of trainable weights was 75 for AuGMEnT and 94 for RECOLLECT.

The no-delay variant of the pro-/anti-saccade task for Fig 2.3A was implemented by first showing the fixation marker (F), followed by the cue without fixation marker (C). The disappearance of the cue prompted the saccade (S).

Reversal bandit

In order to understand how RECOLLECT solves the reversal bandit, the activation plots and neural data comparison figure were created with small networks with two gating, candidate memory and memory cells (Table 1).

To analyse the average reward on the previous three trials across episodes for the data in Fig 2.7C, only averages were calculated from the fourth trial onwards to prevent any confounding with episode reversal effects. To avoid biasing the analysis, only episodes with a mixture of correct and incorrect responses were included.

Statistical analyses

Prior to statistical analysis, assumptions of normality were tested using the Kolmogorov-Smirnov and Shapiro-Wilk tests. If these tests indicated significant deviations from normality for at least one of the two distributions, a non-parametric test was used and the median was reported instead of the mean.

We used a regression analysis to determine whether units showed significant selectivity to features in the pro-/anti-saccade task (Figs 2.6A-C). We fitted a linear regression model with saccade type (pro-saccade or anti-saccade), cue location (left or right) and their interaction to the activity of units in three networks during the cue, memory delay or 'go' phases of the task. If an omnibus test for normality, Durbin-Watson or Jarque-Bera test, indicated significant heteroscedasticity, skewness or kurtosis (alpha of 0.05), a robust regression model was fitted using Huber's t function instead. We included a Bonferroni correction for multiple comparisons and applied an alpha of 0.05.

2.4.2 The relation between backpropagation-through-time and RECOLLECT

In this section, we will demonstrate that backpropagation-through-time is implemented by RECOLLECT with a combination of synaptic traces and tags.

Computing the gradient of $M_j(t)$, $k_j(t)$ and $C_j(t)$

The influence of the activity of memory unit j, $M_j(t)$, on the Q-value of the selected action s, $q_s(t)$, is (Fig 2.1):

$$\frac{\partial q_s(t)}{\partial M_j(t)} = W_{sj}^{FB}, \tag{M.2}$$

which is proportional to the amount of attentional feedback flowing from the winning action s to memory unit j (Roelfsema & van Ooyen, 2005). We can now compute the influence of the memory gate $k_j(t)$ on $q_s(t)$ based on equation 3:

$$\frac{\partial q_s(t)}{\partial k_j(t)} = \frac{\partial M_j(t)}{\partial k_j(t)} \frac{\partial q_s(t)}{\partial M_j(t)} = [M_j(t-1) - C_j(t)]W_{sj}^{FB}. \tag{M.3}$$

Furthermore, it follows from equation 3 that the influence of $C_j(t)$ on $q_s(t)$ depends on $k_j(t)$:

$$\frac{\partial q_s(t)}{\partial C_j(t)} = \frac{\partial M_j(t)}{\partial C_j(t)} \frac{\partial q_s(t)}{\partial M_j(t)} = [1 - k_j(t)]W_{sj}^{FB}. \tag{M.4}$$

Computing the gradient of W_{ij}^C using synaptic traces

We can now compute the instantaneous impact of connections $W_{ij}^C(t)$ on $q_s(t)$:

$$\frac{\partial q_s(t)}{\partial W_{ij}^C} = \frac{\partial C_j(t)}{\partial W_{ij}^C} \frac{\partial q_s(t)}{\partial C_j(t)} = x_i(t)\sigma'(Inp_j^C(t))[1 - k_j(t)]W_{sj}^{FB}, \tag{M.5}$$

where $\sigma'(Inp_j^C)$ is the derivative of the activation function. However, these connections have also had impact on the memory state $M_j(t)$ on all previous time steps according to equation (1). For example, connection W_{ij}^C had an influence on $C_j(t-1)$ which influenced $M_j(t-1)$ and thereby also $M_j(t)$. Although the notation is a bit ugly, for convenience let us write for this influence of W_{ij}^C on $t-1$ on $q_s(t)$:

$$\begin{gathered}\frac{\partial q_s(t)}{\partial W_{ij}^C(t-1)} = \frac{\partial C_j(t-1)}{\partial W_{ij}^C(t-1)} \frac{\partial M_j(t-1)}{\partial C_j(t-1)} \frac{\partial M_j(t)}{\partial M_j(t-1)} \frac{\partial q_s(t)}{\partial M_j(t)} = \\ x_i(t-1)\sigma'(Inp_j^C(t-1))[1 - k_j(t-1)]k_j(t)W_{sj}^{FB}.\end{gathered} \tag{M.6}$$

We can also compute this term for $t-2$:

$$\frac{\partial q_s(t)}{\partial W_{ij}^C(t-2)} = \frac{\partial C_j(t-2)}{\partial W_{ij}^C(t-2)} \frac{\partial M_j(t-2)}{\partial C_j(t-2)} \frac{\partial M_j(t-1)}{\partial M_j(t-2)} \frac{\partial M_j(t)}{\partial M_j(t-1)} \frac{\partial q_s(t)}{\partial M_j(t)} = x_i(t-2)\sigma'(Inp_j^C(t-2))[1-k_j(t-2)]k_j(t-1)k_j(t)W_{sj}^{FB}. \tag{M.7}$$

and, in general, for $t-i$:

$$\frac{\partial q_s(t)}{\partial W_{ij}^C(t-i)} = x_i(t-i)\sigma'(Inp_j^C(t-i))[1-k_j(t-i)]\Pi_{g=t-i+1}^t k_j(g)W_{sj}^{FB}. \tag{M.8}$$

Although this gradient may look complex, it is actually straightforward to store the information in a $trace_{ij}^C$ at the synapse and update it based on information that is locally available:

$$Trace_{ij}^C(0) = 0, \tag{M.9}$$

$$Trace_{ij}^C(t) = k_j(t)Trace_{ij}^C(t-1) + [1-k_j(t)]x_i(t)\sigma'(Inp_j^C(t)). \tag{M.10}$$

Importantly, this information can be made available locally at the synapse, assuming that the gating unit k_j is in the same cortical column as the memory unit M_j. Adding all the time steps, the total influence of W_{ij}^C on $q_s(t)$ becomes:

$$\frac{\partial q_s(t)}{W_{ij}^C} = Trace_{ij}^C W_{sj}^{FB}. \tag{M.11}$$

Computing the gradient of W_{ij}^k using synaptic traces

We can use equation (5) to compute the influence of the synapses W_{ij}^k that influence the memory gate $k_j(t)$ on $q_s(t)$. As before, we start with the *instantaneous* impact of connections $W_{ij}^k(t)$ on $q_s(t)$:

$$\frac{\partial q_s(t)}{\partial W_{ij}^k(t)} = \frac{\partial k_j(t)}{\partial W_{ij}^k} \frac{\partial q_s(t)}{\partial k_j(t)} = x_i(t)\sigma'(Inp_j^k(t))[M_j(t-1) - C_j(t)]W_{sj}^{FB}. \tag{M.12}$$

Let us now consider the influence of this synapse at $t-1$ on the $q_s(t)$:

$$\frac{\partial q_s(t)}{\partial W_{ij}^k(t-1)} = \frac{\partial k_j(t-1)}{\partial W_{ij}^k(t-1)} \frac{\partial M_j(t-1)}{\partial k_j(t-1)} \frac{\partial M_j(t)}{\partial M_j(t-1)} \frac{\partial q_s(t)}{\partial M_j(t)} = x_i(t-1)\sigma'(Inp_j^k(t-1))[M_j(t-2) - C_j(t-1)]k_j(t)W_{sj}^{FB}, \tag{M.13}$$

and at $t-2$:

$$\frac{\partial q_s(t)}{\partial W^k_{ij}(t-2)} = \frac{\partial k_j(t-2)}{\partial W^k_{ij}(t-2)} \frac{\partial M_j(t-2)}{\partial k_j(t-2)} \frac{\partial M_j(t-1)}{\partial M_j(t-2)} \frac{\partial M_j(t)}{\partial M_j(t-1)} \frac{\partial q_s(t)}{\partial M_j(t)} = \\ x_i(t-2)\sigma'(Inp^k_j(t-2))[M_j(t-3) - C_j(t-2)]k_j(t-1)k_j(t)W^{FB}_{sj}. \tag{M.14}$$

In general, for $t-i$:

$$\frac{\partial q_s(t)}{\partial W^k_{ij}(t-i)} = x_i(t-i)\sigma'(Inp^k_j(t-i))[M_j(t-i-1) - C_j(t-i)]\Pi^t_{g=t-i+1}k_j(g)W^{FB}_{sj}. \tag{M.15}$$

This gradient can also be stored in the form of a $trace^k_{ij}$ at the synapse and updated based on information that is locally available:

$$Trace^k_{ij}(0) = 0, \tag{M.16}$$

$$Trace^k_{ij}(t) = k_j(t)Trace^k_{ij}(t-1) + [M_j(t-1) - C_j(t)]x_i(t)\sigma'(Inp^k_j(t)). \tag{M.17}$$

Again, this information is available at the synapse if we assume that the difference in activity between $M_j(t)$ and $C_j(t)$ is computed in the same cortical column as $k_j(t)$, which is common in models of predictive coding. When adding all the time steps, the total influence of W^k_{ij} on $q_s(t)$ becomes:

$$\frac{\partial q_s(t)}{\partial W^k_{ij}} = Trace^k_{ij}W^{FB}_{sj}. \tag{M.18}$$

Tags and traces

RECOLLECT distinguishes between traces and tags (see also Rombouts et al., 2015). Whereas the traces represent the contribution of a synapse to the activity of the memory unit, the tags represent the influence of the synapse on the Q-value of the chosen action. The tag depends on the trace as well as on the amount of attentional feedback that arrives at the memory unit through the feedback connection from the chosen action (see equations 11 and 15).

The tags are used to implement the SARSA(λ) algorithm. If λ is larger than zero, the synapses that contributed to previous actions are also updated, while taking the temporal discount factor γ into account. This is an advantage of RECOLLECT and AuGMEnT (Rombouts et al., 2015) over e-prop (Bellec et al., 2020), which uses a similar approach to approximating backpropagation-through-time. The resulting combination of tags and traces, can be shown to be equivalent to gradient descent through backpropagation-through-time on the temporal difference error in the absence of recurrent connections (see Rombouts et al., 2015 for more detail), and to approximate backpropagation-through-time when recurrent weights are included.

2.4.3 AuGMEnT architecture

This section explains the architecture of the AuGMEnT model (Rombouts et al., 2015) and how it differs from RECOLLECT.

AuGMEnT trains networks with three layers: an input layer, an association layer and a Q-value layer. The input layer consists of instantaneous units and transient units. The instantaneous units encode stimuli in the current timestep, and transient units signal changes in the stimuli. On-units become active if a stimulus appears and off-units if it disappears. The association layer also contains regular units and memory units, which exclusively receive information from instantaneous units and transient units, respectively. The activity of regular units depends on the input received at the current timestep, whereas memory units maintain information about stimuli presented during previous timesteps. Memory units of AuGMEnT lack a gating mechanism to block new sensory information or remove previous memory content. Consequently, memory content in AuGMEnT has to be erased at the end of a simulated trial because the learning rule cannot learn to forget the information in memory from the previous trial when a new trial starts. Both instantaneous and memory units project to Q-value units in the output layer of AuGMEnT, just as in RECOLLECT (see equation 4). The learning rule in AuGMEnT is similar to that of RECOLLECT (see section 'learning rule' of the Results).

Chapter 3

Biologically plausible reinforcement learning with deep gated memory networks

Alexandra R. van den Berg[1,2], Pieter R. Roelfsema[2,3,4,5], Sander M. Bohté[1,6]

[1]*Machine Learning Group, Centrum Wiskunde & Informatica, Amsterdam, the Netherlands*

[2]*Department of Vision & Cognition, Netherlands Institute for Neuroscience, Amsterdam, the Netherlands*

[3]*Department of Integrative Neurophysiology, Center for Neurogenomics and Cognitive Research, Vrije Universiteit Amsterdam, Amsterdam, the Netherlands*

[4]*Department of Neurosurgery, Academic Medical Center, Amsterdam, the Netherlands*

[5]*Laboratory of Visual Brain Therapy, Sorbonne Université, Institut National de la Santé et de la Recherche Médicale, Centre National de la Recherche Scientifique, Institut de la Vision, Paris, France*

[6]*Swammerdam Institute of Life Sciences, University of Amsterdam, Amsterdam, the Netherlands*

Abstract Human memory is remarkably flexible. Different characteristics of the same object or occurrence can be recalled in varying levels of detail, depending on what is currently relevant to accomplish one's goals. How these representations can be represented and maintained online is only partially understood. Existing working memory models lack depth and/or are trained with biologically implausible learning rules, which makes it difficult to draw conclusions about the brain. We contribute to this understanding by developing a new deep gated memory architecture called Stackollect, which we train with a biologically plausible learning rule. We introduce a task that requires different memory representations depending on goal requirements, because existing paradigms do not lend themselves for animal studies, or have simple visual features that do not need deeper perceptual processing. The networks need to assess whether two consecutive stimuli have the same shape or spatial location and we examine the emerging representations. We show that Stackollect successfully trains networks to represent both location- and shape-related information across memory layers to solve the matching tasks, and that the task variant mainly influences the representations of the output layer.

3.1 Introduction

Humans possess remarkably flexible memory capacity. For instance, if one were to purchase apples at a grocery store to bake an apple pie, different types of knowledge have to be drawn upon and maintained online. One has to remember *where* apples can be found within the store, and *which* types of apples are appropriate for a pie. How does the brain represent these different types of information in working memory?

Working memory is the ability to maintain and manipulate information over a period of time, even when this information is no longer accessible in the sensory environment (Baddeley, 2010; Bays et al., 2024; Stroud et al., 2024). There are several theories about the neurobiology of working memory (Baddeley, 2010; Lundqvist et al., 2018; Stokes, 2015). One popular framework postulates that it emerges from persistent activity (Christophel et al., 2017; Curtis & Sprague, 2021; D'Esposito & Postle, 2015; Fuster, 1997). Persistent activity has been documented during various working memory paradigms, in which neurons stay active to maintain task-relevant information across temporal delays (Curtis & Sprague, 2021). Working memory maintenance is unlikely to be constrained to neurons within a specific region, but rather involves multiple connected regions, such as the prefrontal cortex, thalamus, basal ganglia and even the cerebellum (Barbas et al., 2018; Bolkan et al., 2017; Brissenden et al., 2021; Dehghani & Wimmer, 2019; Wilhelm et al., 2023). Moreover, selectively inactivating specific regions can disrupt these functional connections and affect working memory maintenance (Gao et al., 2018; Li et al., 2016; Voitov & Mrsic-Flogel, 2022), and subpopulations within these brain areas have been suggested to be differentially involved in working memory (Bae et al., 2021; Sonneborn et al., 2024). Can persistent activity through layers be modelled in a biologically plausible network? And does the type and complexity of task-relevant features influence the memory representations in the different layers?

There are numerous studies that use neural networks to model working memory (e.g. Beck et al., 2024; Costa et al., 2017; Hedayati et al., 2022; Lei et al., 2024; Ursino et al., 2023). However, there is a lack of studies on how activity is selectively sustained or forgotten across *multiple* memory layers, particularly in models that are inspired from biology and trained using learning rules that use only information that is local at the level of the synapse (Kruijne et al., 2021; Rombouts et al., 2015; van den Berg et al., 2024), compared to the standard error-backpropagation algorithm (Lillicrap & Santoro, 2019). Additionally, there are few tasks for which the type of memory representation required can be flexibly altered to allow a systematic comparison of how working memories can emerge across different layers in a network for different features, especially when balancing the need for simplicity and perceptual richness. It would be beneficial if the task can performed by animals such as rodents for the validation of computational models, which is difficult with tasks akin to Atari games (Skinner & Walmsley, 2019) or other complex structures (Lei et al., 2024). Furthermore, it is desirable if the sensory stimuli are rich enough to require the emergence of shape representations, unlike more traditional memory paradigms such as the 12-AX task (Frank et al., 2001) and previous delayed match-to-sample tasks (Dang et al., 2021).

Here, we therefore developed an architecture with multiple layers of memory units that can each flexibly remember or forget information and is updated using a biologically plausible learning rule – Stackollect. We also designed a novel delayed match-to-sample task with different rules, either based on shape or location matching, used in experimental research with monkeys (e.g. Dang et al., 2021; Meyer et al., 2011). Specifically, networks had to match the location or identity of two handwritten digits from the MNIST dataset (Lecun et al., 1998). We show that networks trained with Stackollect successfully learned both tasks and we show how working memory representations are distributed across the different layers of the biologically plausible neural network.

3.2 Method

3.2.1 Architecture

'Stackollect' extends the RECOLLECT (van den Berg et al., 2024) model in which multiple Light-GRU modules (Ravanelli et al., 2018) are 'stacked' on top of one another to create a deeper gated-memory architecture. Stacking memory layers can improve performance and has been suggested to allow learning over multiple timescales (Choi et al., 2019; Graves et al., 2013; Hermans & Schrauwen, 2013; Sak et al., 2014). In theory, the model generalises to any number of layers, but we here consider networks with an input layer, two Light-GRU layers and an output layer (see Figure 3.1).

Each Light-GRU layer has memory units (M; grey circles in Figure 3.1) that store information over time. New input enters the memory units through the candidate memory units (C; green circles). However, how much of this new input is used to update the memory units, depends on the state of the gating units (k; red circles). Hence, memory units of each layer can flexibly forget or retain information, depending on task requirements.

The first hidden layer receives sensory information ($x_i(t)$) from the environment at time t, which is processed by the candidate memory units and gating units of that layer (i.e. C^1 and k^1 for layer 1):

$$C^1_j = \sigma(\sum_i U^{C^1}_{ij} x_i(t) + b^{C^1}_j).$$

$$k^1_j = \sigma(\sum_i U^{k^1}_{ij} x_i(t) + b^{k^1}_j).$$

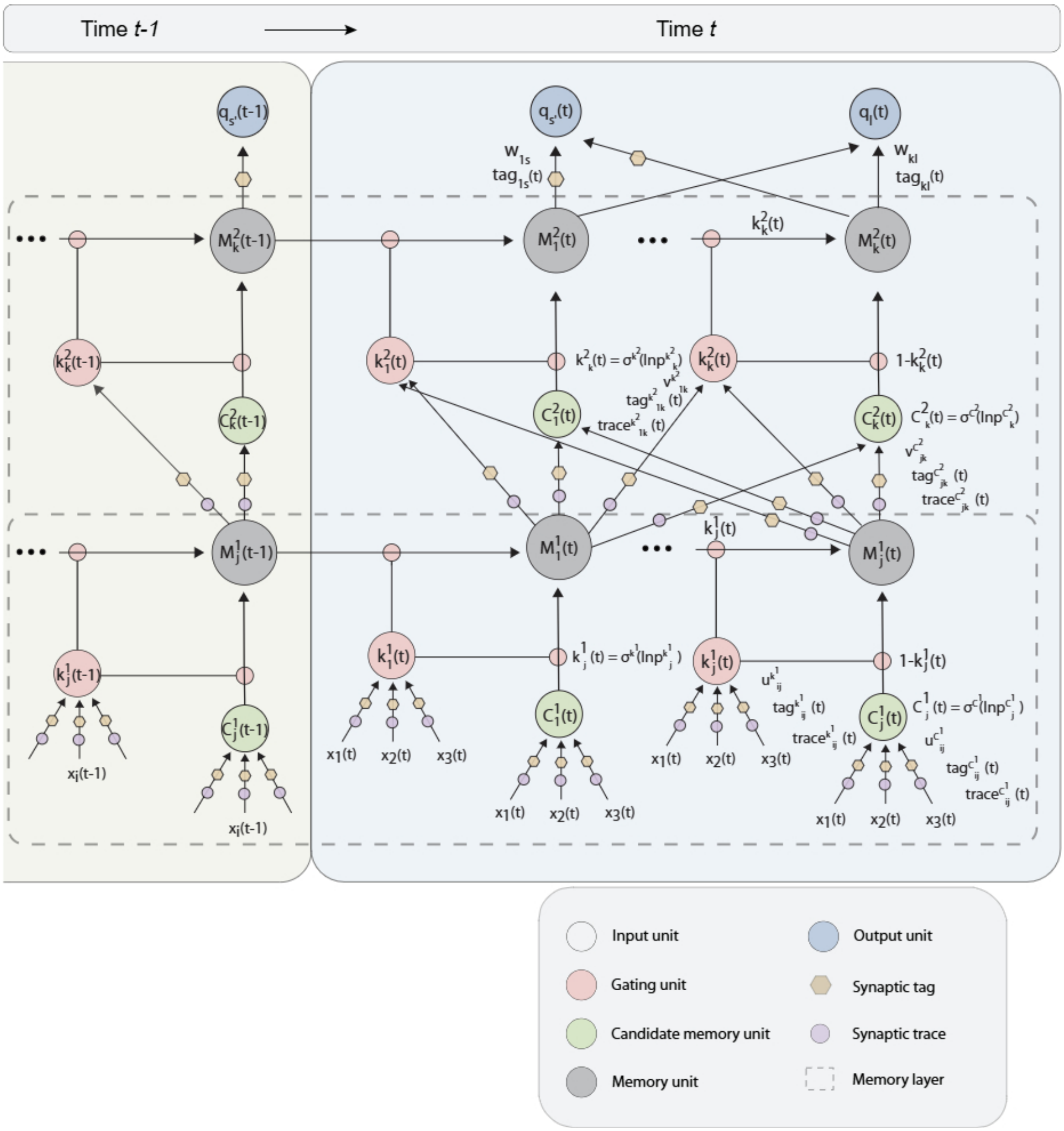

Figure 3.1: Stackollect architecture. Stackollect consists of an input layer followed by two (or more) layers of Light-GRU units and an output layer. Each Light-GRU unit contains a memory unit (M; grey circles), which is updated using new sensory information from candidate memory units (C; green circles), and old memory content that is filtered by gating units (k; red circles). Synaptic traces and tags are used for local credit assignment.

The activation concerns a weighted sum of their input weights (U_{ij}) based on the input $x_i(t)$, combined with a bias term (value of 1). A sigmoidal activation function (σ) determines their final value, where ρ represents the slope of the function, which was set to 2.5:

$$\sigma(input_j(t)) = \frac{1}{(1 + exp(-(\rho \cdot input_j(t))))}.$$

The gating unit is responsible for determining which information from the candidate memory units is allowed to enter the memory unit, as well as which information that is currently maintained in the memory unit should be forgotten:

$$M_j^1(t) = ReLU(k_j^1(t) \odot M_j^1(t-1) + (1 - k_j^1(t)) \odot C_j^1(t),$$

where $\odot$ signifies element-wise multiplication. The gating unit learns to balance old and new information. If the activity of gating unit $k_j^1(t)$ is close to 1, the activity ($M_j^1(t)$) is the same as ($M_j^1(t-1)$) and the memory is protected. If $k_j^1(t)$ is close to zero, new information (from $C_j^1(t)$) will be registered. The resulting memory content then passes through a rectified linear unit (ReLU) activation function that outputs zero for negative values, which is helpful when training deeper architectures (Krizhevsky et al., 2012). The workings of the second memory layer are the same as those of the first layer, but with the memory content of the previous layer as their input (e.g. $M_j^1(t)$ forms the input to the second memory layer).

Finally, Q-value units (q_l) in the output layer computed the expected reward of the actions that could be chosen by the network based on the weighted sum of their weights and the memory content of the second hidden layer (M_k^2):

$$q_l(t) = \sum_k w_{kl}^q M_k^2(t) + b_l^q.$$

An epsilon-greedy strategy determines the winning action s. The network selects the action with the highest Q-value with probability $1 - \epsilon$, or a random, exploratory action with probability ϵ. Several brain regions have been observed to contain neurons that seem to encode action values, such as in the basal ganglia (Hikosaka et al., 2014; Ito & Doya, 2009), midbrain (Morris et al., 2006) and frontal cortex (Cai & Padoa-Schioppa, 2014; Padoa-Schioppa & Assad, 2006; Rushworth et al., 2011).

Lastly, the model includes feedback connections from output units to the memory units of both layers (w'_{sk} and w'_{sj}), which influence the plasticity of candidate memory units and gating units within the memory layers (see below; 'synaptic tags and synaptic traces').

3.2.2 Learning rule

The learning rule of Stackollect largely follows RECOLLECT (van den Berg et al., 2024), but we implemented a truncation mechanism that simplifies the temporal credit assignment problem (see 'Truncated Stackollect') and build upon the BrainProp learning rule (Pozzi et al., 2020) to extend it to deeper networks. The result is a local approximation of backpropagation-through-time using a combination of synaptic tags and truncated synaptic traces to assign spatial and temporal credit within Light-GRU modules, together with a feedback mechanism to propagate activity across different layers of the network. We will here first derive the RECOLLECT learning rule for Stackollect, then show how truncation assists in credit assignment for deeper models and conclude with how the BrainProp mechanisms can work in deeper networks to guide the final learning rule.

As in RECOLLECT, plasticity in Stackollect is driven by trial-and-error learning using the reward prediction error (δ; RPE), which is broadcast as a global neuromodulatory signal to all synapses in the network:

$$\delta(t) = r(t) + \gamma q_s(t) - q_a(t-1).$$

The RPE compares the expected outcome from the previous action $q_a(t-1)$ to the sum of the actual reward received $r(t)$ and the Q-value of the winning action $q_s(t)$discounted by the reward discount factor γ. A positive RPE indicates that the outcome was better than expected, whereas a negative RPE signals that the outcome was disappointing.

To determine which synapses to update (spatial credit assignment) and how they should be updated over time (temporal credit assignment), Stackollect uses the same combination of tags and traces as in RECOLLECT but extends this process to deeper gated memory architectures.

3.2.3 Synaptic tags and synaptic traces

Synaptic tags depend on an attentional feedback signal that enables plasticity in the synapses that were important for the selection of actions that triggered an RPE. Feedback connections from the winning output unit provide feedback to all synapses that provided the input that enabled it to win, also in the lower layers. These feedback connections follow the same learning rule as will be described below for the feedforward connections. As a result, the feedforward and feedback connections become proportional to one another over the course of learning.

In addition to the tags, there are also synaptic traces. The role of the synaptic traces is to enable a local form of bookkeeping of activity at synapses of memory units over time. The synaptic traces encode the degree to which a change in the synapse would alter the activity of the memory unit. Traces do not directly influence plasticity, but only via the formation of tags when units receive top-down input. The resulting tags interact with the RPE to control synaptic plasticity.

Together, tags and traces can inform synapses locally about all information required for network updates. We will explain the equations governing Stackollect below.

Stackollect aims to minimise the loss E, as informed by the RPE (see 'Learning Rule'), to reliably choose actions yielding the highest Q-value:

$$E(q_a(t-1)) = \frac{1}{2}([r(t) + \gamma q_{a'}(t)] - q_a(t-1))^2$$

First, the immediate impact of each of the connections w_{ks} to the output layer on $q_s(t)$ can be calculated:

$$\frac{\partial E}{\partial w_{ks}} = -\delta(t)\frac{\partial q_s(t)}{\partial w_{ks}},$$

$$\frac{\partial E}{\partial w_{ks}} = -\delta(t)M_k^2(t),$$

Where M_k^2 refers to the memory activity in the Light-GRU layer that provides input to the output layer through connections w_{ks} to the winning output unit s.

This gradient is captured by the synaptic tags:

$$Tag_{ks}(t) = \lambda\gamma Tag_{kl}(t-1) + M_k^2(t), \tag{3.1}$$

$$Tag_{kl}(t) = \lambda\gamma Tag_{kl}(t-1); l \neq s, \tag{3.2}$$

Hence, only the tags on the connections to the winning output unit are increased and the others decay according to the tag decay rate (λ) and the reward discount factor (γ). As a result, actions taken in the past can still influence network plasticity in the future, but their influence diminishes over time (Sutton, 1988). The tags interact with the RPE $\delta(t)$ to determine the synaptic update:

$$\Delta w_{kl} = \beta\delta Tag_{kl}, \tag{3.3}$$

where β is the learning rate.

We will now derive the weight update for gating units in the second hidden layer, followed by that for the candidate memory units.

The immediate impact of the weights from the first memory layer to gating units of the second memory layer ($v_{jk}^{k^2}$) on $q_s(t)$ is:

$$\frac{\partial q_s(t)}{\partial v_{jk}^{k^2}(t)} = \frac{\partial q_s(t)}{\partial M_k^2(t)}\frac{\partial M_k^2(t)}{\partial k_k^2(t)}\frac{\partial k_k^2(t)}{\partial v_{jk}^{k^2}(t)},$$

$$\frac{\partial q_s(t)}{\partial v_{jk}^{k^2}(t)} = w_{ks}ReLU'(M_k^2(t))[M_k^2(t-1) - C_k^2(t)]\sigma'(Inp_k^{k^2}(t))M_j^1(t),$$

where σ' stands for the derivative of the sigmoid activation function, $ReLU'$ for the derivative of the $ReLU$ activation function, k^2 and C^2 refer to gating and candidate memory units in the second memory layer, respectively, $Inp_k^{k^2}(t)$ the input to the gating unit in the second memory layer (i.e. the memory content of the first hidden layer), and M^1 to the memory unit in the first hidden layer.

However, there is also an influence of this synapse on $q_s(t)$ because it influenced the memory unit M_k^2 at t-1:

$$\frac{\partial q_s(t)}{\partial v_{jk}^{k^2}(t-1)} = \frac{\partial q_s(t)}{\partial M_k^2(t)} \frac{\partial M_k^2(t)}{\partial M_k^2(t-1)} \frac{\partial M_k^2(t-1)}{\partial k_k^2(t-1)} \frac{\partial k_k^2(t-1)}{\partial v_{jk}^{k^2}(t-1)},$$

$$\frac{\partial q_s(t)}{\partial v_{jk}^{k^2}(t-1)} = w_{ks} ReLU'(M_k^2(t)) k_k^2(t) ReLU'(M_k^2(t-1))$$
$$[M_k^2(t-2) - C_k^2(t-1)]\sigma'(Inp_k^{k^2}(t-1)) M_j^1(t-1).$$

Similarly, the influence at $t-2$ was:

$$\frac{\partial q_s(t)}{\partial v_{jk}^{k^2}(t-2)} = \frac{\partial q_s(t)}{\partial M_k^2(t)} \frac{\partial M_k^2(t)}{\partial M_k^2(t-1)} \frac{\partial M_k^2(t-1)}{\partial M_k^2(t-2)} \frac{\partial M_k^2(t-2)}{\partial k_k^2(t-2)} \frac{\partial k_k^2(t-2)}{\partial v_{jk}^{k^2}(t-2)},$$

$$\frac{\partial q_s(t)}{\partial v_{jk}^{k^2}(t-2)} = w_{ks} ReLU'(M_k^2(t)) k_k^2(t) ReLU'(M_k^2(t-1)) k_k^2(t-1) ReLU'(M_k^2(t-2))$$
$$[M_k^2(t-3) - C_k^2(t-2)]\sigma'(Inp_k^{k^2}(t-2)) M_j^1(t-2).$$

The influence of the synapse based on the entire history of activations is captured by the tags and traces (as in RECOLLECT):

$$Trace_{jk}^{k^2}(0) = 0,$$
$$Trace_{jk}^{k^2}(t) = k_k^2(t) Trace_{jk}^{k^2}(t-1) + ReLU'(M_k^2(t))[M_k^2(t-1) - C_k^2(t)] M_j^1(t) \sigma'(Inp_k^{k^2}(t)),$$
$$Tag_{jk}^{k^2}(t) = \lambda\gamma Tag_{jk}^{k^2}(t-1) + Trace_{jk}^{k^2}(t) w_{ks},$$
$$\Delta v_{jk}^{k^2} = \beta\delta(t) Tag_{jk}^{k^2}.$$

In short, the synaptic traces of gating units in the second layer ($Trace_{jk}^{k^2}$) are initialised at 0 and decay by an amount that reflects the state of the memory gate at every timestep, while the new input at the current timestep adds to the traces. To determine the tag, the trace is multiplied with the output weights (w_{ks}), which reflects the influence on the output units and is available through the feedback connections from output units to memory units (w'_{sk}).

We will now derive the learning rule for the connections to the candidate memory units in layer 2, using a similar approach:

$$Trace_{jk}^{C^2}(0) = 0,$$
$$Trace_{jk}^{C^2}(t) = k_k^2(t) Trace_{jk}^{C^2}(t-1) + ReLU'(M_k^2(t))[1 - k_k^2(t)] M_j^1(t) \sigma'(Inp_k^{C^2}(t)),$$

$$Tag^{C^2}_{jk}(t) = \lambda\gamma Tag^{C^2}_{jk}(t-1) + Trace^{C^2}_{jk}(t) w_{ks},$$
$$\Delta v^{C^2}_{jk} = \beta\delta(t) Tag^{C^2}_{jk}.$$

We will now discuss the update of the lower-level gate and candidate memory units, which is a more complex.

3.2.4 Credit assignment to lower layers

Here, we will derive the updates for the lower-level gate and candidate memory units. First, we determine the immediate impact on the weights from input units to the gating units in the first memory layer $u^{k^1}_{ij}$:

$$\frac{q_s(t)}{u^{k^1}_{ij}(t)} = \sum_k \left[\frac{q_s(t)}{M^2_k(t)} \left[\frac{M^2_k(t)}{k^2_k(t)} \frac{k^2_k(t)}{M^1_j(t)} + \frac{M^2_k(t)}{C^2_k(t)} \frac{C^2_k(t)}{M^1_j(t)} \right] \right] \frac{M^1_j(t)}{k^1_j(t)} \frac{k^1_j(t)}{u^{k^1}_{ij}(t)},$$

$$\frac{q_s(t)}{u^{k^1}_{ij}(t)} = \sum_k \Bigg[w_{ks} \Bigg[ReLU'(M^2_k(t))[M^2_k(t-1) - C^2_k(t)]\sigma'(Inp^{k^2}_k(t))v^{k^2}_{jk} +$$
$$ReLU'(M^2_k(t))[1 - k^2_k(t)]\sigma'(Inp^{C^2}_k(t))v^{C^2}_{jk} \Bigg] \Bigg] ReLU'(M^1_j(t))[M^1_j(t-1) - C^1_j(t)]$$
$$\sigma'(Inp^{k^1}_k(t))x_i(t).$$

At $t-1$, the complexity grows because there are two layers with memory, which can both maintain information from previous time steps. There is now not only a path from $M^2_k(t)$ to $M^2_k(t-1)$ and $M^1_j(t-1)$, but also a path from $M^2_k(t)$ to $M^1_j(t)$ to $M^1_j(t-1)$. As we show in the appendix, assigning credit over time and across layers becomes prohibitive in terms of complexity, since each step backwards in time creates new pathways along which credit must be assigned. Moreover, these pathways require non-local information transfer. This is unlikely to be biologically plausible, especially for deeper networks.

3.2.5 Truncated Stackollect

To alleviate this problem, a truncation can be applied wherein traces are only used within a single layer, thereby disregarding these novel paths emerging at each consecutive timestep. This truncation influences the lower memory layer. As a result, the update in this layer will be an approximation rather than an exact solution to BPTT, but will be more realistic from a biological perspective and less computationally expensive. The path through current state t of the higher memory layer is considered for the plasticity of connections to the lower memory layer, but the paths based on the activity in the higher memory layer on previous time steps $< t$ are disregarded. Non-biologically inspired RNNs are often similarly truncated to reduce training cost (Sutskever, 2013; Williams & Peng, 1990). The truncated equations are simpler:

$$\frac{\partial q_s(t)}{\partial u_{ij}^{k^1}} = Trace_{ij}^{k^1} fb_j^{k^1}, \tag{3.4}$$

$$\frac{\partial q_s(t)}{\partial u_{ij}^{C^1}} = Trace_{ij}^{C^1} fb_j^{C^1}. \tag{3.5}$$

Where $fb_j^{k^1}$ and $fb_j^{C^1}$ denote the amount of feedback arriving at these gating and candidate memory units. How this feedback is calculated is covered in the next section.

3.2.6 The feedback pathway of truncated Stackollect

According to equations 4-5, the weight updates in the lower layers depend on feedback activity from higher layers, which can be resolved by a Brainprop-like mechanism (Pozzi et al., 2020), using a feedback network to perform credit assignment to the lower layers. The required feedback signal comes from the layer directly above the current layer $(d + 1)$. For each unit, the weight update of any weight w_{ij}^d in layer d can be computed as the product of the reward prediction error δ, and the feedforward input from unit j the layer below (y_j^d) and that tag, which reflects the feedback arriving at unit i the layer above (φ_i^{d+1}), and the derivative of the activation function of the layer above (g_i^{d+1}):

$$\Delta w_{ij}^d = \beta\delta\varphi_i^{d+1} g_i^{d+1} y_j^d,$$

Therefore, $fb_j^{k^1}$ and $fb_j^{C^1}$ represent a combination of φ^{d+1} and g^{d+1}:

$$fb_j^{k^1} = \varphi_i^{d+1} g_i^{d+1},$$
$$fb_j^{C^1} = \varphi_i^{d+1} g_i^{d+1}$$

The attentional feedback at each layer (φ_i^d) depends on the strength of the feedback weights, the attentional feedback at the level above and the derivative of the activation function of the level above:

$$\varphi_i^d = \sum_j w_{ij}^{FB,d} \varphi_j^{d+1} g_j^{d+1}.$$

The implementation of this mechanism in Stackollect will be described for the two-layer network below; the same approach can be extended to networks with deeper layers as described in Pozzi et al. (2020).

For the output layer N, the amount of attentional feedback at each unit l is 1 if l is the winning unit and 0 if l is not the winning unit:

$$\varphi_l^N = \begin{cases} 1 & \text{if } l = s \\ 0 & \text{if } l \neq s \end{cases}.$$

The output layer in Stackollect is linear, so the weight update for the output weights w_{kl} becomes:

$$\Delta w_{kl} = \delta \varphi_l M_k^2,$$

which when accounting for tags is the same as the previously formulated output weight update (see equations 1-3):

$$Tag_{ks}(t) = \lambda\gamma Tag_{kl}(t-1) + M_k^2(t),$$

$$Tag_{kl}(t) = \lambda\gamma Tag_{kl}(t-1); l \neq s,$$

$$\Delta w_{kl} = \beta\delta(t) Tag_{kl}.$$

For the weights in the layer below the output layer the feedback activity is:

$$\varphi_k = \sum_l w_{kl}^{FB} \varphi_l,$$

which is equal to:

$$\varphi_k = w_{ks}^{FB}.$$

This would lead to the following update within this layer:

$$\Delta v_{jk}^{k^2} = \delta \varphi_k g_k y_j,$$

For the gating units this yields the following update:

$$\Delta v_{jk}^{k^2} = \delta w_{ks}^{FB} ReLU'(M_k^2) Trace_{jk}^{k^2}(t).$$

Which, accounting for the tags, leads to the following weight update:

$$Tag_{jk}^{k^2}(t) = \lambda\gamma Tag_{jk}^{k^2}(t-1) + w_{ks}^{FB} ReLU'(M_k^2) Trace_{jk}^{k^2}(t).$$

$$\Delta v_{jk}^{k^2} = \beta\delta(t) Tag_{jk}^{k^2}.$$

Similarly, the resulting weight and tag update for the upper-level candidate memory unit become:

$$Tag_{jk}^{C^2}(t) = \lambda\gamma Tag_{jk}^{C^2}(t-1) + w_{ks}^{FB} ReLU'(M_k^2) Trace_{jk}^{C^2}(t).$$

$$\Delta v_{jk}^{C^2} = \beta\delta(t) Tag_{jk}^{C^2}.$$

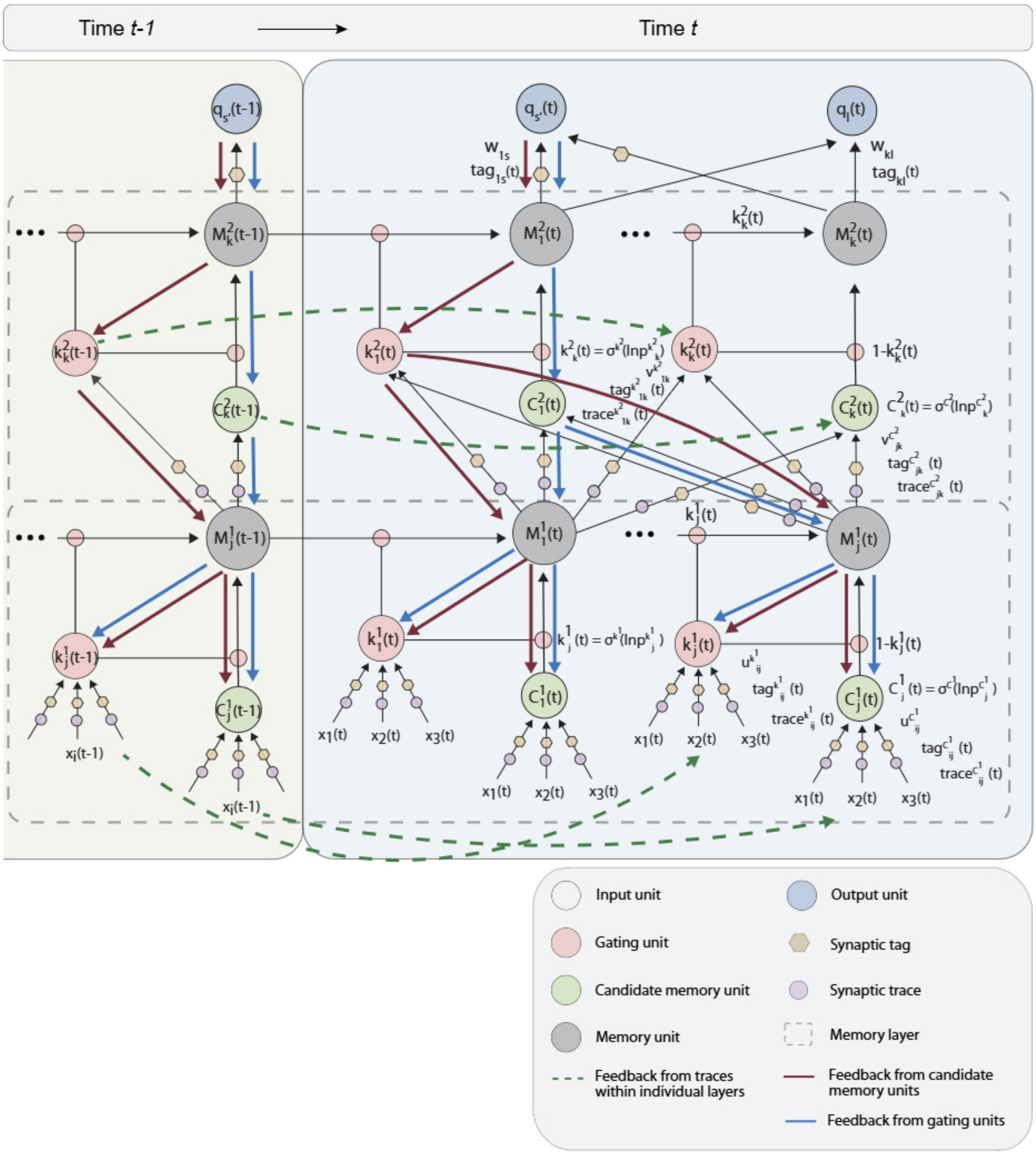

Figure 3.2: Feedback pathways in Stackollect laid out for a single timestep (blue background) with influence from the previous timestep indicated (green background). Truncated synaptic traces (dashed green lines) are used for temporal credit assignment within memory layers. As in BrainProp (Pozzi et al., 2020) feedback from output units to candidate memory units and gating units (blue and red lines, respectively) gate the plasticity of feedforward connections.

The updates of weight to the first layer are more complex because the feedback comes from two sources: the candidate memory units (blue path in Figure 3.2) and the gating units (red path in Figure 3.2), which are combined:

$$\varphi_j^{k^1} = \sum_k w_{jk}^{FB,k^2} \varphi_k \left[\sigma'(Inp_k^{k^2}(t))\right],$$

$$\varphi_j^{C^1} = \sum_k w_{jk}^{FB,C^2} \varphi_k \left[\sigma'(Inp_k^{C^2}(t))\right],$$

$$\varphi_j = \varphi_j^{k^1} + \varphi_j^{C^1}.$$

This results in the following weight update for units in this layer:

$$\Delta u_{ij} = \delta \varphi_j^{N-2} g_j^{N-2} y_i^{N-3},$$

We now write this out for input to the gating units in the first memory layer:

$$\Delta u_{ij}^{k^1} = \delta \varphi_j ReLU'(M_j^1) Trace_{ij}^{k^1}(t),$$

and for the candidate memory units:

$$\Delta u_{ij}^{C^1} = \delta \varphi_j ReLU'(M_j^1) Trace_{ij}^{C^1}(t).$$

The computation can be simplified by using tags:

$$Tag_{ij}^{k^1} = \lambda\gamma Tag_{ij}^{k^1}(t-1) + \varphi_j ReLU'(M_j^1) Trace_{ij}^{k^1}(t).$$

$$\Delta u_{ij}^{k^1} = \beta\delta(t) Tag_{ij}^{k^1}.$$

$$Tag_{ij}^{C^1} = \lambda\gamma Tag_{ij}^{C^1}(t-1) + \varphi_j ReLU'(M_j^1) Trace_{ij}^{C^1}(t).$$

$$\Delta u_{ij}^{C^1} = \beta\delta(t) Tag_{ij}^{C^1}.$$

3.2.7 Task

We tested Stackollect on delayed-match-to-sample tasks (Broadway, 2022; Daniel et al., 2016). The model was trained to match digits across two frames separated by a delay. The task started with a blank screen (33x33 pixels), after which a fixation mark appeared in the middle of the screen. If the model did not acquire fixation within 10 timesteps, the trial was aborted. If the model did fixate, it received a fixation reward of 0.2 and one of the digits (0-9, resized to 16x16 pixels) sampled from the MNIST (Lecun et al., 1998) dataset, containing of 60.000 training images and 10.000 test images, was shown in one of the four corners of the screen. The stimulus then disappeared, and another digit was shown after a delay (2 timesteps). Now the model had to make a decision by making a saccade to the left for a match or to the right for a mismatch. There were two versions of the task. In the position matching task, the goal was to report whether the positions of the two digits on the grid matched and the model could ignore the identity of the digit. In the identity matching task, the model indicated whether the digits were the same, irrespective of their location. If the model made a saccade in the correct direction (left for matches, right for mismatches), it received a reward of 1.5. If the model continued fixating without making a saccade for 8 timesteps, it received no reward and the trial was aborted. Therefore, the model either had to remember the identity or position of the digit.

We equalised the number of matches and mismatches across the training datasets of the two tasks (50/50 ratio) and presented all digit/position combinations equally often. We placed no additional constraints on the precise digit/position allocations for the training dataset since its size was relatively large (200.000 images). However, because the datasets used to assess convergence (900 and 1600 images for identity and position, respectively) and for evaluation (3200 images for both conditions) were smaller, we enforced enough presentations per feature or feature combination. When we tested for convergence in the identity matching task, half of the dataset consisted of matches and the other half of mismatches.

3.2.8 Pretraining

Before we trained the Stackollect models in the identity matching task, we pretrained them to classify individual MNIST digits after a delay. The structure of the pretraining task followed that of the final matching task, with a few minor differences. Firstly, fixation was not enforced. i.e. the trial was not terminated when the fixation was not acquired, but it was encouraged using a fixation reward (0.2). Secondly, no additional digit appeared during the 'go' timestep. Instead, the models reported the category of the digit. The output layer now consisted of 11 units (one unit for the fixation action and 10 to represent the different MNIST digits). To encourage the discovery of new classes during the pretraining phase, the learning rate was increased by a factor of 2.5 when an unexpected reward was received (RPE > 0) after an exploratory action. This was meant to emulate a sense of novelty and promote learning following positive RPEs. Previous studies have indeed found indications that positive and negative RPEs are encoded asymmetrically in terms of strength and that this differs across brain regions (Hoy et al., 2023) and similar increases in learning rate for novel stimuli have been used previously (Kakade & Dayan, 2002). This exploration boost was removed during the final matching training and the output layer was re-initialised with 3 units (fixation, match or mismatch).

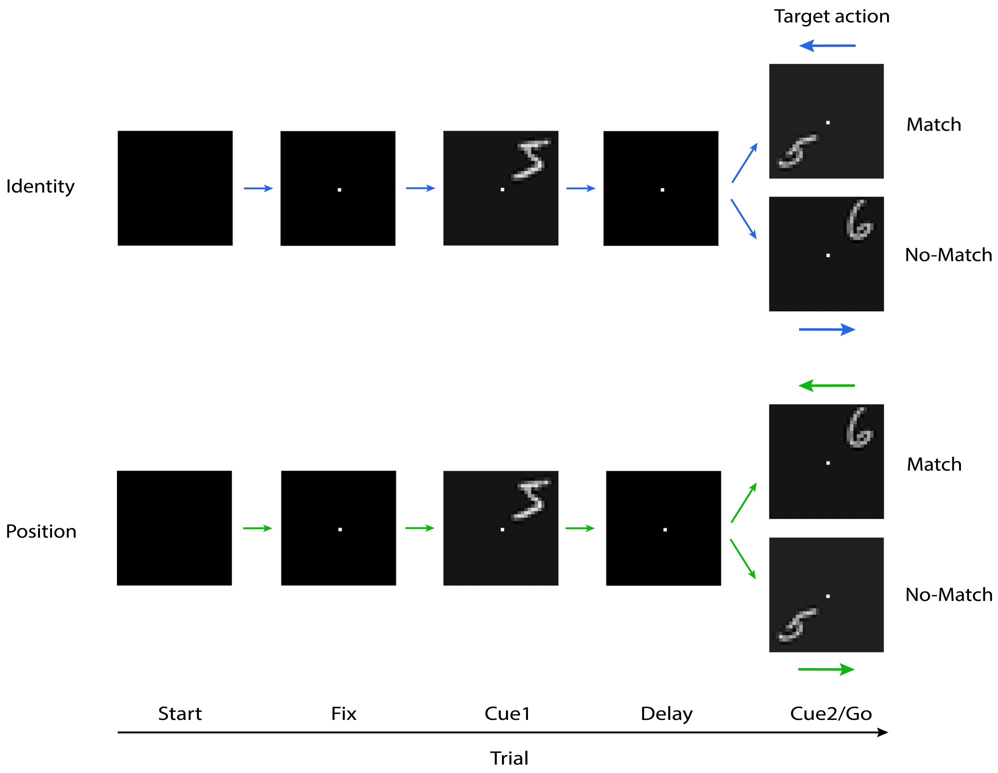

Figure 3.3: The delayed-match-to-MNIST task for the identity and position matching condition.

We used a curriculum that gradually introduced each digit to the network to balance new category learning and memory of the known digits. There were two phases for the learning of each digit. In the first (introduction) phase, a new digit class was presented 50% of the time. The second phase occurred once performance on this digit reached 95% accuracy on the last 100 presentations of the class. Now all known digits were presented equally often. Once all digits reached a first criterion (65% on the last 100 trials), the process was repeated for the next digit. When the model reached a second criterion performance (80%) for all MNIST digits on the last 100 trials, it was evaluated on 200 images from the test dataset in the absence of new learning and exploration. Pretraining concluded when the model attained criterion performance (80%) on all digits. All datasets used during pretraining phases contained a total of 150.000 training images. The representational distance between MNIST digits (based on principal component analysis) determined the order of digit presentation. Specifically, the order was 0, 1, 7, 6, 8, 3, 2, 9, 4, and then 5. The networks could learn position matching without pretraining.

3.2.9 Hyper-parameters

We tested both smaller ("regular") networks and larger ("over-parameterised") networks on both tasks. Networks were only tested on either identity or position matching to prevent carry-over effects. Regular networks contained 750 and 250 Light-GRU units in the first and second hidden layer, respectively, while over-parameterised networks had a first hidden layer with 1250 of these units and a second hidden layer with 550 units. Over-parameterised networks were used to assess whether feature selectivity is affected by network size. We tested three seeds per condition.

The learning rate for regular networks on identity matching was 0.02 and 0.015 for position matching. For over-parameterised networks these values were 0.0075 for identity matching and 0.005 for position matching. These parameters were chosen based on a grid search and represented the highest learning rate that could be achieved while maintaining stability of the network. The learning rate for the gating units was smaller than the overall learning rate to improve stability, and was scaled down by a factor of 0.8 in the first hidden layer and 0.6 in the second hidden layer. All networks had an exploration rate ϵ of 0.05, a tag decay λ of 0.4, reward discount rate γ of 0.9.

3.2.10 Statistical analyses

To assess selectivity in trained networks, we fitted multiple linear regression models to each unit's activations at four timesteps: the first cue presentation, the two memory delay steps and the second cue presentation. For the first three timesteps, the predictors included in the model (dummy coding) were the identity and position of the first digit and their interaction. Afterwards, an elastic lasso regression assessed which coefficients were important to include in the model. K-fold cross-validation (k=5) was used to determine the L1 and L2 ratio. For the L1 ratio, values of 0.1, 0.5, 0.7, 0.9 and 1.0 were sampled, and for the L2 ratio these were 0.01, 0.1 and 0.5. All non-zero coefficients from the elastic lasso were registered. If any levels of the interaction term were non-zero, the final model included both main effects. If not, it included all levels of only the features with non-zero coefficients for at least one level (e.g. '2' for identity). If a feature had no non-zero levels, this feature was discarded. The final features were then used to fit an ordinary least squares regression, unless an omnibus test, a Durbin-Watson test or the Jarque-Bera test indicated heteroscedasticity, skewness or kurtosis. In this case a robust regression (using Tukey's Biweight) was fit instead.

If the final model significantly predicted variability in unit activations (at an alpha of 0.01), we counted which levels of the features significantly predicted the unit activations. If there was at least one significant level of a factor, we labelled this unit as 'selective' for that factor. Bonferroni correction with an alpha level of 0.01 was used for this procedure. This procedure was repeated for every unit of every type so that it could be registered which percentage of unit types was selective for which task features.

We also conducted the selectivity analysis for the second cue presentation, using the identity and position of the second feature, as well as match status with the first feature and all first-order and second-order interaction effects between these features. In this analysis we first evaluated the highest-order interaction for significance. If significant, the model included all other effects as well. If not, all lower-order interaction effects were assessed and the features involved in significant interactions were included in the model. If no interaction effects were significant, only main significant effects were included.

We repeated this procedure for all three initialisations of the network for each condition (identity matching, position matching) and network size (standard or over-parameterised).

3.3 Results

We investigated whether the biologically plausible learning rule RECOLLECT, designed for networks with a memory layer (van den Berg et al., 2024) could be extended to deeper architectures with multiple layers with memory units. We studied whether the memories differ across the layers between tasks in which spatial locations or letter identities needed to be memorised. We trained three networks to solve a version of the task in which the networks needed to match the identity of two consecutive digits, and three additional networks (with the same initialisation) to match the position of the digits. We first describe the performance of the networks trained with Stackollect. We then investigate the selectivity of units in the different layers. We conclude with a preliminary investigation of the influence of network over-parametrisation on performance and feature representation for both task variants.

3.3.1 Task performance

Networks trained with Stackollect successfully learned to match to position. Taken across three initialisations of the network, learning converged (for criterion see Methods) on average after 212,781 ± 3,179 (mean ± s.d.) trials (Figure 3.4). The accuracy was high, with the models correctly classifying the match status of all stimuli with 86.9 ± 1.4%. Match and mismatch trials were classified with 85.3 ± 3.2% and 87.5 ± 2.0% accuracy, respectively.

In the identity matching task, the networks had to memorise the identity of a digit during a delay and match it to the identity of a second digit. On average, the networks completed the pretraining phase after 969,182 ± 59,632 trials and reached an accuracy of 86.5 ± 0.4%. The models needed another 2,294,752 ±

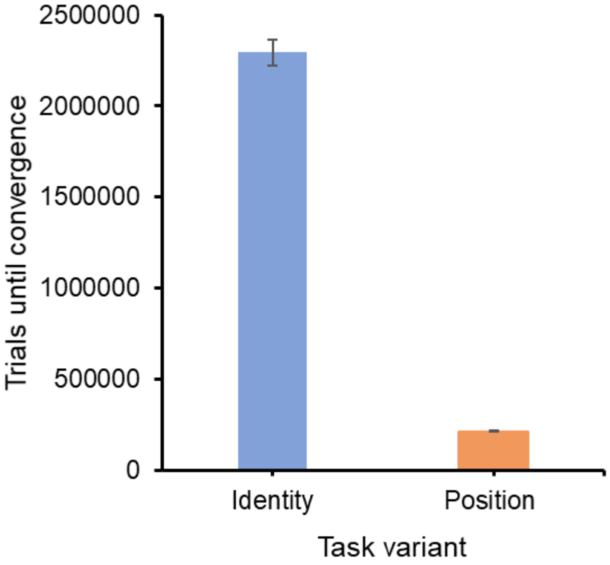

Figure 3.4: Number of trials to convergence for identity and position matching networks. Error bars denote the standard deviation.

70,763 trials to reach convergence on identity matching. The overall accuracy was 85.1 ± 2.8%, that for mismatches was 86.5 ± 3.3% and for matches 72.7 ± 2.4%.

3.3.2 Position matching

Next, we conducted a selectivity analysis with the weights fixed and exploration disabled to elucidate which representations emerged in the position matching task. We created multiple linear regression models based on the activations of the network during the task (for details see Methods). We calculated the percentage of units that were selective for a particular feature or feature combination (Figure 3.5).

During the presentation of the first cue, most units were selective for position, identity and their interaction. Interestingly, the percentage of units encoding the position and identity of the first cue was higher in the first than in the second layer. The output units were primarily selective for identity information, but showed large variability across model instantiations and were only moderately selective for position and the interaction between position and identity.

During the memory delay, the selectivity of units in the second layer was similar to that during the first cue. The candidate memory and gating units in the first layer units did not receive any input during the delay. The output units now exhibited some tuning to identity, position and the interaction between these features.

During the second cue presentation, units were tuned to the interaction between identity and position of the second cue, and the interaction with the position of the first cue, which is related to determining the position match. The units of the first and second layer were selective for the second cue position and the interaction between identity and position and the interaction between position, identity and match. As required by the task, output units were primarily selective for match information.

Overall, the tuning profile in the two layers was varied, except in the output layer, where units primarily encoded match information.

3.3.3 Identity matching

We repeated the selectivity analysis for the networks trained on identity matching. The tuning of the units in the first layer was similar to that in the position match task (Figure 3.6), but differences emerged in the second hidden layer, because the percentage of selective units was lower. Interestingly, the output units were selective for digit identity during the first cue presentation.

During the presentation of the second cue, the units in the second layer became slightly more sensitive to identity than in the position matching task, and this difference was, as expected, most apparent for output units. All output units were selective for match information and a majority was also selective for identity and the interaction between identity and match information.

Overall, we did not observe large differences in the percentages of tuned units in the various layers between the tasks. We also examined whether the absence of these differences was related to compression effects caused by a limited network size. Hence, we also trained "over-parameterised" networks with more units. We increased the number of units in the first memory layer from 750 to 1250 and in the second memory layer from 250 to 550. However, we found no notable differences compared to the networks described above (see Appendix).

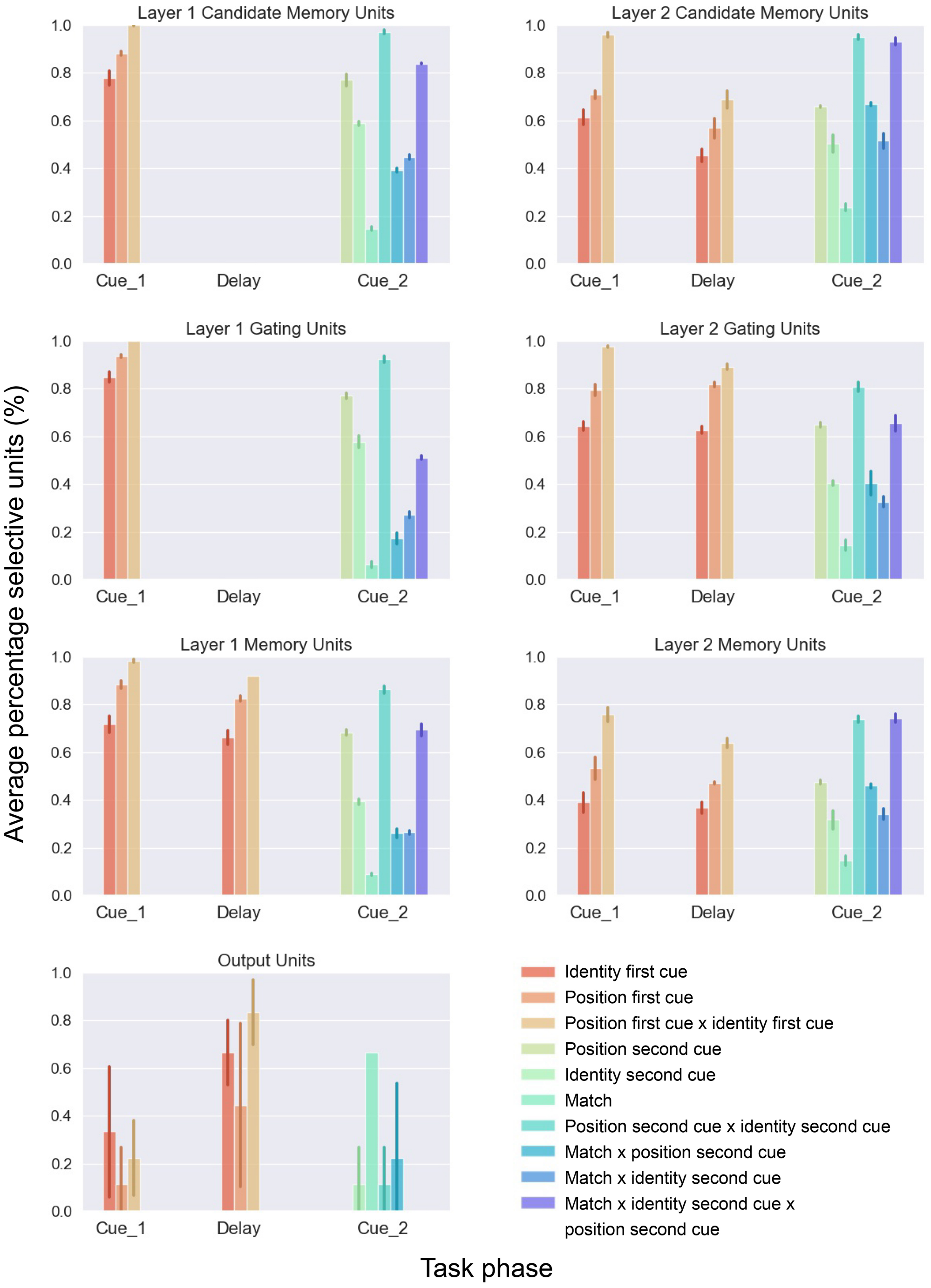

Figure 3.5: Feature selectivity of Stackollect units across network layers for the delayed match-to-position task during the presentation of the first cue ('Cue_1'), the memory delay period ('Delay'; averaged across the two timesteps), and the second cue presentation ('Cue_2'). Error bars represent the standard deviation and were capped if they extended beyond 0% and 100%.

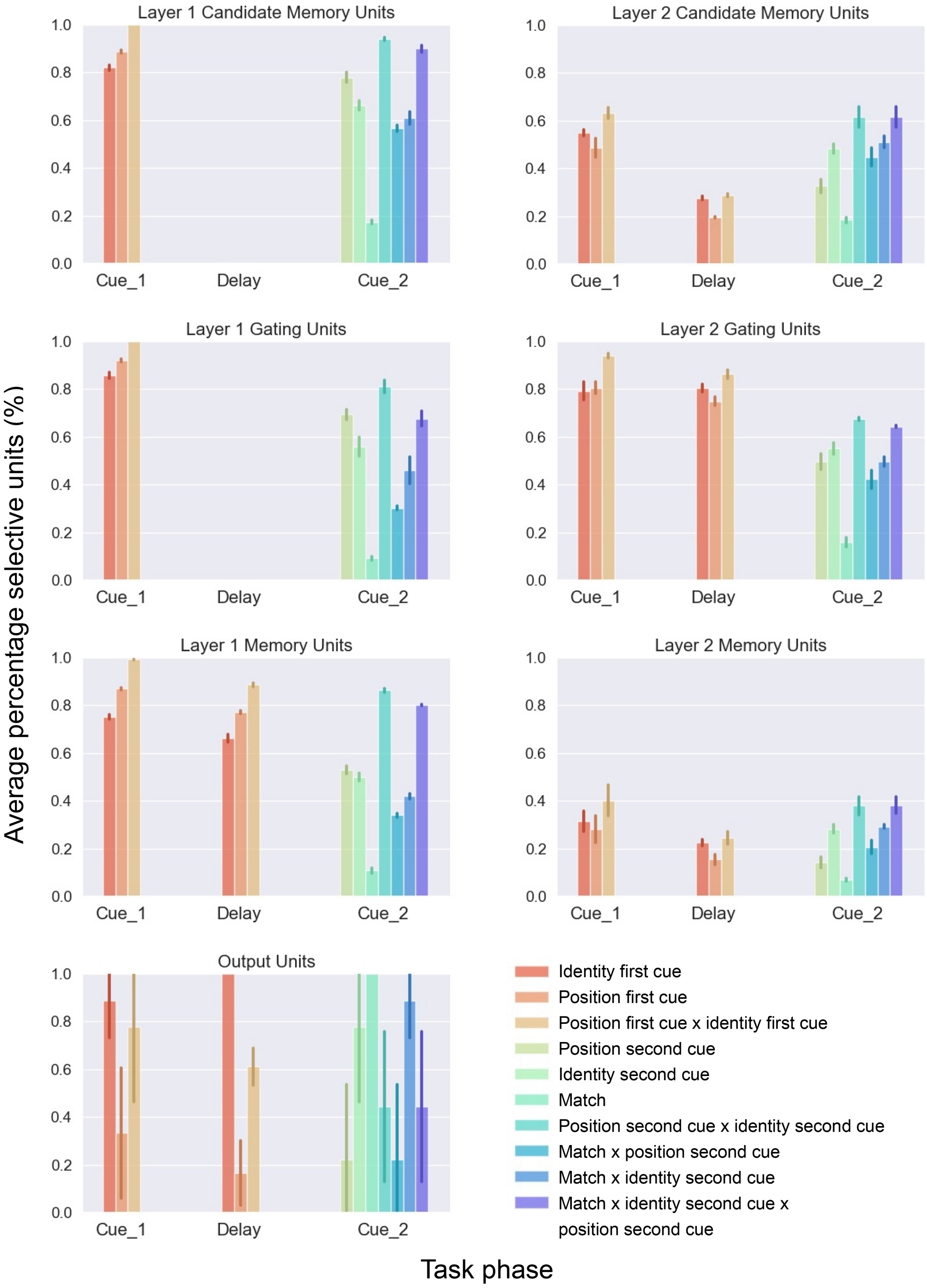

Figure 3.6: Feature selectivity of Stackollect units across network layers for the delayed match-to-identity task during the presentation of the first cue (‘Cue_1’), the memory delay period (‘Delay’; averaged across the two timesteps), and the second cue presentation (‘Cue_2’). Error bars represent the standard deviation and were capped if they extended beyond 0% and 100%.

3.4 Discussion

This study aimed to develop a biologically plausible learning rule for deep gated memory networks. We used a task in which networks had to match the identity or position of digits and examined the emerging representations when biologically plausible memory networks learned these tasks.

The resulting architecture – Stackollect – combines the RECOLLECT (van den Berg et al., 2024) and BrainProp (Pozzi et al., 2020) learning rules to approximate BPTT for networks using only local information at the level of the synapse. Stackollect can train deeper architectures, giving it an advantage over alternative biologically plausible learning rules for recurrent architectures, such as AuGMEnT (Rombouts et al., 2015) and e-prop (Bellec et al., 2018). Stackollect included Light-GRU units with one gate (Ravanelli et al., 2018), which are simpler than the units of long-short term memory (LSTM) models, which have three gates (Hochreiter & Schmidhuber, 1997). Units in Stackollect are therefore easier to map to the brain (van den Berg et al., 2024), without sacrificing performance (Cho et al., 2014; Ravanelli et al., 2018; van den Berg et al., 2024).

The memory units in the hidden layers of Stackollect would allow the flexible remembering and forgetting of information at different levels of abstraction. However, we did not observe clear differences in the tuning in the two memory layers and tuning did not depend strongly on the task. To examine whether these findings were explained by compression effects caused by a limited network size, we also trained over-parameterised networks but observed similar patterns of selectivity.

It is, however, conceivable that larger differences between the memories in different layers would emerge in deeper networks or in networks trained in other types of tasks. The use of complex visual stimuli and deeper networks would encourage different features and hierarchical encoding of features. A further limitation is our use of fully connected networks allowing the integration of all information at early network levels. Indeed, studies using convolutional neural networks demonstrated a more gradual integration of visual features in successive layers (Bilal et al., 2018; Lee et al., 2009). Future studies could adapt the Stackollect learning rule to networks with locally-connected weights to examine the emergence of hierarchical representations in networks trained with biologically plausible learning rules.

In conclusion, the current work adds to the existing literature on working memory encoding and extends it to networks trained using biologically plausible learning rules. Because the Stackollect learning rule generalises to even deeper networks than those tested here, this opens up the opportunity for using biologically plausible deep gated memory architectures on more complicated tasks.

3.5 Appendix

3.5.1 Non-truncated credit assignment to lower layers

Here, we will derive the credit assignment to lower layers for $t-1$ and $t-2$ to illustrate the complexity and non-locality of credit assignment when truncation is not employed.

First, we will derive the path from $M_k^2(t)$ to $M_k^2(t-1)$ and $M_j^1(t-1)$ for $t-1$:

$$\frac{\partial q_s(t)}{\partial u_{ij}^{k^1}(t-1)} = \sum_k \left[\frac{\partial q_s(t)}{\partial M_k^2(t)} \frac{\partial M_k^2(t)}{\partial M_k^2(t-1)} \left[\frac{\partial M_k^2(t-1)}{\partial k_k^2(t-1)} \frac{\partial k_k^2(t-1)}{\partial M_j^1(t-1)} \right.\right.$$
$$\left.\left. + \frac{\partial M_k^2(t-1)}{\partial C_k^2(t-1)} \frac{\partial C_k^2(t-1)}{\partial M_j^1(t-1)} \right]\right] \frac{\partial M_j^1(t-1)}{\partial k_j^1(t-1)} \frac{\partial k_j^1(t-1)}{\partial u_{ij}^{k^1}(t-1)},$$

$$\frac{\partial q_s(t)}{\partial u_{ij}^{k^1}(t-1)} = \sum_k \left[w_{ks} ReLU'(M_k^2(t)) k_k^2(t) \left[ReLU'(M_k^2(t-1)) [M_k^2(t-2) - C_k^2(t-1)] \right.\right.$$
$$\left.\left. \sigma'(Inp_k^{k^2}(t-1)) v_{jk}^{k^2} + ReLU'(M_k^2(t-1))[1 - k_k^2(t-1)] \sigma'(Inp_k^{C^2}(t-1)) v_{jk}^{C^2} \right]\right]$$
$$ReLU'(M_j^1(t-1))[M_j^1(t-2) - C_j^1(t-1)] \sigma'(Inp_k^{k^1}(t-1)) x_i(t-1).$$

The second path flows from $M_k^2(t)$ to $M_j^1(t)$ and then to $M_j^1(t-1)$:

$$\frac{\partial q_s(t)}{\partial u_{ij}^{k^1}(t-1)} = \sum_k \left[\frac{\partial q_s(t)}{\partial M_k^2(t)} \left[\frac{\partial M_k^2(t)}{\partial k_k^2(t)} \frac{\partial k_k^2(t)}{\partial M_j^1(t)} + \frac{\partial M_k^2(t)}{\partial C_k^2(t)} \frac{\partial C_k^2(t)}{\partial M_j^1(t)} \right]\right] \frac{\partial M_j^1(t)}{\partial M_j^1(t-1)}$$
$$\frac{\partial M_j^1(t-1)}{\partial k_j^1(t-1)} \frac{\partial k_j^1(t-1)}{\partial u_{ij}^{k^1}(t-1)},$$

$$\frac{\partial q_s(t)}{\partial u_{ij}^{k^1}(t-1)} = \sum_k \left[w_{ks} \left[ReLU'(M_k^2(t)) [M_k^2(t-1) - C_k^2(t)] \sigma'(Inp_k^{k^2}(t)) v_{jk}^{k^2} \right.\right.$$
$$\left.\left. + ReLU'(M_k^2(t))[1 - k_k^2(t)] \sigma'(Inp_k^{C^2}(t)) v_{jk}^{C^2} \right]\right] ReLU'(M_j^1(t)) k_j^1(t)$$
$$ReLU'(M_j^1(t-1))[M_j^1(t-2) - C_j^1(t-1)] \sigma'(Inp_k^{k^1}(t-1)) x_i(t-1).$$

Therefore, the full update at *t*-1 becomes:

$$\frac{\partial q_s(t)}{\partial u_{ij}^{k1}(t-1)} = \sum_k \Bigg[\frac{\partial q_s(t)}{\partial M_k^2(t)}\Bigg[\frac{\partial M_k^2(t)}{\partial M_k^2(t-1)}\Bigg[\frac{\partial M_k^2(t-1)}{\partial k_k^2(t-1)}\frac{\partial k_k^2(t-1)}{\partial M_j^1(t-1)} + \frac{\partial M_k^2(t-1)}{\partial C_k^2(t-1)}$$

$$\frac{\partial C_k^2(t-1)}{\partial M_j^1(t-1)}\Bigg]\Bigg]\Bigg[\frac{\partial M_k^2(t)}{\partial k_k^2(t)}\frac{\partial k_k^2(t)}{\partial M_j^1(t)} + \frac{\partial M_k^2(t)}{\partial C_k^2(t)}\frac{\partial C_k^2(t)}{\partial M_j^1(t)}\Bigg]\frac{\partial M_j^1(t)}{\partial M_j^1(t-1)}\Bigg]\Bigg]\frac{\partial M_j^1(t-1)}{\partial k_j^1(t-1)}\frac{\partial k_j^1(t-1)}{\partial u_{ij}^{k1}(t-1)}.$$

At $t-2$ this becomes even more complicated, given that three paths need to be computed:

1. $M_k^2(t)$ to $M_j^1(t)$, to $M_j^1(t-1)$, to $M_j^1(t-2)$.
2. $M_k^2(t)$ to $M_k^2(t-1)$, to $M_j^1(t-1)$, to $M_j^1(t-2)$ [new at t-1].
3. $M_k^2(t)$ to $M_k^2(t-1)$, to $M_k^2(t-2)$, to $M_j^1(t-2)$ [new at t-2].

The first path is:

$$\frac{\partial q_s(t)}{\partial u_{ij}^{k1}(t-2)} = \sum_k \Bigg[\frac{\partial q_s(t)}{\partial M_k^2(t)}\Bigg[\frac{\partial M_k^2(t)}{\partial k_k^2(t)}\frac{\partial k_k^2(t)}{\partial M_j^1(t)} + \frac{\partial M_k^2(t)}{\partial C_k^2(t)}\frac{\partial C_k^2(t)}{\partial M_j^1(t)}\Bigg]\Bigg]\frac{\partial M_j^1(t)}{\partial M_j^1(t-1)}$$

$$\frac{\partial M_j^1(t-1)}{\partial M_j^1(t-2)}\frac{\partial M_j^1(t-2)}{\partial k_j^1(t-2)}\frac{\partial k_j^1(t-2)}{\partial u_{ij}^{k1}(t-2)}.$$

The second path is:

$$\frac{\partial q_s(t)}{\partial u_{ij}^{k1}(t-2)} = \sum_k \Bigg[\frac{\partial q_s(t)}{\partial M_k^2(t)}\frac{\partial M_k^2(t)}{\partial M_k^2(t-1)}\Bigg[\frac{\partial M_k^2(t-1)}{\partial k_k^2(t-1)}\frac{\partial k_k^2(t-1)}{\partial M_j^1(t-1)} + \frac{\partial M_k^2(t-1)}{\partial C_k^2(t-1)}$$

$$\frac{\partial C_k^2(t-1)}{\partial M_j^1(t-1)}\Bigg]\Bigg]\frac{\partial M_j^1(t-1)}{\partial M_j^1(t-2)}\frac{\partial M_j^1(t-2)}{\partial k_j^1(t-2)}\frac{\partial k_j^1(t-2)}{\partial u_{ij}^{k1}(t-2)}.$$

The third path is:

$$\frac{\partial q_s(t)}{\partial u_{ij}^{k1}(t-2)} = \sum_k \Bigg[\frac{\partial q_s(t)}{\partial M_k^2(t)}\frac{\partial M_k^2(t)}{\partial M_k^2(t-1)}\frac{\partial M_k^2(t-1)}{\partial M_k^2(t-2)}\Bigg[\frac{\partial M_k^2(t-2)}{\partial k_k^2(t-2)}\frac{\partial k_k^2(t-2)}{\partial M_j^1(t-2)}$$

$$+ \frac{\partial M_k^2(t-2)}{\partial C_k^2(t-2)}\frac{\partial C_k^2(t-2)}{\partial M_j^1(t-2)}\Bigg]\Bigg]\frac{\partial M_j^1(t-2)}{\partial k_j^1(t-2)}\frac{\partial k_j^1(t-2)}{\partial u_{ij}^{k1}(t-2)}.$$

Which leads to the full update:

$$
\begin{aligned}
&\frac{\partial q_s(t)}{\partial u_{ij}^{k1}(t-2)} = \sum_k \bigg[\frac{\partial q_s(t)}{\partial M_k^2(t)} \bigg[\bigg[\frac{\partial M_k^2(t)}{\partial k_k^2(t)} \frac{\partial k_k^2(t)}{\partial M_j^1(t)} + \frac{\partial M_k^2(t)}{\partial C_k^2(t)} \frac{\partial C_k^2(t)}{\partial M_j^1(t)}\bigg] \frac{\partial M_j^1(t)}{\partial M_j^1(t-1)} \\
&\frac{\partial M_j^1(t-1)}{\partial M_j^1(t-2)} + \frac{\partial M_k^2(t)}{\partial M_k^2(t-1)} \bigg[\frac{\partial M_k^2(t-1)}{\partial k_k^2(t-1)} \frac{\partial k_k^2(t-1)}{\partial M_j^1(t-1)} + \frac{\partial M_k^2(t-1)}{\partial C_k^2(t-1)} \frac{\partial C_k^2(t-1)}{\partial M_j^1(t-1)}\bigg] \\
&\frac{\partial M_j^1(t-1)}{\partial M_j^1(t-2)} + \frac{\partial M_k^2(t)}{\partial M_k^2(t-1)} \frac{\partial M_k^2(t-1)}{\partial M_k^2(t-2)} \bigg[\frac{\partial M_k^2(t-2)}{\partial k_k^2(t-2)} \frac{\partial k_k^2(t-2)}{\partial M_j^1(t-2)} + \frac{\partial M_k^2(t-2)}{\partial C_k^2(t-2)} \\
&\qquad\qquad\qquad\qquad \frac{\partial C_k^2(t-2)}{\partial M_j^1(t-2)}\bigg]\bigg]\bigg] \frac{\partial M_j^1(t-2)}{\partial k_j^1(t-2)} \frac{\partial k_j^1(t-2)}{\partial u_{ij}^{k1}(t-2)},
\end{aligned}
$$

$$
\begin{aligned}
&\frac{\partial q_s(t)}{\partial u_{ij}^{k1}(t-2)} = \sum_k \bigg[w_{ks} \bigg[\bigg[ReLU'(M_k^2(t))[M_k^2(t-1) - C_k^2(t)]\sigma'(Inp_k^{k^2}(t))v_{jk}^{k^2} \\
&+ReLU'(M_k^2(t))[1-k_k^2(t)]\sigma'(Inp_k^{C^2}(t))v_{jk}^{C^2}\bigg] ReLU'(M_j^1(t))k_j^1(t) ReLU'(M_j^1(t-1)) \\
&k_j^1(t-1) + ReLU'(M_k^2(t))k_k^2(t) \bigg[ReLU'(M_k^2(t-1))[M_k^2(t-2) - C_k^2(t-1)]\sigma'(Inp_k^{k^2}(t-1)) \\
&v_{jk}^{k^2} + ReLU'(M_k^2(t-1))[1-k_k^2(t-1)]\sigma'(Inp_k^{C^2}(t-1))v_{jk}^{C^2}\bigg] ReLU'(M_j^1(t-1))k_j^1(t-1) \\
&+ReLU'(M_k^2(t))k_k^2(t) ReLU'(M_k^2(t-1))k_k^2(t-1) \bigg[ReLU'(M_k^2(t-2))[M_k^2(t-3) - C_k^2(t-2)] \\
&\quad \sigma'(Inp_k^{k^2}(t-2)v_{jk}^{k^2} + ReLU'(M_k^2(t-2))[1-k_k^2(t-2)]\sigma'(Inp_k^{C2}(t-2)v_{jk}^{C2}\bigg]\bigg]\bigg] \\
&\qquad\qquad ReLU'(M_j^1(t-2))[M_j^1(t-3) - C_j^1(t-2)]\sigma'(Inp_k^{k^1}(t-2))x_i(t-2).
\end{aligned}
$$

3.5.2 Over-parameterised networks

To investigate the effect over-parameterisation, we trained larger networks increasing the number of units in the first memory layer from 750 to 1250 and in the second memory layer from 250 to 550. This over-parameterisation decreased learning speed (Figure S.1). The models required 561,350 ± 46,894 trials to learn position matching and 8,559,783 ± 953,678 trials to learn identity matching. Pretraining also took longer (2,011,262 trials on average). Therefore, the bigger networks needed more training to learn the task (roughly 2.6 times as many trials for position matching and 3.7 times as many trials for identity matching).

The selectivity analysis for position matching (Figure S.2) revealed few differences compared to the selectivity of units of smaller networks. Most units were selective for all features during the presentation of the first cue and the delay steps. The selectivity during the presentation of the second cue also followed a

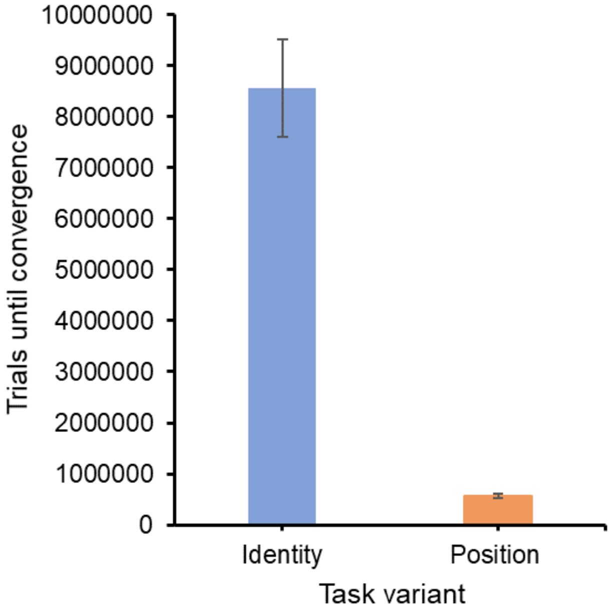

Figure S.1: Number of trials until convergence for over-parameterised networks. Error bars denote the standard deviation.

similar pattern as before. The primary difference was that some output units encoded position and match. We also observed similar patterns of selectivity during all task phases and all types of unit for the bigger networks trained on identity matching (see Figure S.3).

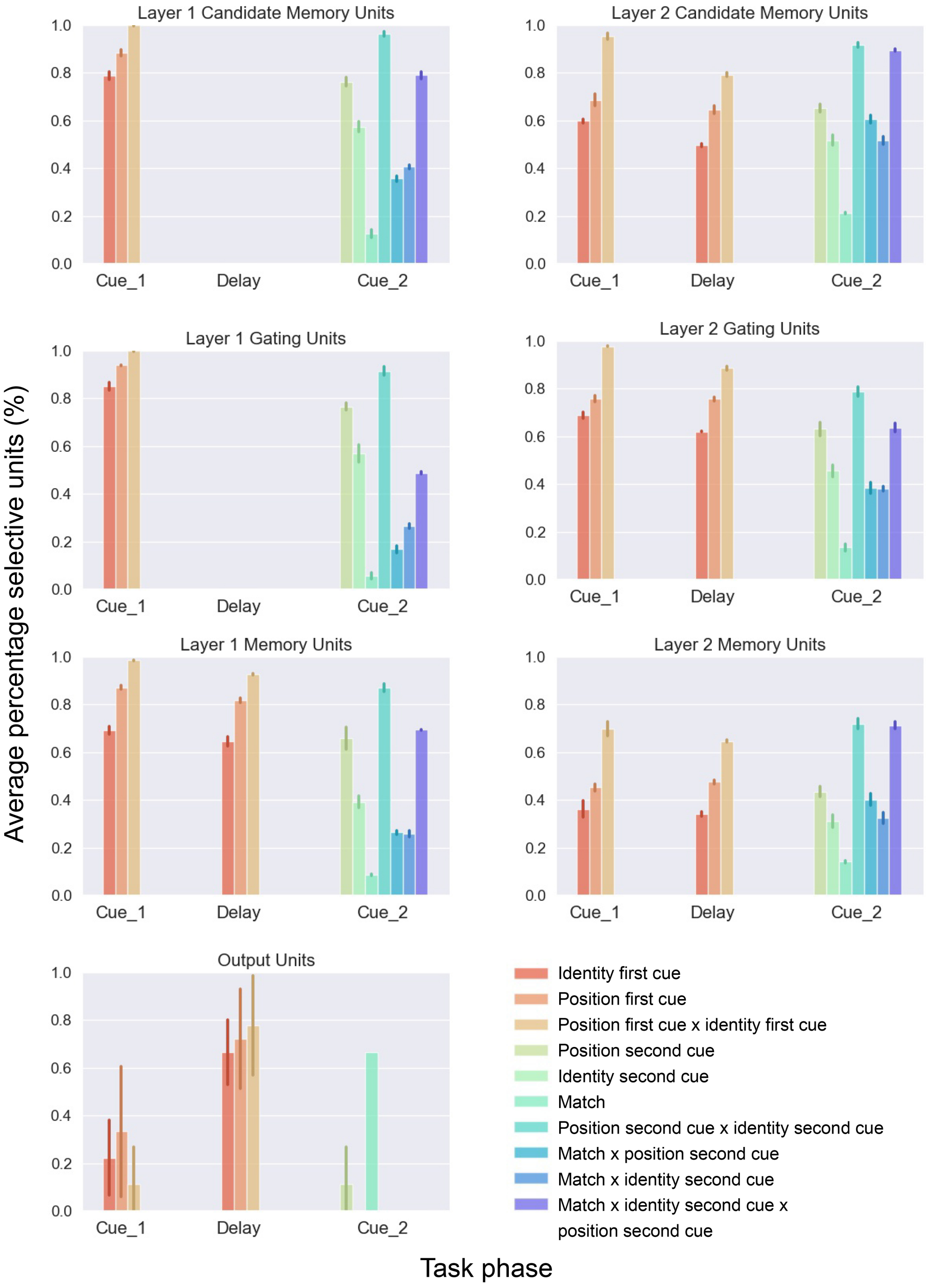

Figure S.2: Feature selectivity of the over-parameterised Stackollect network across units and layers for the delayed match-to-position task during the presentation of the first cue ('Cue_1'), the memory delay period ('Delay'; averaged across the two timesteps), and the second cue presentation ('Cue_2'). Error bars represent the standard deviation and were capped if they extended beyond 0% and 100%.

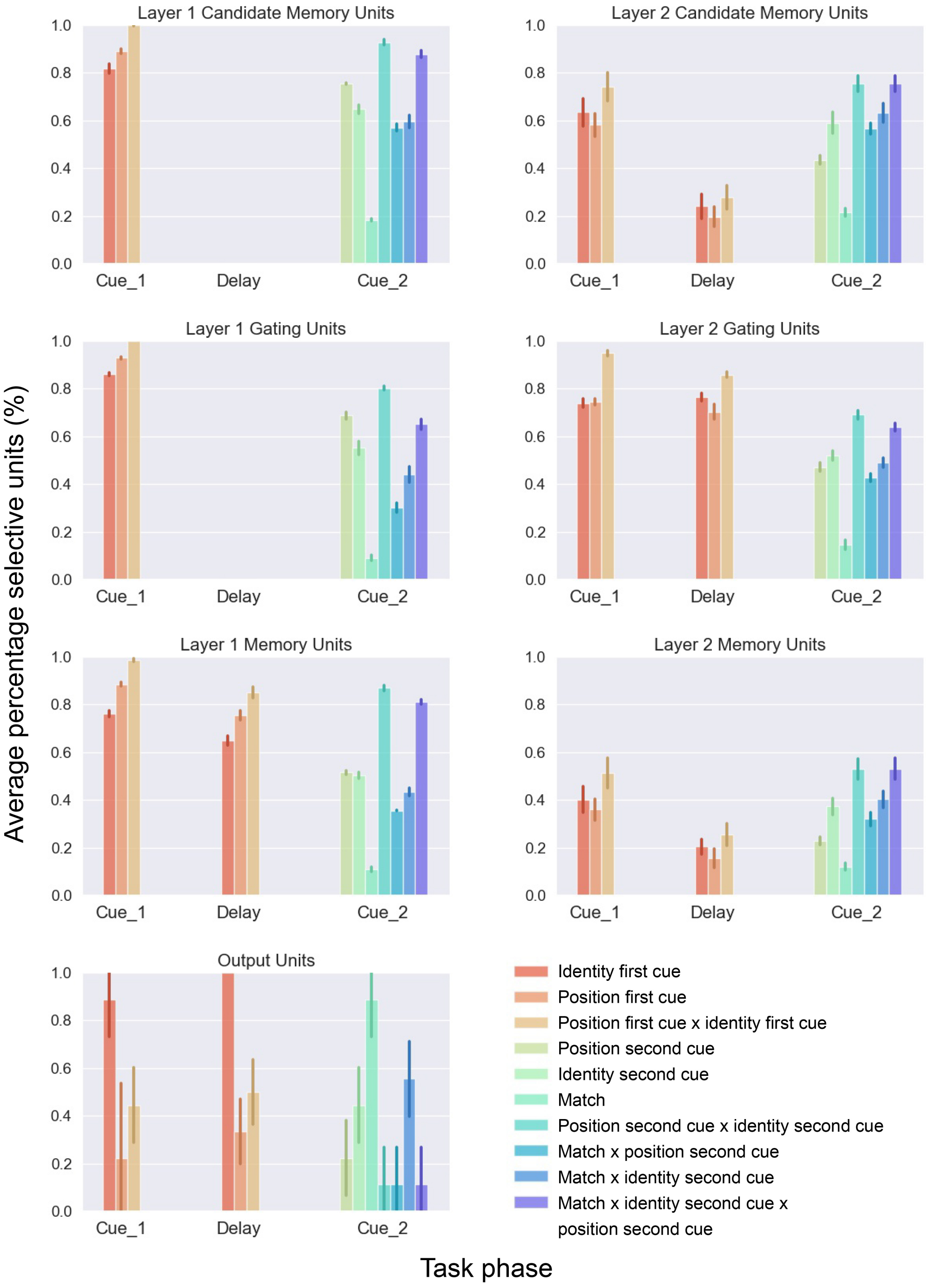

Figure S.3: Feature selectivity of the over-parameterised Stackollect network across units and layers for the delayed match-to-identity task during the presentation of the first cue ('Cue_1'), the memory delay period ('Delay'; averaged across the two timesteps), and the second cue presentation ('Cue_2'). Error bars represent the standard deviation and were capped if they extended beyond 0% and 100%.

Chapter 4

Curriculum Design for Scalable Biologically Plausible Deep Reinforcement Learning

Alexandra R. van den Berg[1,2], Pieter R. Roelfsema[2,3,4,5], Sander M. Bohté[1,6]

[1] *Machine Learning Group, Centrum Wiskunde & Informatica, Amsterdam, the Netherlands*

[2] *Department of Vision & Cognition, Netherlands Institute for Neuroscience, Amsterdam, the Netherlands*

[3] *Department of Integrative Neurophysiology, Center for Neurogenomics and Cognitive Research, Vrije Universiteit Amsterdam, Amsterdam, the Netherlands*

[4] *Department of Neurosurgery, Academic Medical Center, Amsterdam, the Netherlands*

[5] *Laboratory of Visual Brain Therapy, Sorbonne Université, Institut National de la Santé et de la Recherche Médicale, Centre National de la Recherche Scientifique, Institut de la Vision, Paris, France*

[6] *Swammerdam Institute of Life Sciences, University of Amsterdam, Amsterdam, the Netherlands*

Abstract Humans have a remarkable capacity for learning, yet neuronal learning is constrained to locality in time and space and limited feedback. While neural learning rules have been designed that adhere to these principles and constraints, they exhibit difficulty in scaling to deep networks and complicated datasets. BrainProp is a biologically plausible learning rule, learning from trial-and-error feedback through reinforcement learning, that does generalise to deep networks and achieves good performance on traditional machine learning benchmarks. It does however falter on problems with a large number of output categories, such as the classical ImageNet vision benchmark: while standard BrainProp eventually succeeds, learning is not robust and highly sensitive to hyper-parameter optimisation and proper initialisation. Here, we leverage insights from behavioural science by developing a curriculum that structures how samples are presented to a network to optimise learning. The key features of the curriculum involve progressively introducing new classes to the dataset based on performance metrics, and using a recency bias to protect recently acquired classes. We demonstrate that our curriculum approach makes BrainProp-style learning robust and more rapid, while substantially improving classification accuracy. We also show the curriculum similarly improves performance for networks trained using error-backpropagation. We thus establish a new state-of-the-art performance for large-scale deep reinforcement learning. Our results show the potential of curriculum learning in local learning settings with limited feedback and further bridges the gap between biologically plausible learning rules and error-backpropagation.

4.1 Introduction

The human brain is remarkable in its capacity for learning. By modelling the brain, we can gain both a deeper understanding of biology, and biology itself also provides a source of inspiration for AI. In particular for AI constrained to efficient and effective learning rules in settings with limited feedback. How does learning differ between biological and artificial systems?

One crucial way how animal learning differs from AI is the source and locality of information used for neural learning. Credit assignment in classical error-backpropagation (EBP) is performed in a supervised manner, where each synapse is informed about its contribution to the output across all layers of the neural network, as well as about what the output should have been when errors are made. However, this premise is biologically implausible for two reasons. Firstly, it assumes that each neuron has access to this non-local information. Secondly, a large amount of learning in humans and animals occurs without explicit feedback about which outputs should have been generated when errors are made. Instead, the environment either provides a reward – or not – and one has to infer through experience what the correct output (or action) should be (Sutton & Barto, 2018). Thus, we hereby consider "biologically plausible" to mean learning that is done through reinforcement and using information available only locally, at the level of the synapse.

To overcome these limitations, BrainProp was proposed as a biologically plausible learning scheme for training deep neural networks. BrainProp achieves state-of-the-art performance on traditional machine learning benchmarks such as CIFAR-100 and TinyImageNet (Pozzi et al., 2020) when set in a reinforcement learning paradigm. BrainProp learns exclusively through trial-and-error, and has to discover the correct output for each class. Training on datasets with many classes therefore becomes very challenging. If the model selects the wrong class, the model first has to unlearn this class and sample others until the right class is found. As the number of classes grows, the probability diminishes that the model guesses correctly, thereby making the class selection problem exceedingly hard. As a result, biologically plausible reinforcement learning methods have thus far been unable to scale to the size and complexity of benchmarks such as the ImageNet vision classification benchmark, which requires deep networks for the classification of 1,000 categories, despite extensive effort. Here, we show a way to overcome this problem by drawing inspiration from how biological systems learn.

In contrast to AI, where networks are often repeatedly exposed to large amounts of unstructured data, humans typically learn in a very structured manner. Rather than immediately attempting high-level mathematics, children first learn how to perform simple addition and subtraction before they are exposed to increasingly complex problems. These types of curricula form the foundation of modern education systems and are designed to facilitate efficient learning (He, Schultz, & Schubert, 2015). The power of using a curriculum becomes even more evident

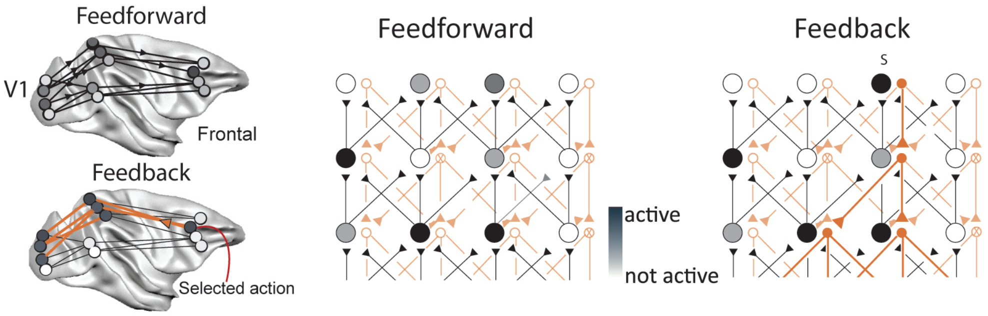

Figure 4.1: The BrainProp algorithm is a three-factor learning rule that, based on the presence of feedforward activity and a reward prediction error, assigns credit to synapses involved in action selection using a feedback network. Figure reproduced from Pozzi et al. (2020).

when training animals, since one cannot rely on verbal instructions. In a series of classic experiments in the early 1900s, Skinner already showed that by initially rewarding animals for exhibiting very simple motions and afterwards rewarding them for more complex behavioural sequences (i.e. "shaping"), animals quickly acquire new behaviour that they might otherwise learn very slowly – if at all. For this reason, using a curriculum is considered standard practice when training animals on behavioural tasks (Pryor & Ramirez, 2014; McGreevy & Boakes, 2011).

Within AI, numerous studies have also demonstrated the beneficial effects of using curricula for reinforcement learning (Narvekar et al., 2020; Soviany et al., 2022). One potential contributor to the success of curriculum learning regards the difficulty level of training examples, which has been shown to be an important influence on training efficiency (Weinshall et al., 2018; Weinshall & Amir, 2020; Zaremba & Sutskever, 2014). A curriculum could allow for initial optimisation of an easy function, which can then progressively be refined as it becomes more non-convex through more difficult examples (Bengio, 2009). It has been suggested that the most optimal training regime is one in which training accuracy hovers around 85% (Wilson et al., 2019); thus not being too difficult, but also not too easy. A curriculum can be used to dynamically adjust the difficulty of training examples to achieve this state. Another advantage of curriculum learning concerns the acquisition of primitives underlying more complex behaviour, which could facilitate learning the overarching behaviour (Lee, Mannelli, & Saxe, 2024; Hocker, Constantinople & Savin, 2024; Dekker, Otto & Summerfield, 2022), and result in networks behaving more similarly to animals on the same task (Hocker, Constantinople & Savin, 2024). In fact, given that associative learning in mice has been shown to occur in a stepwise manner (Manzur et al., 2023), forcing an artificial neural network to learn in discrete steps using a curriculum might therefore not only simplify the learning process, but also result in more biologically plausible learning.

We here developed a novel curriculum to overcome the class-learning problem for biologically plausible learning rules based on reinforcement learning – such as BrainProp. The curriculum, specifically a form of structured class-incremental curriculum learning, operates by gradually introducing each class to the network. We first demonstrate that with precise initialisation and optimisation BrainProp can scale to ImageNet, but that the standard trial-and-error learning is very slow, and unstable because only few initialisations converge, and networks that appear to learn well regularly collapse. Next, we show that using the curriculum makes learning robust, more rapid and more accurate. We thereby establish a new state-of-the-art performance for biologically plausible reinforcement learning on the ImageNet benchmark, and demonstrate the ability of approaches such as BrainProp to scale to biologically relevant domain sizes and network complexities with local learning and limited feedback. Finally, we show that a curriculum also aids in the performance of EBP on ImageNet, and conduct an ablation analysis to investigate which elements are important for the curriculum's success.

4.2 A curriculum for overcoming the class-learning problem

4.2.1 Biologically plausible learning

BrainProp is a biologically inspired reinforcement learning scheme for neural networks that performs credit assignment using feedback connections to successively lower layers in the network (Pozzi et al., 2020); see Fig. 4.1). We here briefly summarise its working mechanism and how this relates to the class-learning problem.

After choosing an action s based on output layer activations, the network receives reward information r from the environment. This is 1 if the correct choice was made, and 0 otherwise. The model then computes the reward prediction error (RPE; δ), which quantifies the discrepancy between the expected reward (the activation of the chosen unit y_s^N) and the actual reward (r):

$$E(w) = \frac{1}{2}(r - y_s^N)^2 \tag{4.1}$$

RPE signalling is believed to be performed by the dopaminergic system (Schultz, 2016) and in BrainProp is broadcast to the network as a global neuromodulatory signal. The aim of learning is to minimise this RPE over the course of training. Importantly, the RPE is only calculated for the selected class (s) and only those weights that contributed to this action are updated. The resulting weight update for weights ($w_{n,m}^{N-1}$) connecting to the output layer (N) becomes:

$$E(w) = \Delta w_{n,m}^{N-1} = \begin{cases} \delta y_m^{N-1}, & \text{if } n = s \\ 0, & \text{if } n \neq s \end{cases} \tag{4.2}$$

For lower layers, a separate feedback network carries an attentional signal conveying activity (rather than error signals) from the selected output, gating plasticity in the forward network in a manner compatible with locally available information.

For these layers, weight updates rely not only on the RPE, but are also gated by the level of feedback neurons (φ_k^{l+1}) receive from higher layers. Feedback is only provided if a higher-level neuron was active and is propagated using feedback connections ($w_{j,i}^{FB}$) from higher to lower layers. These feedback connections are assumed to be reciprocal to the feedforward connections ($w_{i,j}^{FF}$) within a cortical column; a feature that also emerges for BrainProp over the course of learning (Roelfsema & van Ooyen, 2005). For the output layer, feedback activity emerges from the selected unit:

$$\varphi_n^N = \begin{cases} 1, & \text{if } n = s \\ 0, & \text{if } n \neq s \end{cases} \tag{4.3}$$

Feedback activity from the layer prior is calculated by means of the feedback weights from the output layer and feedback activity of the output layer:

$$\varphi_m^{N-1} = \sum_n w_{n,m}^{FB,N-1} \varphi_n^N = w_{s,m}^{FB,N-1}. \tag{4.4}$$

For all lower layers, feedback activity is computed based on a combination of the feedback weights, feedback activity and the derivative of the activation function (g):

$$\varphi_i^l = \sum_k w_{k,i}^{FB,l} \varphi_k^{l+1} g_k^{l+1}. \tag{4.5}$$

The resulting weight update from any neuron i in lower layers to unit j in the layer above becomes a combination of the RPE, feedback from the higher-level unit (φ_i^l), the derivative of the activation function in the feedforward pathway (g_i^l) and feedforward activity of the unit itself (y_j^{l-1}):

$$\Delta w_{i,j}^{l-1} = \delta \varphi_i^l g_i^l y_j^{l-1}. \tag{4.6}$$

Since BrainProp uses rectified linear units (ReLUs) as the activation function, the resulting derivative becomes 0 if the higher-level unit was inactive and 1 otherwise. As such, the weight update relies on the presence of both feedback activity and feedforward activity, making it Hebbian in nature.

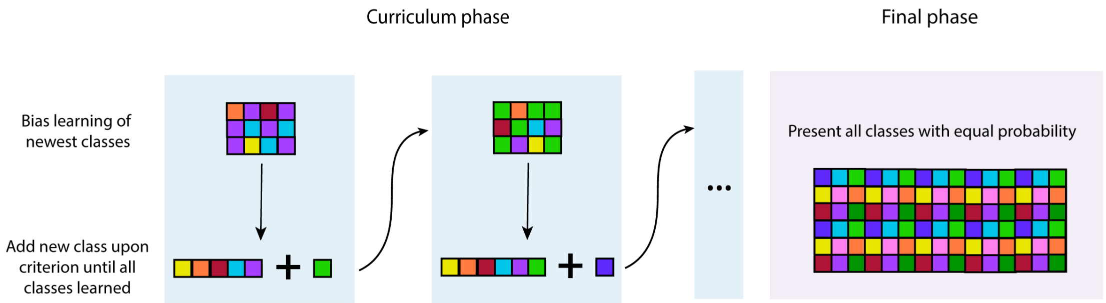
Curriculum phase
Final phase
Bias learning of newest classes
Add new class upon criterion until all classes learned
...
Present all classes with equal probability

Figure 4.2: Training commenced with a curricular pre-training stage, wherein classes are successively added to the training set, while promoting learning of new classes (first purple, then green) while protecting the most recently introduced class prior to that (light blue, followed by purple). Once all ImageNet categories were learned, a final training phase presented the full dataset to the model until convergence.

The main difference between EBP and BrainProp lies in which weights are updated at any given time: while EBP can assign credit to all weights, BrainProp optimises only the subset of connections contributing to the chosen action through the reinforcement learning signal. As a result, the two algorithms are mathematically equivalent in the case that each possible action is sampled by BrainProp once, and the network weights are subsequently altered, for each sample individually. If the action chosen by BrainProp is correct, the network can be updated efficiently. In contrast, if it is incorrect, the network first has to unlearn the class associated with the stimulus before it selects a new class. If this class is also incorrect, this process has to be repeated until the right class is found. Hence, BrainProp enables networks to learn smaller benchmarks to near error-backpropagation accuracy, but learning is less efficient for datasets with large outcome spaces such as ImageNet, because the iterative process of learning and unlearning is slow. Can a curriculum be designed in such a way to remedy this?

4.2.2 Curriculum design

To encourage learning of new classes, we developed a curriculum that gradually exposed the model to the different classes in the ImageNet dataset (see Fig. 4.2). Rather than presenting the full dataset from the beginning, the model was first presented with images from a single class. A new class was added to the dataset once the previous class was learned. This procedure was repeated until all classes were incorporated into the training set. The full network was used from the start, with no alterations made to network size and the number of output units throughout the curriculum. All output units (including those for classes not observed yet) were considered during decision-making and were treated identically during training.

Importantly, the curriculum was designed to incentivise learning of new classes, while protecting the performance on recently introduced classes. Therefore, the newest class was presented with a higher probability (30%) than the classes that were already learned[1], so that the model has sufficient opportunity to sample the outcome categories until it determined the correct class. Moreover, this approach might help the model to learn to not only correctly classify the typical samples of a class (Smith et al., 2010), but also some of the more exceptional cases, allowing the model to delineate boundaries between the categories. The most recently introduced class prior to the current class was also presented with higher probability (10%) to prevent forgetting of this class, given the instability of recently acquired memories (see also Mosha & Robertson, 2016) and the beneficial effects of rehearsal on memory consolidation (Himmer et al., 2019). A new class was introduced once the model either classified the newest class with high accuracy (at least 75% correct on the most recent 25 presentations) or after a maximum of 15 epochs. This process was repeated until all classes in ImageNet had been presented.

Upon completion of the curriculum phase, the model was trained on the full dataset with the same probability of each class (0.1%). The goal of this final training stage was to enhance overall training and validation accuracy on the dataset and to prevent overfitting, because the curriculum contained only a limited number of samples per class (only a total of 50.000 images were shown during this phase). During this second training stage we presented the full training dataset (1.2 million images) to the network to improve generalisation. An early stopping criterion assessed validation accuracy on 5.000 (of 50.000 validation images) with a patience of 45 and a minimum delta of 0.001. Once validation accuracy plateaued, the model was tested on the full validation dataset (excluding the 5.000 validation images that had been used).

4.2.3 Architecture

For all experiments, we used a VGG-inspired network (Pozzi et al., 2020) with seven convolutional layers, followed by two fully-connected layers of sizes 8192x3000 and 3000x1000 (see Fig. 4.3). Epsilon-greedy served as the action selection mechanism for BrainProp. This mechanism usually chose the action associated with the most highly active output unit, but selected an action randomly with a probability of 2%. For EBP we used a softmax function instead. For the curriculum-trained networks, a learning rate of 0.005 was used. BrainProp without curriculum was trained using higher learning rates of 0.04, 0.05 and 0.06. The same learning rates were used for experiments with EBP. A batch size of 125 was utilised and the curriculum evaluated criterion performances every 20th batch. ImageNet stimuli

[1] From the 10th class onwards. Prior to this, all classes were presented with equal probability during the introduction phase.

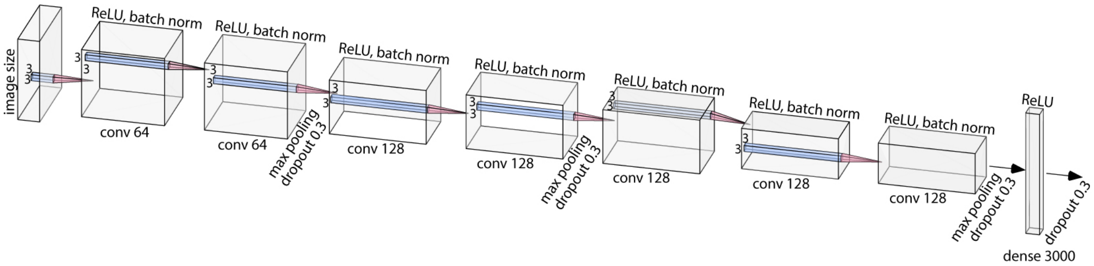

Figure 4.3: Network architecture (figure adapted from Pozzi et al., 2020). Each convolutional layer had a kernel size of 3 by 3 with a stride of 1 and used zero-padding. The first two convolutional layers had 64 channels and all other convolutional layers had 128 channels. Batch normalisation was applied after every convolutional layer. Moreover, max pooling (kernel size and stride of 2) and dropout (0.3) were performed after the second, fourth and seventh layer. A final dropout layer occurred after the first fully-connected layer. ReLU activation functions were employed for all layers except for the output layer.

were downsized to 64x64 pixels, converted to RGB and normalised.

4.3 Experiments

Biologically plausible learning rules have proven difficult to scale to large problems such as ImageNet (Bartunov et al., 2018). We here demonstrate that the BrainProp learning rule can accomplish this under certain conditions, although learning proceeds very slowly and is generally not robust. However, we show that employing a curriculum that gradually introduces each class to the network successfully overcomes these difficulties and substantially improves accuracy, training speed and stability.

Without a curriculum, BrainProp learned to classify ImageNet to some degree, given the right initialisation and with careful hyper-parameter tuning. Networks that do learn, do so very slowly; requiring 1372 (SD = 1295) epochs on average to converge (see Fig. 4.4). Moreover, top-1 and top-5 test accuracy was low, averaging only 6.4% (SD = 8.1%) and 9.9% (SD = 11.9%), respectively. Furthermore, learning was not robust. Of the five seeds trained on the task, two seeds quickly (i.e. within the first epoch) exhibited unstable performance and did not converge. Two of the seeds that initially demonstrated stable learning also eventually showed a sharp decline in their validation accuracy, which dropped to chance level during a later stage of training (around 1300 and 1500 epochs; see Fig. 4.5). When the learning rate was decreased from 0.05 to 0.04 ('lowLR' condition), learning was stable, but while none of the seeds showed deterioration in validation accuracy at a later training stage, performance on three of the five seeds never exceeded chance level (average top-1 and top-5 test accuracy for these seeds of 0.11% and

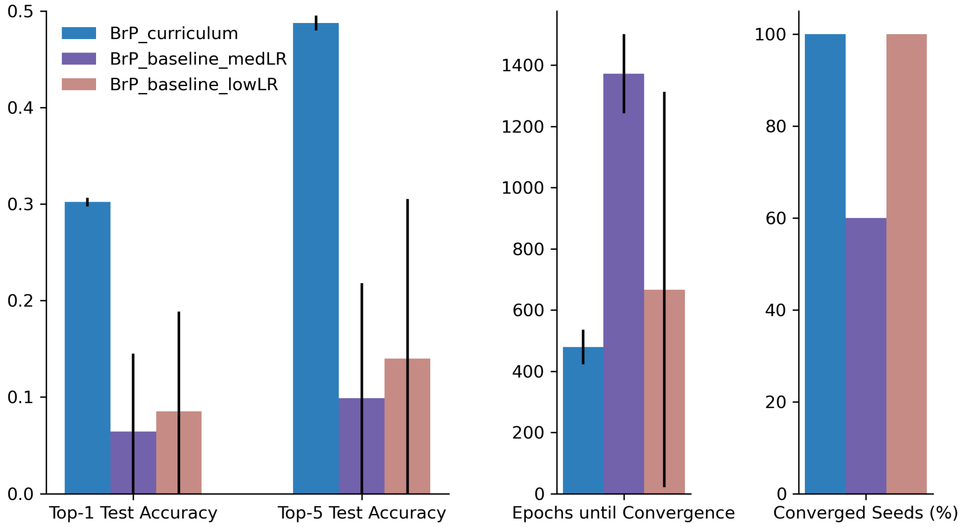

Figure 4.4: Average (+ SD) top-1 and top-5 test accuracy, number of epochs until convergence and percentage of converged seeds for networks trained with BrainProp with a curriculum ('BrP_curriculum') and without the curriculum using two types of learning rates (0.05 for 'BrP_baseline_medLR' and 0.04 for 'BrP_baseline_lowLR'). Networks trained with the highest learning rate (0.06) are not shown since their training was aborted very early due to instabilities. Using a curriculum enhanced performance on all metrics.

0.49%, respectively, with an overall top-1 and top-5 accuracy of 8.55% and 14.0% for all 5 seeds). With a slightly higher learning rate of 0.06, none of the seeds were viable due to numerical instabilities. Thus, while some networks learned, learning was slow, unstable, and sensitive to hyper-parameter changes.

The curriculum largely solved these issues. Classification performance, training speed and robustness all improved substantially. None of the seeds showed instabilities during training, and the networks reached on average a top-1 test accuracy of 30.2% ($SD = 0.5\%$) and a top-5 test accuracy of 48.8% ($SD = 0.7\%$) while requiring only 478 ($SD = 56$) epochs to converge (Fig. 4.4-4.5).

We also trained the network with EBP, with and without the curriculum. As expected, EBP yielded a higher performance than BrainProp in terms of training speed and classification accuracy (Fig. 4.6). Without the curriculum, EBP obtained an average top-1 test accuracy of 35.9% ($SD = 0.2\%$) and an average top-5 test accuracy of 59.9% ($SD = 0.3\%$), while requiring merely 53.8 ($SD = 2.5$) epochs on average to converge. Interestingly, adding the curriculum enhanced top-1 and top-5 classification performance further to 43.4% ($SD = 0.1\%$) and 67.8% ($SD = 0.2\%$), respectively, despite a small increase in the number of epochs required for training ($M = 83$, $SD = 7$). All seeds showed stable performance and converged for each learning rate (Fig. 4.6).

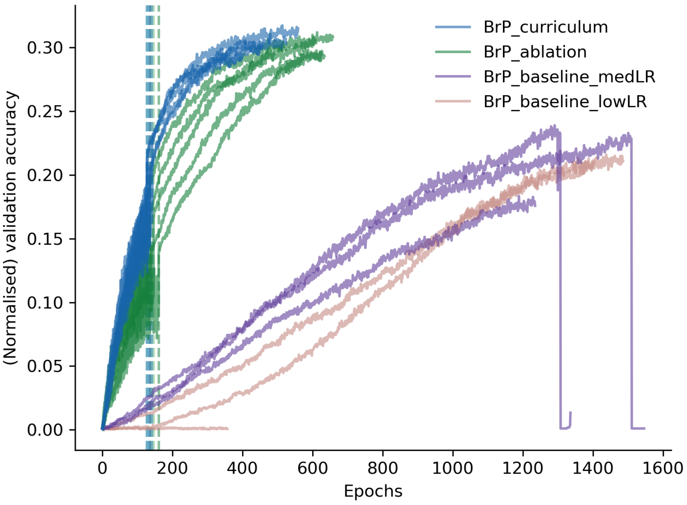

Figure 4.5: (Normalised) validation accuracy during training for all conditions. Dashed lines indicate the completion of the curriculum phase for networks trained in the curriculum condition. Networks trained using BrainProp ('BrP') with a curriculum outperform those without the curriculum in both training speed and validation accuracy.

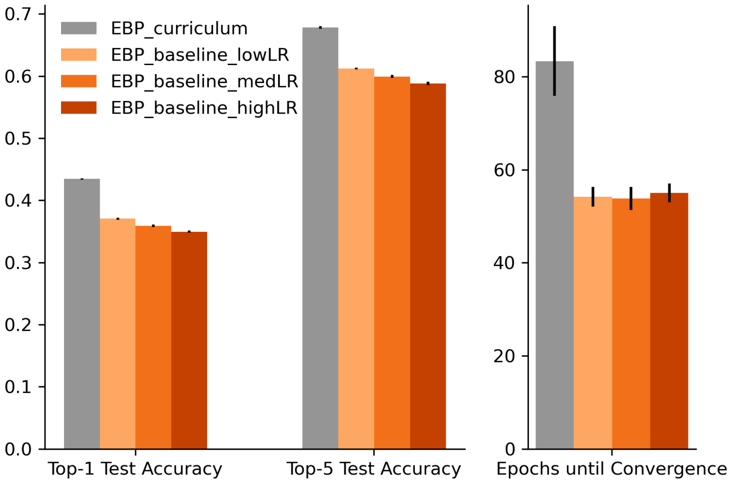

Figure 4.6: Average (+ SD) top-1 and top-5 test accuracy, as well as the number of epochs until the networks converged for EBP when trained with or without a curriculum. We used three learning rates ('low_LR'=0.4, 'med_LR'=0.5, 'high_LR'=0.6). Accuracy was higher and training proceeded faster with the curriculum, for each learning rate.

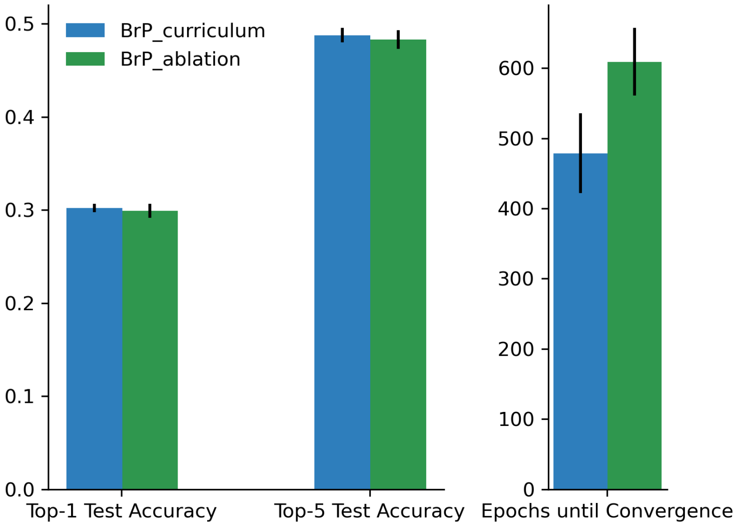

Figure 4.7: Average (+ SD) top-1 and top-5 test accuracy, in addition to the number of epochs required until convergence for networks trained with BrainProp using the full curriculum ('BrP_curriculum') and with a version of the curriculum without recency bias ('BrP_ablation'). Although final performance was comparable, the training speed was higher with the full curriculum.

Finally, we carried out an ablation study to investigate the role of the recency bias in the curriculum (Fig. 4.7). This recency bias corresponds to slightly enhancing the probability of the previous class (10%) to prevent forgetting this class. When networks were trained without this recency bias, top-1 and top-5 test performance were still markedly improved compared to the baseline without curriculum and were comparable to that for the full curriculum (M = 29.9%, SD = 0.7%, and M = 48.3%, SD = 1.0%, respectively). However, the full curriculum converged more than 100 epochs faster than the ablated curriculum without the recency bias (M = 609, SD = 48).

In conclusion, using a curriculum allowed BrainProp to successfully learn on ImageNet, thereby proving to be a useful strategy in training networks with biologically plausible reinforcement learning.

4.4 Discussion

Thus far, it proved to be too challenging to train networks on large scale classification benchmarks such as the ImageNet dataset with biologically plausible reinforcement learning rules. Here, we therefore investigated whether a curriculum would help these networks scale to these larger datasets. We first showed that learning with the biologically plausible learning rule BrainProp on ImageNet is in fact possible, but only given careful hyper-parameter tuning. Moreover, learning was overall very slow and accuracy remained limited.

We speculated that the difficulty of a reinforcement learning rule such as BrainProp in scaling to large datasets could originate from having to sample many different classes before the correct class is found. Therefore, to enhance performance, we generated a curriculum that gradually exposed the network to each class in the dataset. We introduced one new class at a time and biased the selection of samples to the new class so that the model had ample opportunity to discover the correct class label by trial-and-error. When the validation accuracy of the newest class reached criterion, we introduced the next class until the full dataset was acquired. We also increased the probability of the class that was introduced most recently to reduce forgetting.

The curriculum enhanced learning on the ImageNet benchmark. Learning accelerated and the convergence time decreased by a factor of nearly three. In addition, learning became more robust and classification performance improved substantially. Specifically, top-1 and top-5 test accuracy rose by a factor of 5 to 30.2% and 48.8%, respectively. To our knowledge, this constitutes state-of-the-art performance for reinforcement learning on ImageNet. Hence, representation learning with biologically plausible learning rules can scale to biologically relevant large-scale domains and powerful network architectures.

The curriculum also improved classification performance of networks trained using EBP, although it necessitated a modest number of additional training epochs to converge. This finding is counter-intuitive given that EBP is a supervised learning method and it should therefore not suffer from the class-learning problem that BrainProp is sensitive to. One possible explanation is related to the lottery ticket hypothesis (Mannelli et al., 2024), which states that learning becomes more robust when networks have many parameters so that the efficacy of curricula decreases. Here we used relatively small networks, and the curriculum may have overcome such unfavourable initialisations. This explanation is supported by the BrainProp experiments without a curriculum, demonstrating instabilities and the absence of learning with lower learning rates. However, Mannelli et al. (2024) studied smaller datasets (up to CIFAR-10) and networks in supervised settings. Future work could investigate the effect of curricula on deep neural networks.

The recency bias did not materially affect final model performance, but improved the speed of convergence. However, our implementation of the recency bias was relatively simple and it is possible that using a more elaborate approach (e.g. as in Chen et al., 2024) would also increase the classification accuracy. For instance, one could explore the protection of multiple classes that were recently introduced, decaying their probability based on how long ago they were learned. Other elements could also be added to the curriculum. For instance, a type of consolidation mechanism or other regularisation or rehearsal-based techniques from continual learning and class-incremental learning (Masana et al., 2022) may protect older classes against forgetting. Alternatively, the overarching class structure in ImageNet (based on the WordNet hierarchy as in Wen et al. (2022) or a compositional analysis of network motifs from a trained network such as in Driscoll et al. (2024) could be leveraged to identify – and protect – classes with representational overlap with new classes and might therefore be overwritten. Finally, several studies used difficulty metrics in designing the order of curricula (as reviewed by Soviany et al., 2022). We leave the examination of these other approaches as opportunities for future work.

It is of interest to compare category learning between artificial systems and biological systems. Humans and non-human primates have specific biases during category learning (Smith et al., 2010, 2012), which are qualitatively different from those in other vertebrates (Smith et al., 2012). For example, primates tend to first learn the stereotypical samples of a class, and only later the more exceptional samples (Rubruck et al., 2024; Smith et al., 2010). These biases have also been observed in both deep neural networks trained with supervised algorithms (Kang et al., 2024; Rubruck et al., 2024) and could be exploited by curricula to further enhance learning, by first training on the canonical samples and introducing the non-typical samples later. Additionally, the ordering of examples within classes can be structured in a way to optimise category discovery and generalisation further (Mathy & Feldman, 2009, 2016).

BrainProp is one of several biologically inspired learning rules. Other approaches include predictive coding-based rules (Song et al., 2024; Spratling, 2017), equilibrium propagation (Laborieux & Zenke, 2022), target propagation (Ernoult et al., 2022; Meulemans et al., 2021), feedback alignment (Crafton et al., 2019; Ji-An & Benna, 2024; Launay et al., 2020), e-prop (Bellec et al., 2020), sign-symmetry (Xiao et al., 2018), the forward-forward algorithm (Hinton, 2022), and several other unsupervised (Shen et al., 2023; Talloen et al., 2021) or self-supervised (Halvagal & Zenke, 2023; Illing et al., 2021; Journé et al., 2023; Oquab et al., 2024; M. Ren et al., 2023; Shen et al., 2023; Siddiqui et al., 2023; Sobal et al., 2024; Tang et al., 2022) approaches. Most of these learning rules however do not scale to deeper networks or larger datasets such as ImageNet (Bartunov et al., 2018). The exceptions that do scale to deeper architectures and/or larger data sets such as ImageNet (Aghabarar et al., 2024; Ernoult et al., 2022; Ghaemi et al., 2023; Ji-An & Benna, 2024; Launay et al., 2020; Oquab et al., 2024; Ren et al., 2023; Shen et al., 2023; Siddiqui et al., 2023; Sobal et al., 2024; Xiao et al., 2018), use some type of supervisory signal telling the network what the correct response should be combined with error-backpropagation, for instance in the output layer. This limits their biological plausibility. Alternatively, they use a more biologically plausible approximation of error-backpropagation but then do not scale to deep networks (e.g. Journé et al., 2023). While there are indications that devising scalable self-supervised approaches that are compatible with biological constraints are non-trivial (Weiler et al., 2024; Zenke, n.d.), these alternative approaches offer interesting prospects when combined with the BrainProp framework, which might enhance their overall biological plausibility and augment the overall performance. One interesting avenue for future research could also be to combine a self-supervised pre-training phase to allow the network to learn the statistics of the dataset prior to learning class identities through reinforcement learning curriculum with BrainProp (Cusack et al., 2023). Another possibility would be to add an active learning (Ren et al., 2022) phase subsequent to learning all outcome categories to allow the model to continue learning based on predictions about unlabelled samples that it is sufficiently confident about.

In conclusion, we showed that a curriculum enables BrainProp, a biologically plausible reinforcement learning rule, to train deep networks on the challenging ImageNet benchmark and to achieve state-of-the-art performance across existing biologically plausible learning rules. The curriculum was relatively simple, and we noted many opportunities for expansion and combining it with other approaches. This approach specifically demonstrates how deep networks can be trained with local learning rules and limited feedback, which is of importance for example when training EdgeAI devices in the field. We hope that our work inspires future studies that could achieve even better performance, and generate new insights into the relation between category learning in artificial and biological systems.

Chapter 5
Discussion

This thesis aimed to develop more biologically plausible neural networks with which we can further our understanding of the brain. In this final section, I will consider the general limitations of our approach and the remaining opportunities that lie ahead.

5.1 Bridging the gap

Although this thesis covered several biologically plausible learning rules, the last chapter in particular demonstrates how challenging it can be to learn as effectively as error-backpropagation when working within biological constraints with large-scale networks on complex problems. While the BrainProp algorithm can learn ImageNet through reinforcement learning, supervised learning yields superior performance in terms of speed and accuracy. This phenomenon has been observed for a wide variety of biologically inspired learning rules, which often have difficulties scaling to deeper networks and more complex problem settings (Bartunov et al., 2018). Aside from the fact that these rules tend to approximate the gradient and that this may lead to their inefficiency, their limited applicability to deep networks could also hinder their ability to win the 'lottery ticket hypothesis'. This hypothesis states there is a higher chance for over-parameterised networks of a favourable initialisation that facilitates learning (Mannelli et al., 2024). Alongside scalability concerns, research has also indicated a generalisation gap between non-biologically and biologically plausible learning rules for recurrent neural networks, which could be attributed to the latter being more prone to getting stuck in suboptimal regions in the loss landscape (Liu et al., 2022). As the final chapter showed, curriculum learning can be a useful tool to overcome these limitations and hopefully help to bridge the gap between biological and non-biological learning rules.

5.2 Extending reinforcement learning

The learning rules used and developed in this thesis were all based on reinforcement learning, and specifically, on temporal difference learning. While there is ample empirical support for temporal difference learning (Schultz, 2016; Starkweather & Uchida, 2021; Watabe-Uchida et al., 2017), it is not the only available model for trial-and-error learning as there are different ways of conceptualising the reward prediction error and the role of dopamine in learning (Gershman et al., 2024). Rather than just being recognised for its role in reward-related behaviour, dopamine is becoming appreciated for its involvement in various other functions, such as in sensory prediction, movement, and saliency, in addition to its region-specific effects instead of being considered a universal signal that carries the same information irrespective of the target location (Gershman et al., 2024). Following these empirical results, new models have been created that use more heterogeneous reward prediction signalling through for instance feature-specific errors (Lee et al., 2024) and employing a distributional perspective (Dabney et al., 2020). Another simplification that is often used is that of a static and global learning rate, whereas this may depend on the type of outcome (positive or negative reward prediction error), environmental variability, and rely on different neural mechanisms (Cazé & van der Meer, 2013). Finally, it could be worthwhile to also consider the influence of other neuromodulators than dopamine in decision-making (Doya, 2002), such as serotonin (Colwell et al., 2024; Daw et al., 2002; Grossman et al., 2022; Wert-Carvajal et al., 2022), noradrenaline (Breton-Provencher et al., 2022; Dubois et al., 2021), and acetylcholine (Franklin & Frank, 2015; Sturgill et al., 2020).

5.3 Beyond reinforcement learning

Apart from reinforcement learning itself, there is also great promise in other frameworks. Namely, not all learning is associated with reward or punishment.

One of these alternative frameworks is based on exploiting statistical regularities in the world through self-supervised learning (Jing & Tian, 2021; Liu et al., 2021). Self-supervised learning can be performed by models in multiple ways, such as through reconstructing input data (Tschannen et al., 2018), by acquiring representations that group samples from the same classes together and those from dissimilar classes apart (Jaiswal et al., 2021), or a combination thereof (Wang et al., 2022). A major advantage of this type of approach from a practical standpoint is that it does not rely on labelled datasets (Liu et al., 2021). From a biological perspective, self-supervised learning (and unsupervised learning in general) is also useful. Although class information is not always available, it is possible to infer associations through accumulated experience. For example, trees are more likely found in forests than in the laboratory, while the reverse is true for computers. Therefore, a relationship is suggested between trees and forests, rather than between computers and forests; useful information which can be capitalised upon to structure future learning. While self-supervised learning shows great promise (Liu et al., 2021; Rani et al., 2023) and similarities have been shown between networks trained with this method and neural activity (Bakhtiari et al., 2021; Schaeffer et al., 2023), work remains to be done when it comes to implementing it in a biologically plausible manner since the methods that scale to deeper architectures still tend use error-backpropagation based on a type of supervisory signal (e.g. Oquab et al., 2024; Siddiqui et al., 2023; Sobal et al., 2024).

Another closely related framework that is interesting from a biological perspective is that of predictive coding, which conceptualises the brain as a prediction machine that aims to minimise discrepancies between its predictions and the actual sensory input in the environment coding (Bastos et al., 2012; Huang & Rao, 2011; Rao & Ballard, 1999). To do this, each layer in the network predicts activity in the layer below, and learns through error signals from these lower layers that convey the discrepancy between these predictions and the actual activities (Huang & Rao, 2011), rather than by a supervised error signal propagated from the output layer all the way down to the input layer. Some of the aspects that make predictive coding an appealing theory of brain function is that it provides an explanation for the large amount of feedback processing in the brain, and has been linked to various neurophysiological and psychophysical phenomena (Huang & Rao, 2011; Walsh et al., 2020) – although empirical support is not yet conclusive (Walsh et al., 2020). A practical advantage that was suggested of predictive coding regards the mitigation of catastrophic forgetting associated with error-backpropagation (Song et al., 2024). A number of formalisations of predictive coding exist (e.g. Spratling, 2017), but thus far are limited in scope due to scalability issues (Pinchetti et al., 2024) and the computational cost associated with inference steps (Alonso et al., 2024; Pinchetti et al., 2024; Song et al., 2024; Tscshantz et al., 2023).

Thus, while promising alternatives to reinforcement learning exist, these frameworks are currently limited in scope in terms of biological plausibility and/or scalability.

5.4 The importance of data

The chapters in this thesis also illustrate the impact of training data and training methods. The old adage “Garbage in, garbage out” holds true, of course, but rather than exclusively focusing on the content of training data as this statement might suggest, it is also valuable to consider how content is presented to a network. Consider, for instance, the problem of learning-to-learn that was central in the second chapter. Whereas humans are capable of learning from experience and re-using their knowledge for new tasks, ANNs suffer from catastrophic forgetting

(French, 1999). Meta-learning (as in the second chapter; Duan et al., 2016; Wang et al., 2018) is an effective method for overcoming this, but also requires careful design of inner and outer training loops to successfully generalise learning to similar tasks. Related to this, large scale language models have recently been shown to be capable of solving new problems after exposure to only a few demonstrations – in the absence of new learning (Brown et al., 2020). This type of 'in-context learning' can even amplify itself, thereby allowing these models to meta-in-context learn (Coda-Forno et al., 2023). The advantage of these capabilities is that they provide a more efficient alternative to extensive (and costly) fine-tuning of such models (Brown et al., 2020; Coda-Forno et al., 2023), thereby granting them a certain degree of flexibility such as the one seen in the learning-to-learn problem of the second chapter. Finally, the beneficial effects of curriculum learning in the fourth chapter further underline the large role training distributions can have on the performance of ANNs.

Ultimately, these examples emphasise that we should not neglect the development and utilisation of learning protocols and methods – such as curriculum learning – that prove beneficial for human and animal learning, for (biologically inspired) ANNs might profit from them as well.

5.5 Overall perspective

Both neuroscience and artificial intelligence have experienced great technological progress in recent years, which has allowed computational neuroscience to flourish. As more empirical data is gathered in neuroscience, there is an increasing need of models so that this data can be summarised and explained. Moreover, these models provide the opportunity to test hypotheses that may not be ethically or technologically feasible to study experimentally in humans or animals. While great progress has been made in developing models of the brain, there are still aspects of human intelligence that are out of reach for artificial intelligence. Moreover, a performance gap between error-backpropagation and biologically inspired learning rules in terms of speed, accuracy and generalisability remains to be filled. With the work covered in this thesis, we contributed to these aims by developing a new biologically plausible learning rule for learning-to-learn and working memory that can be extended to deeper networks. Moreover, we designed a curriculum approach that aids training with biologically plausible learning rules so that they can be applied to more complex domains. However, many promising avenues of research in neuroscience-inspired artificial intelligence remain, and more layers of biological plausibility can be added and integrated with previous approaches.

Summary

Biologically Plausible Reinforcement Learning of Deep Cognitive Processing

This thesis was aimed at the development of biologically plausible learning rules. Here, I will summarise each chapter of the thesis and how these contributed towards this goal.

Chapter II: Learning-to-Learn

The second chapter centred on how people can accumulate knowledge over experiences and infer their underlying common structure so they can improve future learning. This process of learning-to-learn poses difficulties, since it requires a model that can flexibly choose which information to remember and forget. Currently existing methods for learning-to-learn are limited in their biological plausibility due to two main reasons: firstly, the complexity of the models makes them challenging to map to the brain, and secondly, the reliance of the models on the backpropagation-through-time algorithm for learning, which is neither local in space, nor time. We developed a new model called RECOLLECT that overcomes both of these limitations. The first limitation was solved by using a simplification of more complicated gated memory models: the Light-GRU. This model uses a single memory gate to determine which information to forget, and when to forget it. We resolved the second limitation by developing a new learning rule based on synaptic tags and traces, which address the spatial and temporal credit assignment problem, respectively. We demonstrated that RECOLLECT can effectively use its memory gates to remember task-relevant information across a temporal delay on a pro-/anti-saccade task. Moreover, RECOLLECT demonstrated strategies similar to those observed in animal studies to navigate this task. Additionally, we found that RECOLLECT can learn-to-learn on a reversal bandit task where reward contingencies associated with actions are periodically altered, requiring the model to flexibly change its response strategy to anticipate these changes. Thus, RECOLLECT is a biologically plausible model of working memory and learning-to-learn.

Chapter III: Deep working memory

The third chapter investigated whether we could extend the RECOLLECT model to create a biologically plausible deep gated memory model. To accomplish this, we applied a truncation mechanism on the synaptic traces, in combination with the BrainProp learning rule. We trained the resulting architecture, Stackollect, on a novel working memory task that requires comparing two stimuli after a temporal delay. This new task was designed to be more perceptually rich than existing working memory tasks, yet simple enough to extrapolate to animal studies. There were two task variants, of which one variant required matching the stimuli on digit identity (0-9) and the other on digit location (four possible positions). We demonstrate that Stackollect can successfully learn to match to both digit identity and location after a delay, and show how the model encodes task-relevant features across memory layers for the different matching goals.

Chapter IV: Curriculum learning for large-scale problems

The fourth chapter focused on enhancing the learning capacity of biologically plausible learning rules by using a curriculum strategy, since these learning rules have been shown to have difficulty scaling to deep networks and complicated datasets. We designed a curriculum to gradually introduce each class of the ImageNet benchmark to a network trained with the BrainProp reinforcement learning algorithm. At each step of the curriculum, we prioritised the learning of the newest class, and prevented forgetting of the most recently introduced class by presenting this class more often as well. We showed that BrainProp can learn on the ImageNet dataset, but that learning is brittle and slow without a curriculum. With the curriculum, BrainProp achieved state-of-the-art performance on ImageNet. Learning was faster, more accurate and more robust. Additionally, the curriculum improved top-1 and top-5 accuracy for error-backpropagation on ImageNet. In conclusion, this is the first approach that demonstrates how biologically plausible reinforcement learning can reliably train deep and complex networks.

Conclusion

Overall, we developed new biologically plausible learning rules for working memory and learning to learn, which can be extended to deeper layers. Furthermore, we designed a new curriculum strategy that helps to bridge the performance gap between biologically plausible learning rules and error-backpropagation on relevant, large-scale datasets.

Samenvatting

Biologisch Plausibel Leren door Beloning voor Diepe Cognitieve Processen

Het doel van dit proefschrift betrof de ontwikkeling van biologisch plausibele leerregels. Deze sectie zal uiteenzetten hoe elk hoofdstuk hieraan heeft bijgedragen.

Hoofdstuk II: Leren-te-leren

In het tweede hoofdstuk werd behandeld hoe mensen kennis vergaren door ervaringen en hoe wij de onderliggende structuur van deze leerprocessen kunnen extraheren om toekomstig leren te faciliteren. Dit proces genaamd 'leren-te-leren' is een uitdaging om te repliceren, aangezien het een model vergt dat flexibel kan selecteren welke informatie herinnerd en vergeten dient te worden. De biologische plausibiliteit van huidige methodes voor leren-te-leren wordt beperkt door twee factoren. Ten eerste zijn deze methodes dermate complex dat het lastig is om ze te relateren aan het brein. Ten tweede hanteren zij het backpropagation-through-time algoritme om te leren, welke niet-lokaal is in ruimte, noch tijd. Wij hebben hierom een nieuw model ontwikkeld, genaamd RECOLLECT. Dit model lost de eerste limitatie op door een simplificatie te gebruiken van een meer complex selectief geheugenmodel: namelijk de Light-GRU. Dit model gebruikt slechts één selectiemechanisme om te beslissen welke informatie vergeten moet worden, en wanneer. Wij hebben de tweede limitatie geadresseerd door een nieuwe leerregel te ontwikkelen gebaseerd op synaptische tags en traces, die respectievelijk het ruimtelijke en temporele krediet toewijzingsprobleem oplossen. We hebben gedemonstreerd hoe RECOLLECT effectief belangrijke taak-gerelateerde informatie kan onthouden gedurende een tijdsspanne op de pro-/anti-saccade taak. Daarnaast ontwikkelde het RECOLLECT-model strategieën die vergelijkbaar waren met strategieën die dieren hanteren tijdens het maken van deze taak. Bovendien was RECOLLECT in staat om te leren-te-leren op een reversal bandit taak, waarop de beloningen die geassocieerd zijn met bepaalde acties periodiek worden gewijzigd en het model flexibel van strategie moet wisselen om deze veranderingen te kunnen anticiperen. We concluderen dat RECOLLECT een biologisch plausibel model van werkgeheugen en leren-te-leren is.

Hoofdstuk III: Diepe geheugenmodellen

Het derde hoofdstuk onderzocht of RECOLLECT gegeneraliseerd kon worden naar een biologisch plausibel, diep geheugenmodel. Om dit te bewerkstellingen, hebben wij de synaptische traces ingekort en de BrainProp leerregel ingezet. We hebben het resulterende model, Stackollect, op een nieuwe werkgeheugentaak getraind, waarop twee stimuli met elkaar vergeleken moeten worden na een pauze. Deze nieuwe taak was ontwikkeld zodat meer complexe perceptuele input ingezet kon worden dan welke beschikbaar is in huidige paradigma's, maar tegelijkertijd simpel genoeg was om te extrapoleren naar dierenexperimenten. Er waren twee taakvarianten, waarvan het voor één variant belangrijk was om te vergelijken of twee stimuli hetzelfde getal (0-9) betroffen (cijfer matching), en voor de andere variant of deze getallen op dezelfde locatie op het scherm verschenen (vier mogelijke locaties; locatie matching). Wij demonstreren dat Stackollect in staat is om beide taakvarianten te leren. Daarnaast laten wij zien hoe de selectiviteit voor taak-relevante variabelen is verdeeld over de verschillende geheugenlagen, afhankelijk van de taakvariant.

Hoofdstuk IV: Curriculum leren voor grootschalige problemen

Het vierde hoofdstuk richtte zich op het versterken van het leervermogen van biologisch plausibele leerregels door een curriculum strategie te hanteren, aangezien deze leerregels lastig te schalen blijken te zijn naar diepere netwerken en meer complexe datasets. Wij hebben een curriculum ontwikkeld dat geleidelijk elke categorie van de ImageNet dataset introduceert aan een netwerk dat leert aan de hand van beloningen met het BrainProp algoritme. Bij elke stap van het curriculum werd het leren van de allernieuwste categorie het sterkst gefaciliteerd door deze onevenredig vaak aan te bieden aan het netwerk. Daarnaast werd de geheugenrepresentatie van de een-na-laatste categorie beschermd door deze ook vaker te presenteren dan andere categorieën. We hebben gedemonstreerd dat BrainProp netwerken succesvol kan trainen op ImageNet, maar dat het leerproces fragiel en traag is zonder curriculum. Mét curriculum was de prestatie van BrainProp op deze dataset state-of-the-art. Het curriculum maakte het leerproces sneller, accurater en robuuster. Het curriculum verbeterde zelfs de top-1 en top-5 accuratesse voor het error-backpropagation algoritme. Concluderend is dit de eerste methode die demonstreert hoe biologisch plausibele leerregels op basis van beloning op betrouwbare wijze diepe en complexe netwerken kunnen trainen.

Conclusie

Samengevat hebben wij nieuwe biologisch plausibele leerregels ontwikkeld voor werkgeheugen en het 'leren-te-leren' proces die generaliseerd kunnen worden naar diepere architecturen. Daarnaast hebben wij een curriculum strategie ontwikkeld die het verschil in capaciteit tussen biologisch plausibele leerregels en error-backpropagation kan overbruggen op relevante, grootschalige datasets.

List of publications

Chapters of this thesis have previously been published. This overview details which chapters are featured in which publications:

- Chapter 2 is based on:

  van den Berg, A. R., Roelfsema, P. R., & Bohte, S. M. (2024). Biologically plausible gated recurrent neural networks for working memory and learning-to-learn. *PloS ONE, 19*(12), e0316453.

- Chapter 3 is new.

- Chapter 4 to appear as:

  Alexandra R. van den Berg, P. R. Roelfsema, & Sander M. Bohté. (2025). Curriculum Design for Scalable Biologically Plausible Deep Reinforcement Learning. *2025 International Joint Conference on Neural Networks (IJCNN).*

Acknowledgements

None of this would have been remotely possible without the incredible support of my family, friends and supervisors.

First and foremost, I would like to thank my parents **Johan** and **Lucy**, as well as my brother **Jur**. Thank you for always having my back. The Ph.D. is a goal I had been set on for a long time and without your incredible support all throughout life I would not have become half the scientist, and person, that I am now. You were always there for me and ready to offer advice, a thuisfront retreat, instrumental support or a listening ear. There probably have been few conversations in the last five years that did not feature the Ph.D. at least at some point and that must have been tedious at best at times, but you never once complained. A special shout-out to my mom for our daily phone calls after work and allowing me to rant about the good as much as the bad, as well as for my dad for always reminding me to take things easier and to take a step back so I can consider things in a more positive light. Also big thanks to my brother, for readily offering IT support to help me tackle matters such as LaTeX hell, supercomputer shenanigans or 3D printing the beautiful bee bookmarks, but also for our pizza parties whenever I ended up in Ph.D. crisis. It never failed to cheer me up. I am incredibly grateful to have you three as my family and I love you very, very much.

Next, I would like to thank my fellow turtles **Medina** and **Miriam**. We have been together ever since that fateful first day of the neuro master. It has been a wild ride since then, with all of us on our own Ph.D. trajectories. Thank you for all the laughs, experiences, science talks and adventures both within the country as well as beyond. I would not have missed it for the world. Thanks to Medina for the frequent reality checks, fitgirl talks and hair progress updates, and showing me how to break out of my comfort zone. Thanks to Miriam for inspiring me with your perseverance and zest for life, the fun post-contract unemployment hangouts, and always offering a listening ear. It will be tough once both of you move away to start your next adventures in Australia/Germany/Switzerland, and I am going to miss you a lot. But, you are not nearly rid of me yet.

Next, I would like to thank my paranymphs **Sami** and **Parva**. Sami, we started this journey together and I am glad we are still here to see it through until the end (*knock knock* [1]). Doing a collab and working at different groups/locations can

[1] I'll knock until the end (*knock knock*).

be tricky, and I was very grateful to have you by my side. From project musings to conference planning, from general venting sessions to tea spilling, and from strange Corona start-up times to the re-emergence of the status quo, you were a source of stability among volatility and I highly appreciate you for being along for the ride. As for Parva, we initially bonded over our mutual hatred of mushrooms; a life-long nemesis. While I have slowly – against all expectations, and admittedly, efforts[2] – been growing out of my rancour, our friendship remained. Thank you for all the long life talks and hangouts. It is sad we did not end up sharing many working days at CWI itself, but we will definitely keep meeting up.

I would also like to thank **Valerie**. We may have not known each other extremely long yet, but in short time you have become a very dear friend to me. I greatly enjoy our – very – lengthy text chains ranging anywhere from fashion advice, pop culture updates and life events to deep psychological analyses. Whenever I have a problem, I can count on you for your empathy and problem-solving skills. With your frequent changes of hobbies and interests, you open my world up to lots of new things and I enjoy experiencing it vicariously.

Next, thank you to **Mouraya** for all our (science) talks and for being the first of my friends to pave the way for both obtaining the Ph.D. as well as navigating life outside academia. Thank you to **Rick** for the many years of friendship. You have known me at my cringiest (and I you at yours), and I am happy we are still in touch. Thank you to **Natalia** for all the fun stories about your many colourful adventures and your help with job applications. Thank you to **Burcu** for our cultural expeditions and (Ph.D.) life talks. Thank you to **Anne-Clarine** for your friendship even past the dark Socrates days. You have been privy to all my personal drama, and I yours, but we still rise beyond. Thank you for all our meet-ups and showing me life beyond science.

I would also like to thank everyone at the **Vision & Cognition group at the NIN** and the Machine Learning group at the CWI for the fun lab retreats, project feedback, talks and inspiring me with your projects. Thank you to the **CWI Activity Committee** as well, for all the fun events we organised together.

Moreover, thank you to **Iris Groen**, **Marcel van Gerven**, **Cyriel Pennartz**, **Steven Scholte**, and **Dileep George** for being on my committee.

Additionally, a shout-out to Malvin Gattinger, whom I have never met, but whose Latex template for this thesis was available on Github. Thank you for saving me from that headache.

[2] I'm looking at you, Jur.

Thank you for all the reviewers who received my papers, both positive and negative. Special shout-out to **reviewer #9** for a poster submission of RECOLLECT at a conference, which culminated in this gem:

> *"The acronym they invented to name their model is a stupid monstrosity. It's nice to name things, but the letters have nothing to do with the first letters of its full-worded description, let alone its most important syllables. Let's not abbreviate models things with commonly used cognitive terms either. It's confusing, farcical, and makes a mockery out of real and meaningful abbreviations in the field. Heck, name it after yourself if you are so vain and driven to name things, better yet, brainstorm a smarter acronym, or let other researchers find shorthands for you, which will happen anyways if and when they find it meaningful to refer to."*

I am unsure who hurt you #9[3], but thank you for making my day – to this day[4].

Also thank you to my new manager **Sander**[5] and daily supervisor **Stephanie** at the **AFM** for giving me this new opportunity after the Ph.D. Thank you for the guidance and support you have already given me so far, and thank you to the entirety of **team PI** for receiving me with open arms. I really enjoy working here so far, and I am looking forward to continuing collaborating with you.

Finally, very importantly, thank you to my supervisors **Pieter Roelfsema** and **Sander Bohté** for supervising me during the Ph.D. Thank you for giving me this opportunity and for maintaining your faith in me, even during the times when my faith in myself faltered. Thank you to Pieter for all the valuable feedback on my papers and projects, and for always making time in your busy schedule for Sami and me for the DeepBrain meetings. Thank you to Sander also for all the feedback of course, but mainly for always being there to support me. I really appreciate the open-door policy you maintained throughout the journey and for always having my back, ranging from advocating for more supercomputer budget to doing everything you could to make sure I was able to graduate with as little delay as possible. Whenever I lost hope, you were able to find a solution or reassured me with encouragements. I would not have managed without you.

[3] Perhaps a psychological effect of multiplying reviewer #3 with 3?

[4] It sparks joy whenever I present RECOLLECT. The only sad thing is that I'll never know how you would respond to Stackollect. Reach out to me anonymously in the unlikely event you see this?

[5] And thereby continuing the Sanne-Sander supervision legacy. If you're confused, read the next paragraph.

Bibliography

Aghabarar, H., Keshavarzi, P., & Kiani, K. (2024). Reinforcement Learning in Deep Spiking Neural Networks with Eligibility Traces and Modifying the Threshold Parameter. https://doi.org/10.21203/rs.3.rs-3830542/v1.

Alonso, N., Krichmar, J., & Neftci, E. (2024). Understanding and Improving Optimization in Predictive Coding Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, *38*(10), 10812–10820. https://doi.org/10.1609/aaai.v38i10.28954.

Baddeley, A. (1992). Working Memory. *Science*, *255*(5044), 556–559. https://doi.org/10.1126/science.1736359.

Baddeley, A. (2010). Working memory. *Current Biology*, *20*(4), R136–R140. https://doi.org/10.1016/j.cub.2009.12.014.

Bae, J. W., Jeong, H., Yoon, Y. J., Bae, C. M., Lee, H., Paik, S.-B., & Jung, M. W. (2021). Parallel processing of working memory and temporal information by distinct types of cortical projection neurons. *Nature Communications*, *12*(1), 4352. https://doi.org/10.1038/s41467-021-24565-z.

Bandettini, P. A. (2012). Twenty years of functional MRI: The science and the stories. *NeuroImage*, *62*(2), 575–588. https://doi.org/10.1016/j.neuroimage.2012.04.026.

Barbas, H., Wang, J., Joyce, M. K. P., & García-Cabezas, M. Á. (2018). Pathway mechanism for excitatory and inhibitory control in working memory. *Journal of Neurophysiology*, *120*(5), 2659–2678. https://doi.org/10.1152/jn.00936.2017.

Bartunov, S., Santoro, A., Richards, B., Marris, L., Hinton, G. E., & Lillicrap, T. (2018). Assessing the Scalability of Biologically-Motivated Deep Learning Algorithms and Architectures. *Advances in Neural Information Processing Systems*, *31*.

Bastos, A. M., Usrey, W. M., Adams, R. A., Mangun, G. R., Fries, P., & Friston, K. J. (2012). Canonical Microcircuits for Predictive Coding. *Neuron*, *76*(4), 695–711. https://doi.org/10.1016/j.neuron.2012.10.038.

Bays, P. M., Schneegans, S., Ma, W. J., & Brady, T. F. (2024). Representation and computation in visual working memory. *Nature Human Behaviour*, *8*(6), 1016–1034. https://doi.org/10.1038/s41562-024-01871-2.

Beck, M., Pöppel, K., Spanring, M., Auer, A., Prudnikova, O., Kopp, M., Klambauer, G., Brandstetter, J., & Hochreiter, S. (2024). xLSTM: Extended long short-term memory. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, & C. Zhang (Editors), *Advances in neural information processing systems* (Pages 107547–107603, Volume 37). Curran Associates, Inc.

Bellec, G., Salaj, D., Subramoney, A., Legenstein, R., & Maass, W. (2018). Long short-term memory and Learning-to-learn in networks of spiking neurons. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Editors), *Advances in neural information processing systems* (Volume 31). Curran Associates, Inc.

Bellec, G., Scherr, F., Hajek, E., Salaj, D., Legenstein, R., & Maass, W. (2019). *Biologically inspired alternatives to backpropagation through time for learning in recurrent neural nets*. arXiv: 1901.09049 `[cs]`.

Bellec, G., Scherr, F., Subramoney, A., Hajek, E., Salaj, D., Legenstein, R., & Maass, W. (2020). A solution to the learning dilemma for recurrent networks of spiking neurons. *Nature Communications*, *11*(1), 3625. https://doi.org/10.1038/s41467-020-17236-y.

Bellemare, M. G., Dabney, W., & Munos, R. (2017). A Distributional Perspective on Reinforcement Learning. *Proceedings of the 34th International Conference on Machine Learning*, *70*, 449–458.

Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009). Curriculum learning. *Proceedings of the 26th Annual International Conference on Machine Learning*, 41–48. https://doi.org/10.1145/1553374.1553380.

Bilal, A., Jourabloo, A., Ye, M., Liu, X., & Ren, L. (2018). Do Convolutional Neural Networks Learn Class Hierarchy? *IEEE Transactions on Visualization and Computer Graphics*, *24*(1), 152–162. https://doi.org/10.1109/TVCG.2017.2744683.

Bolkan, S. S., Stujenske, J. M., Parnaudeau, S., Spellman, T. J., Rauffenbart, C., Abbas, A. I., Harris, A. Z., Gordon, J. A., & Kellendonk, C. (2017). Thalamic projections sustain prefrontal activity during working memory maintenance. *Nature Neuroscience*, *20*(7), 987–996. https://doi.org/10.1038/nn.4568.

Box, G. E. P., Hunter, J. S., & Hunter, W. G. (1978). *Statistics for experimenters: An introduction to design, data analysis, and model building*. Wiley.

Breton-Provencher, V., Drummond, G. T., Feng, J., Li, Y., & Sur, M. (2022). Spatiotemporal dynamics of noradrenaline during learned behaviour. *Nature*, *606*(7915), 732–738. https://doi.org/10.1038/s41586-022-04782-2.

Brissenden, J. A., Tobyne, S. M., Halko, M. A., & Somers, D. C. (2021). Stimulus-specific visual working memory representations in human cerebellar lobule viib/viiia. *Journal of Neuroscience*, *41*(5), 1033–1045. https://doi.org/10.1523/JNEUROSCI.1253-20.2020.

Broadway, M. (2022). Delayed match-to-sample. In J. Vonk & T. K. Shackelford (Editors), *Encyclopedia of animal cognition and behavior* (Pages 1977–1979).

Springer International Publishing. https://doi.org/10.1007/978-3-319-55065-7_1736.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). Language Models are Few-Shot Learners. *Advances in neural information processing systems*, *33*, 1877–1901.

Brunswik, E. (1939). Probability as a determiner of rat behavior. *Journal of Experimental Psychology*, *25*(2), 175–197. https://doi.org/10.1037/h0061204.

Cai, X., & Padoa-Schioppa, C. (2014). Contributions of Orbitofrontal and Lateral Prefrontal Cortices to Economic Choice and the Good-to-Action Transformation. *Neuron*, *81*(5), 1140–1151. https://doi.org/10.1016/j.neuron.2014.01.008.

Carley, K. M. (1999). On generating hypotheses using computer simulations. *Systems Engineering*, *2*(2), 69–77.

Carpenter, G. A., & Grossberg, S. (1987). ART 2: Self-organization of stable category recognition codes for analog input patterns. *Applied Optics*, *26*(23), 4919–4930.

Cazé, R. D., & van der Meer, M. A. A. (2013). Adaptive properties of differential learning rates for positive and negative outcomes. *Biological Cybernetics*, *107*(6), 711–719. https://doi.org/10.1007/s00422-013-0571-5.

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P. S., Yang, Q., & Xie, X. (2024). A Survey on Evaluation of Large Language Models. *ACM Transactions on Intelligent Systems and Technology*, *15*(3), 1–45. https://doi.org/10.1145/3641289.

Chen, S., Zhang, M., Zhang, J., & Huang, K. (2024). Exemplar-based Continual Learning via Contrastive Learning. *IEEE Transactions on Artificial Intelligence*, *5*, 3313–3324. https://doi.org/10.1109/TAI.2024.3355879.

Chen, X., Wang, F., Fernandez, E., & Roelfsema, P. R. (2020). Shape perception via a high-channel-count neuroprosthesis in monkey visual cortex. *Science*, *370*(6521), 1191–1196. https://doi.org/10.1126/science.abd7435.

Cho, K., van Merrienboer, B., Bahdanau, D., & Bengio, Y. (2014). *On the Properties of Neural Machine Translation: Encoder-Decoder Approaches*. arXiv: 1409.1259 [cs]. https://doi.org/10.48550/arXiv.1409.1259.

Choi, J., Kim, T., & Lee, S.-g. (2019, November 17–19). Cell-aware stacked LSTMs for modeling sentences. In W. S. Lee & T. Suzuki (Editors), *Proceedings of The Eleventh Asian Conference on Machine Learning* (Pages 1172–1187, Volume 101). PMLR.

Christophel, T. B., Klink, P. C., Spitzer, B., Roelfsema, P. R., & Haynes, J.-D. (2017). The Distributed Nature of Working Memory. *Trends in Cognitive Sciences*, *21*(2), 111–124. https://doi.org/10.1016/j.tics.2016.12.007.

Coda-Forno, J., Binz, M., Akata, Z., Botvinick, M., Wang, J. X., & Schulz, E. (2023). Meta-in-context learning in large language models. *Advances in Neural Information Processing Systems*, *36*, 65189–65201.

Colwell, M. J., Tagomori, H., Shang, F., Cheng, H. I., Wigg, C. E., Browning, M., Cowen, P. J., Murphy, S. E., & Harmer, C. J. (2024). Direct serotonin release in humans shapes aversive learning and inhibition. *Nature Communications*, *15*(1), 6617. https://doi.org/10.1038/s41467-024-50394-x.

Costa, R., Assael, I. A., Shillingford, B., de Freitas, N., & Vogels, T. (2017). Cortical microcircuits as gated-recurrent neural networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Editors), *Advances in neural information processing systems* (Volume 30). Curran Associates, Inc.

Crafton, B., Parihar, A., Gebhardt, E., & Raychowdhury, A. (2019). Direct Feedback Alignment With Sparse Connections for Local Learning. *Frontiers in Neuroscience*, *13*, 525. https://doi.org/10.3389/fnins.2019.00525.

Cunningham, P., Cord, M., & Delany, S. J. (2008). Supervised Learning. In M. Cord & P. Cunningham (Editors), *Machine Learning Techniques for Multimedia* (Pages 21–49). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-75171-7_2.

Curtis, C. E., & Sprague, T. C. (2021). Persistent Activity During Working Memory From Front to Back. *Frontiers in Neural Circuits*, *15*. https://doi.org/10.3389/fncir.2021.696060.

Cusack, R., Ranzato, M., & Charvet, C. J. (2024). Helpless infants are learning a foundation model. *Trends in Cognitive Sciences*, *28*(8), 726–738. https://doi.org/10.1016/j.tics.2024.05.001.

Dabney, W., Kurth-Nelson, Z., Uchida, N., Starkweather, C. K., Hassabis, D., Munos, R., & Botvinick, M. (2020). A distributional code for value in dopamine-based reinforcement learning. *Nature*, *577*(7792), 671–675. https://doi.org/10.1038/s41586-019-1924-6.

Dang, W., Jaffe, R. J., Qi, X.-L., & Constantinidis, C. (2021). Emergence of Nonlinear Mixed Selectivity in Prefrontal Cortex after Training. *Journal of Neuroscience*, *41*(35), 7420–7434. https://doi.org/10.1523/JNEUROSCI.2814-20.2021.

Daniel, T. A., Katz, J. S., & Robinson, J. L. (2016). Delayed match-to-sample in working memory: A BrainMap meta-analysis. *Biological Psychology*, *120*, 10–20. https://doi.org/10.1016/j.biopsycho.2016.07.015.

Daw, N. D., Kakade, S., & Dayan, P. (2002). Opponent interactions between serotonin and dopamine. *Neural Networks*, *15*(4-6), 603–616. https://doi.org/10.1016/S0893-6080(02)00052-7.

Dayan, P., & Balleine, B. W. (2002). Reward, Motivation, and Reinforcement Learning. *Neuron*, *36*(2), 285–298. https://doi.org/10.1016/S0896-6273(02)00963-7.

De Zeeuw, C. I., Lisberger, S. G., & Raymond, J. L. (2021). Diversity and dynamism in the cerebellum. *Nature Neuroscience*, *24*(2), 160–167. https://doi.org/10.1038/s41593-020-00754-9.

Dehghani, N., & Wimmer, R. D. (2019). A Computational Perspective of the Role of the Thalamus in Cognition. *Neural Computation*, *31*(7), 1380–1418. https://doi.org/10.1162/neco_a_01197.

Dekker, R. B., Otto, F., & Summerfield, C. (2022). Curriculum learning for human compositional generalization. *Proceedings of the National Academy of Sciences*, *119*(41), e2205582119. https://doi.org/10.1073/pnas.2205582119.

D'Esposito, M., & Postle, B. R. (2015). The cognitive neuroscience of working memory. *Annual Review of Psychology*, *66*(1), 115–142. https://doi.org/10.1146/annurev-psych-010814-015031.

Dey, R., & Salem, F. M. (2017). Gate-variants of Gated Recurrent Unit (GRU) neural networks. *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 1597–1600. https://doi.org/10.1109/MWSCAS.2017.8053243.

Driscoll, L. N., Shenoy, K., & Sussillo, D. (2024). Flexible multitask computation in recurrent networks utilizes shared dynamical motifs. *Nature Neuroscience*, *27*(7), 1349–1363. https://doi.org/10.1038/s41593-024-01668-6.

Duan, Y., Schulman, J., Chen, X., Bartlett, P. L., Sutskever, I., & Abbeel, P. (2016). *RL^2: Fast Reinforcement Learning via Slow Reinforcement Learning.*

Dubois, M., Habicht, J., Michely, J., Moran, R., Dolan, R. J., & Hauser, T. U. (2021). Human complex exploration strategies are enriched by noradrenaline-modulated heuristics (T. Kahnt, C. Büchel, & C. Warren, Editors). *eLife*, *10*, e59907. https://doi.org/10.7554/eLife.59907.

Dugué, G. P., Akemann, W., & Knöpfel, T. (2012). Chapter 1 - A comprehensive concept of optogenetics. In T. Knöpfel & E. S. Boyden (Editors), *Progress in Brain Research* (Pages 1–28, Volume 196). Elsevier. https://doi.org/10.1016/B978-0-444-59426-6.00001-X.

Dujmović, M., Malhotra, G., & Bowers, J. S. (2020). What do adversarial images tell us about human vision? (G. J. Berman, R. L. Calabrese, & C. Firestone, Editors). *eLife*, *9*, e55978. https://doi.org/10.7554/eLife.55978.

Eadie, L., & Ghosh, T. K. (2011). Biomimicry in textiles: Past, present and potential. An overview. *Journal of The Royal Society Interface*, *8*(59), 761–775. https://doi.org/10.1098/rsif.2010.0487.

Empson, J. (1986). The history and origin of the EEG. In *Human brainwaves: The psychological significance of the electroencephalogram* (Pages 1–20). Macmillan Education UK. https://doi.org/10.1007/978-1-349-18312-8_1.

Ernoult, M. M., Normandin, F., Moudgil, A., Spinney, S., Belilovsky, E., Rish, I., Richards, B., & Bengio, Y. (2022). Towards Scaling Difference Target Propagation by Learning Backprop Targets. *Proceedings of the 39th International Conference on Machine Learning*, *162*, 5968–5987.

Feynman, R. (2018). *Feynman Lectures On Gravitation*. CRC Press. https://doi.org/10.1201/9780429502859.

Fini, M., & Tyler, W. J. (2017). Transcranial focused ultrasound: A new tool for non-invasive neuromodulation. *International Review of Psychiatry*, *29*(2), 168–177. https://doi.org/10.1080/09540261.2017.1302924.

Frank, M. J., Loughry, B., & O'Reilly, R. C. (2001). Interactions between frontal cortex and basal ganglia in working memory: A computational model. *Cognitive, Affective, & Behavioral Neuroscience*, *1*(2), 137–160. https://doi.org/10.3758/CABN.1.2.137.

Franklin, N. T., & Frank, M. J. (2015). A cholinergic feedback circuit to regulate striatal population uncertainty and optimize reinforcement learning (U. S. Bhalla, Editor). *eLife*, *4*, e12029. https://doi.org/10.7554/eLife.12029.

French, R. M. (1999). Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, *3*(4), 128–135. https://doi.org/10.1016/S1364-6613(99)01294-2.

Fujii, N., & Graybiel, A. M. (2003). Representation of action sequence boundaries by macaque prefrontal cortical neurons. *Science (New York, N.Y.)*, *301*(5637), 1246–1249. https://doi.org/10.1126/science.1086872.

Fuster, J. M. (1997). Network memory. *Trends in Neurosciences*, *20*(10), 451–459. https://doi.org/10.1016/S0166-2236(97)01128-4.

Gao, Z., Davis, C., Thomas, A. M., Economo, M. N., Abrego, A. M., Svoboda, K., De Zeeuw, C. I., & Li, N. (2018). A cortico-cerebellar loop for motor planning. *Nature*, *563*(7729), 113–116. https://doi.org/10.1038/s41586-018-0633-x.

Gershman, S. J., Assad, J. A., Datta, S. R., Linderman, S. W., Sabatini, B. L., Uchida, N., & Wilbrecht, L. (2024). Explaining dopamine through prediction errors and beyond. *Nature Neuroscience*, *27*(9), 1645–1655. https://doi.org/10.1038/s41593-024-01705-4.

Gerstner, W., Lehmann, M., Liakoni, V., Corneil, D., & Brea, J. (2018). Eligibility Traces and Plasticity on Behavioral Time Scales: Experimental Support of NeoHebbian Three-Factor Learning Rules. *Frontiers in Neural Circuits*, *12*, 53. https://doi.org/10.3389/fncir.2018.00053.

Ghaemi, H., Mirzaei, E., Nouri, M., & Kheradpisheh, S. R. (2023). BioLCNet: Reward-Modulated Locally Connected Spiking Neural Networks. In G. Nicosia, V. Ojha, E. La Malfa, G. La Malfa, P. Pardalos, G. Di Fatta, G. Giuffrida, & R. Umeton (Editors), *International Conference on Machine Learning, Optimization, and Data Science* (Pages 564–578). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-25891-6_42.

Gnadt, J. W., & Andersen, R. A. (1988). Memory related motor planning activity in posterior parietal cortex of macaque. *Exp Brain Res*, *70*(1), 216–220. https://doi.org/10.1007/BF00271862.

Gottlieb, J., & Goldberg, M. E. (1999). Activity of neurons in the lateral intraparietal area of the monkey during an antisaccade task. *Nature Neuroscience*, *2*(10), 906–912. https://doi.org/10.1038/13209.

GPT-4 Technical Report. (2024). arXiv: 2303.08774. https://doi.org/10.48550/arXiv.2303.08774.

Graves, A., Mohamed, A.-r., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 6645–6649. https://doi.org/10.1109/ICASSP.2013.6638947.

Grossman, C. D., Bari, B. A., & Cohen, J. Y. (2022). Serotonin neurons modulate learning rate through uncertainty. *Current Biology*, *32*(3), 586–599. https://doi.org/10.1016/j.cub.2021.12.006.

Gui, J., Chen, T., Zhang, J., Cao, Q., Sun, Z., Luo, H., & Tao, D. (2024). A Survey on Self-supervised Learning: Algorithms, Applications, and Future Trends. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *46*(12), 9052–9071. https://doi.org/10.1109/TPAMI.2024.3415112.

Halvagal, M. S., & Zenke, F. (2023). The combination of Hebbian and predictive plasticity learns invariant object representations in deep sensory networks. *Nature Neuroscience*, *26*(11), 1906–1915. https://doi.org/10.1038/s41593-023-01460-y.

Hao, Y., Ge, H., Sun, M., & Gao, Y. (2019). Selecting an Appropriate Animal Model of Depression. *International Journal of Molecular Sciences*, *20*(19), 4827. https://doi.org/10.3390/ijms20194827.

Harlow, H. F. (1949). The formation of learning sets. *Psychological Review*, *56*(1), 51–65. https://doi.org/10.1037/h0062474.

Hassabis, D., Kumaran, D., Summerfield, C., & Botvinick, M. (2017). Neuroscience-Inspired Artificial Intelligence. *Neuron*, *95*(2), 245–258. https://doi.org/10.1016/j.neuron.2017.06.011.

He, M. F., Schultz, B. D., & Schubert, W. H. (2015). *The SAGE Guide to Curriculum in Education.* SAGE Publications.

Hedayati, S., O'Donnell, R. E., & Wyble, B. (2022). A model of working memory for latent representations. *Nature Human Behaviour*, *6*(5), 709–719. https://doi.org/10.1038/s41562-021-01264-9.

Hermans, M., & Schrauwen, B. (2013). Training and Analysing Deep Recurrent Neural Networks. *Advances in Neural Information Processing Systems*, *26*.

Hikosaka, O., Kim, H. F., Yasuda, M., & Yamamoto, S. (2014). Basal Ganglia Circuits for Reward Value–Guided Behavior. *Annual Review of Neuroscience*, *37*(1), 289–306. https://doi.org/10.1146/annurev-neuro-071013-013924.

Himmer, L., Schönauer, M., Heib, D. P. J., Schabus, M., & Gais, S. (2019). Rehearsal initiates systems memory consolidation, sleep makes it last. *Science Advances*, *5*(4), eaav1695. https://doi.org/10.1126/sciadv.aav1695.

Hinton, G. (2022). *The Forward-Forward Algorithm: Some Preliminary Investigations.* arXiv: 2212.13345 [cs.LG]. https://arxiv.org/abs/2212.13345.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

Hocker, D., Constantinople, C. M., & Savin, C. (2024). *Curriculum learning inspired by behavioral shaping trains neural networks to adopt animal-like decision making strategies*. 2024.01.12.575461. https://doi.org/10.1101/2024.01.12.575461.

Hodgkin, A. L., & Huxley, A. F. (1939). Action Potentials Recorded from Inside a Nerve Fibre. *Nature*, *144*(3651), 710–711. https://doi.org/10.1038/144710a0.

Hoy, C. W., Quiroga-Martinez, D. R., Sandoval, E., King-Stephens, D., Laxer, K. D., Weber, P., Lin, J. J., & Knight, R. T. (2023). Asymmetric coding of reward prediction errors in human insula and dorsomedial prefrontal cortex. *Nature Communications*, *14*(1), 8520. https://doi.org/10.1038/s41467-023-44248-1.

Huang, Y., & Rao, R. P. N. (2011). Predictive coding. *WIREs Cognitive Science*, *2*(5), 580–593. https://doi.org/10.1002/wcs.142.

Huisman, M., van Rijn, J. N., & Plaat, A. (2021). A survey of deep meta-learning. *Artificial Intelligence Review*, *54*(6), 4483–4541. https://doi.org/10.1007/s10462-021-10004-4.

Illing, B., Ventura, J., Bellec, G., & Gerstner, W. (2021). Local plasticity rules can learn deep representations using self-supervised contrastive predictions. *Advances in Neural Information Processing Systems*, *34*, 30365–30379.

Ito, M., & Doya, K. (2009). Validation of Decision-Making Models and Analysis of Decision Variables in the Rat Basal Ganglia. *Journal of Neuroscience*, *29*(31), 9861–9874. https://doi.org/10.1523/JNEUROSCI.6157-08.2009.

Izquierdo, A., Brigman, J. L., Radke, A. K., Rudebeck, P. H., & Holmes, A. (2017). The neural basis of reversal learning: An updated perspective. *Neuroscience*, *345*, 12–26. https://doi.org/10.1016/j.neuroscience.2016.03.021.

Ji-An, L., & Benna, M. K. (2024). *Deep Learning without Weight Symmetry*. arXiv: 2405.20594 `[cs.LG]`. https://arxiv.org/abs/2405.20594.

Journé, A., Rodriguez, H. G., Guo, Q., & Moraitis, T. (2023). *Hebbian Deep Learning Without Feedback*. arXiv: 2209.11883 `[cs.NE]`. https://arxiv.org/abs/2209.11883.

Jozefowicz, R., Zaremba, W., & Sutskever, I. (2015). An Empirical Exploration of Recurrent Network Architectures. *Proceedings of the 32nd International Conference on Machine Learning*, *37*, 2342–2350.

Jun, J. J., Steinmetz, N. A., Siegle, J. H., Denman, D. J., Bauza, M., Barbarits, B., Lee, A. K., Anastassiou, C. A., Andrei, A., Aydın, Ç., Barbic, M., Blanche, T. J., Bonin, V., Couto, J., Dutta, B., Gratiy, S. L., Gutnisky, D. A., Häusser, M., Karsh, B., … Harris, T. D. (2017). Fully integrated silicon probes for high-density recording of neural activity. *Nature*, *551*(7679), 232–236. https://doi.org/10.1038/nature24636.

Kakade, S., & Dayan, P. (2002). Dopamine: Generalization and bonuses. *Neural Networks*, *15*(4-6), 549–559. https://doi.org/10.1016/S0893-6080(02)00048-5.

Kalin, N. H. (2021). Understanding the Value and Limitations of MRI Neuroimaging in Psychiatry. *American Journal of Psychiatry*, *178*(8), 673–676. https://doi.org/10.1176/appi.ajp.2021.21060616.

Kang, K., Setlur, A., Tomlin, C., & Levine, S. (2024). *Deep Neural Networks Tend To Extrapolate Predictably*. arXiv: 2310.00873 `[cs.LG]`. https://arxiv.org/abs/2310.00873.

Kawai, T., Yamada, H., Sato, N., Takada, M., & Matsumoto, M. (2015). Roles of the Lateral Habenula and Anterior Cingulate Cortex in Negative Outcome Monitoring and Behavioral Adjustment in Nonhuman Primates. *Neuron*, *88*(4), 792–804. https://doi.org/10.1016/j.neuron.2015.09.030.

Keller, G. B., & Mrsic-Flogel, T. D. (2018). Predictive Processing: A Canonical Cortical Computation. *Neuron*, *100*(2), 424–435. https://doi.org/10.1016/j.neuron.2018.10.003.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, *25*.

Kruijne, W., Bohte, S. M., Roelfsema, P. R., & Olivers, C. N. L. (2021). Flexible Working Memory Through Selective Gating and Attentional Tagging. *Neural Computation*, *33*(1), 1–40. https://doi.org/10.1162/neco_a_01339.

Kruse, R., Mostaghim, S., Borgelt, C., Braune, C., & Steinbrecher, M. (2022). Multi-layer Perceptrons. In R. Kruse, S. Mostaghim, C. Borgelt, C. Braune, & M. Steinbrecher (Editors), *Computational Intelligence: A Methodological Introduction* (Pages 53–124). Springer International Publishing. https://doi.org/10.1007/978-3-030-42227-1_5.

Laborieux, A., & Zenke, F. (2022). Holomorphic Equilibrium Propagation Computes Exact Gradients Through Finite Size Oscillations. *Advances in neural information processing systems*, *35*, 12950–12963.

Launay, J., Poli, I., Boniface, F., & Krzakala, F. (2020). Direct Feedback Alignment Scales to Modern Deep Learning Tasks and Architectures. *Advances in Neural Information Processing Systems*, *33*, 9346–9360.

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. https://doi.org/10.1109/5.726791.

Lee, H., Grosse, R., Ranganath, R., & Ng, A. Y. (2009). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. *Proceedings of the 26th Annual International Conference on Machine Learning*, 609–616. https://doi.org/10.1145/1553374.1553453.

Lee, J. H., Mannelli, S. S., & Saxe, A. M. (2024, 21–27 Jul). Why do animals need shaping? A theory of task composition and curriculum learning. In R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, &

F. Berkenkamp (Editors), *Proceedings of the 41st international conference on machine learning* (Pages 26837–26855, Volume 235). PMLR. https://proceedings.mlr.press/v235/lee24r.html.

Lee, R. S., Sagiv, Y., Engelhard, B., Witten, I. B., & Daw, N. D. (2024). A feature-specific prediction error model explains dopaminergic heterogeneity. *Nature Neuroscience*, *27*(8), 1574–1586. https://doi.org/10.1038/s41593-024-01689-1.

Lei, X., Ito, T., & Bashivan, P. (2024). Geometry of naturalistic object representations in recurrent neural network models of working memory. *Advances in Neural Information Processing Systems*, *37*, 100604–100629.

Lens, W. (1987). Theoretical Research Should Be Useful and Used. *International Journal of Psychology*, *22*(4), 453–461. https://doi.org/10.1080/00207598708246787.

Levenstein, D., Alvarez, V. A., Amarasingham, A., Azab, H., Chen, Z. S., Gerkin, R. C., Hasenstaub, A., Iyer, R., Jolivet, R. B., Marzen, S., Monaco, J. D., Prinz, A. A., Quraishi, S., Santamaria, F., Shivkumar, S., Singh, M. F., Traub, R., Nadim, F., Rotstein, H. G., & Redish, A. D. (2023). On the Role of Theory and Modeling in Neuroscience. *Journal of Neuroscience*, *43*(7), 1074–1088. https://doi.org/10.1523/JNEUROSCI.1179-22.2022.

Li, N., Daie, K., Svoboda, K., & Druckmann, S. (2016). Robust neuronal dynamics in premotor cortex during motor planning. *Nature*, *532*(7600), 459–464. https://doi.org/10.1038/nature17643.

Lillicrap, T. P., & Santoro, A. (2019). Backpropagation through time and the brain. *Current Opinion in Neurobiology*, *55*, 82–89. https://doi.org/10.1016/j.conb.2019.01.011.

Liu, X., Zhang, F., Hou, Z., Wang, Z., Mian, L., Zhang, J., & Tang, J. (2021). Self-supervised Learning: Generative or Contrastive. *IEEE Transactions on Knowledge and Data Engineering*, *35*(1), 857–876. https://doi.org/10.1109/TKDE.2021.3090866.

Liu, Y. H., Ghosh, A., Richards, B., Shea-Brown, E., & Lajoie, G. (2022). Beyond accuracy: Generalization properties of bio-plausible temporal credit assignment rules. *Advances in Neural Information Processing Systems*, *35*, 23077–23097.

Lorach, H., Galvez, A., Spagnolo, V., Martel, F., Karakas, S., Intering, N., Vat, M., Faivre, O., Harte, C., Komi, S., Ravier, J., Collin, T., Coquoz, L., Sakr, I., Baaklini, E., Hernandez-Charpak, S. D., Dumont, G., Buschman, R., Buse, N., … Courtine, G. (2023). Walking naturally after spinal cord injury using a brain–spine interface. *Nature*, *618*(7963), 126–133. https://doi.org/10.1038/s41586-023-06094-5.

Lundqvist, M., Herman, P., & Miller, E. K. (2018). Working memory: Delay activity, yes! Persistent activity? Maybe not. *Journal of Neuroscience*, *38*(32), 7013–7019. https://doi.org/10.1523/JNEUROSCI.2485-17.2018.

Macovski, A. (2009). MRI: A charmed past and an exciting future. *Journal of Magnetic Resonance Imaging*, *30*(5), 919–923. https://doi.org/10.1002/jmri.21962.

Magee, J. C., & Grienberger, C. (2020). Synaptic Plasticity Forms and Functions. *Annual Review of Neuroscience*, *43*, 95–117. https://doi.org/10.1146/annurev-neuro-090919-022842.

Mannelli, S. S., Ivashynka, Y., Saxe, A. M., & Saglietti, L. (2024, 21–27 Jul). Tilting the odds at the lottery: The interplay of overparameterisation and curricula in neural networks. In R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, & F. Berkenkamp (Editors), *Proceedings of the 41st international conference on machine learning* (Pages 34586–34602, Volume 235). PMLR. https://proceedings.mlr.press/v235/mannelli24a.html.

Manzur, H. E., Vlasov, K., Jhong, Y.-J., Chen, H.-Y., & Lin, S.-C. (2023). The behavioral signature of stepwise learning strategy in male rats and its neural correlate in the basal forebrain. *Nature Communications*, *14*(1), 4415. https://doi.org/10.1038/s41467-023-40145-9.

Marcotte, E. R., Pearson, D. M., & Srivastava, L. K. (2001). Animal models of schizophrenia: A critical review. *Journal of Psychiatry and Neuroscience*, *26*(5), 395–410.

Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A. D., & van de Weijer, J. (2022). Class-incremental learning: Survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(5), 5513–5533.

Mathy, F., & Feldman, J. (2009). A rule-based presentation order facilitates category learning. *Psychonomic Bulletin & Review*, *16*(6), 1050–1057. https://doi.org/10.3758/PBR.16.6.1050.

Mathy, F., & Feldman, J. (2016). The Influence of Presentation Order on Category Transfer. *Experimental Psychology*, *63*(1), 59–69. https://doi.org/10.1027/1618-3169/a000312.

McGreevy, P., & Boakes, R. (2011). *Carrots and Sticks: Principles of Animal Training*. Darlington Press.

Meulemans, A., Farinha, M. T., Ordóñez, J. G., Aceituno, P. V., Sacramento, J., & Grewe, B. F. (2021). Credit Assignment in Neural Networks through Deep Feedback Control. *Advances in Neural Information Processing Systems*, *34*, 4674–4687.

Meyer, T., Qi, X.-L., Stanford, T. R., & Constantinidis, C. (2011). Stimulus selectivity in dorsal and ventral prefrontal cortex after training in working memory tasks. *Journal of Neuroscience*, *31*(17), 6266–6276. https://doi.org/10.1523/JNEUROSCI.6798-10.2011.

A Model of How the Basal Ganglia Generate and Use Neural Signals That Predict Reinforcement. (1995). In J. C. Houk, J. L. Davis, & D. G. Beiser (Editors), *Models of Information Processing in the Basal Ganglia*. The MIT Press. https://doi.org/10.7551/mitpress/4708.003.0020.

Montague, P. R., Hyman, S. E., & Cohen, J. D. (2004). Computational roles for dopamine in behavioural control. *Nature*, *431*(7010), 760–767. https://doi.org/10.1038/nature03015.

Morris, G., Nevet, A., Arkadir, D., Vaadia, E., & Bergman, H. (2006). Midbrain dopamine neurons encode decisions for future action. *Nature Neuroscience*, *9*(8), 1057–1063. https://doi.org/10.1038/nn1743.

Mosha, N., & Robertson, E. M. (2016). Unstable Memories Create a High-Level Representation that Enables Learning Transfer. *Current Biology*, *26*(1), 100–105. https://doi.org/10.1016/j.cub.2015.11.035.

Narvekar, S., Peng, B., Leonetti, M., Sinapov, J., Taylor, M. E., & Stone, P. (2020). Curriculum Learning for Reinforcement Learning Domains: A Framework and Survey. *Journal of Machine Learning Research*, *21*(181), 1–50.

Neha, Sodhi, R. K., Jaggi, A. S., & Singh, N. (2014). Animal models of dementia and cognitive dysfunction. *Life Sciences*, *109*(2), 73–86. https://doi.org/10.1016/j.lfs.2014.05.017.

Nicola, W., & Clopath, C. (2017). Supervised learning in spiking neural networks with FORCE training. *Nature Communications*, *8*(1), 2208. https://doi.org/10.1038/s41467-017-01827-3.

Ning, S. A., Flanzer, T. C., & Kroo, I. M. (2011). Aerodynamic Performance of Extended Formation Flight. *Journal of Aircraft*, *48*(3), 855–865. https://doi.org/10.2514/1.C031046.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y., Li, S.-W., Misra, I., Rabbat, M., Sharma, V., … Bojanowski, P. (2024). *DINOv2: Learning Robust Visual Features without Supervision*. arXiv: 2304.07193. https://doi.org/10.48550/arXiv.2304.07193.

O'Reilly, R. C., & Frank, M. J. (2006). Making Working Memory Work: A Computational Model of Learning in the Prefrontal Cortex and Basal Ganglia. *Neural Computation*, *18*(2), 283–328. https://doi.org/10.1162/089976606775093909.

Padfield, G. D., & Lawrence, B. (2003). The birth of flight control: An engineering analysis of the Wright brothers' 1902 glider. *The Aeronautical Journal*, *107*(1078), 697–718. https://doi.org/10.1017/S0001924000013464.

Padoa-Schioppa, C., & Assad, J. A. (2006). Neurons in the orbitofrontal cortex encode economic value. *Nature*, *441*(7090), 223–226. https://doi.org/10.1038/nature04676.

Parker, N. F., Baidya, A., Cox, J., Haetzel, L. M., Zhukovskaya, A., Murugan, M., Engelhard, B., Goldman, M. S., & Witten, I. B. (2022). Choice-selective sequences dominate in cortical relative to thalamic inputs to NAc to support reinforcement learning. *Cell Reports*, *39*(7), 110756. https://doi.org/10.1016/j.celrep.2022.110756.

Paugam-Moisy, H., & Bohte, S. (2012). Computing with Spiking Neuron Networks. In G. Rozenberg, T. Bäck, & J. N. Kok (Editors), *Handbook of Natural Computing* (Pages 335–376). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-92910-9_10.

Pedersen, M. L., Frank, M. J., & Biele, G. (2017). The drift diffusion model as the choice rule in reinforcement learning. *Psychonomic Bulletin & Review*, *24*, 1234–1251. https://doi.org/10.3758/s13423-016-1199-y.

Penick, C. A., Cope, G., Morankar, S., Mistry, Y., Grishin, A., Chawla, N., & Bhate, D. (2022). The Comparative Approach to Bio-Inspired Design: Integrating Biodiversity and Biologists into the Design Process. *Integrative and Comparative Biology*, *62*(5), 1153–1163. https://doi.org/10.1093/icb/icac097.

Pepels, T., Cazenave, T., Winands, M. H. M., & Lanctot, M. (2014). Minimizing Simple and Cumulative Regret in Monte-Carlo Tree Search. In T. Cazenave, M. H. M. Winands, & Y. Björnsson (Editors), *Computer Games: Third Workshop on Computer Games, CGW 2014, Held in Conjunction with the 21st European Conference on Artificial Intelligence, ECAI 2014, Prague, Czech Republic, August 18, 2014, Revised Selected Papers 3* (Pages 1–15). Springer International Publishing. https://doi.org/10.1007/978-3-319-14923-3_1.

Pinchetti, L., Qi, C., Lokshyn, O., Olivers, G., Emde, C., Tang, M., M'Charrak, A., Frieder, S., Menzat, B., Bogacz, R., Lukasiewicz, T., & Salvatori, T. (2025). Benchmarking Predictive Coding Networks – Made Simple. *The Thirteenth International Conference on Learning Representations.*

Pozzi, I., Bohté, S. M., & Roelfsema, P. R. (2020). Attention-gated brain propagation: How the brain can implement reward-based error backpropagation. *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2516–2526.

Pryor, K., & Ramirez, K. (2014). Modern Animal Training. In *The Wiley Blackwell Handbook of Operant and Classical Conditioning* (Pages 453–482). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781118468135.ch18.

Rani, V., Nabi, S. T., Kumar, M., Mittal, A., & Kumar, K. (2023). Self-supervised Learning: A Succinct Review. *Archives of Computational Methods in Engineering*, *30*(4), 2761–2775. https://doi.org/10.1007/s11831-023-09884-2.

Rao, R. P. N., & Ballard, D. H. (1999). Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, *2*(1), 79–87. https://doi.org/10.1038/4580.

Ravanelli, M., Brakel, P., Omologo, M., & Bengio, Y. (2018). Light Gated Recurrent Units for Speech Recognition. *IEEE Transactions on Emerging Topics in Computational Intelligence*, *2*(2), 92–102. https://doi.org/10.1109/TETCI.2017.2762739.

Ren, M., Kornblith, S., Liao, R., & Hinton, G. (2023). *Scaling Forward Gradient With Local Losses*. arXiv: 2210.03310 `[cs.LG]`. https://arxiv.org/abs/2210.03310.

Ren, P., Xiao, Y., Chang, X., Huang, P.-Y., Li, Z., Gupta, B. B., Chen, X., & Wang, X. (2022). A Survey of Deep Active Learning. *ACM Computing Surveys*, *54*(9), 1–40. https://doi.org/10.1145/3472291.

Rescorla, R. A., & Wagner, A. R. (1972). A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and non-reinforcement. In A. H. Black & W. F. Prokasy (Editors), *Classical conditioning, Current research and theory* (Pages 64–99, Volume 2). Appleton-Century-Crofts.

Roelfsema, P. R., & Holtmaat, A. (2018). Control of synaptic plasticity in deep cortical networks. *Nature Reviews Neuroscience*, *19*(3), 166–180. https://doi.org/10.1038/nrn.2018.6.

Roelfsema, P. R., & van Ooyen, A. (2005). Attention-gated reinforcement learning of internal representations for classification. *Neural Computation*, *17*(10), 2176–2214.

Rojas, R. (1996). The Backpropagation Algorithm. In R. Rojas (Editor), *Neural Networks: A Systematic Introduction* (Pages 149–182). Springer. https://doi.org/10.1007/978-3-642-61068-4_7.

Rombouts, J. O., Roelfsema, P. R., & Bohte, S. M. (2014). Learning Resets of Neural Working Memory. *ESANN*, 6.

Rombouts, J. O., Bohte, S. M., & Roelfsema, P. R. (2015). How Attention Can Create Synaptic Tags for the Learning of Working Memories in Sequential Tasks. *PLOS Computational Biology*, *11*(3), e1004060. https://doi.org/10.1371/journal.pcbi.1004060.

Rubruck, J., Bauer, J. P., Saxe, A., & Summerfield, C. (2024). *Early learning of the optimal constant solution in neural networks and humans*. arXiv: 2406.17467 `[cs.LG]`. https://arxiv.org/abs/2406.17467.

Rushworth, M. F. S., Noonan, M. P., Boorman, E. D., Walton, M. E., & Behrens, T. E. (2011). Frontal Cortex and Reward-Guided Learning and Decision-Making. *Neuron*, *70*(6), 1054–1069. https://doi.org/10.1016/j.neuron.2011.05.014.

Rusu, S. I., & Pennartz, C. M. A. (2020). Learning, memory and consolidation mechanisms for behavioral control in hierarchically organized cortico-basal ganglia systems. *Hippocampus*, *30*(1), 73–98. https://doi.org/10.1002/hipo.23167.

Sak, H., Senior, A., & Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. *Interspeech 2014*, 338–342. https://doi.org/10.21437/Interspeech.2014-80.

Schaeffer, R., Khona, M., Ma, T., Eyzaguirre, C., Koyejo, S., & Fiete, I. (2024). Self-Supervised Learning of Representations for Space Generates Multi-Modular Grid Cells. *Advances in Neural Information Processing Systems*, *36*, 23140–23157.

Scherr, F., Stöckl, C., & Maass, W. (2020). *One-shot learning with spiking neural networks*. 2020.06.17.156513. https://doi.org/10.1101/2020.06.17.156513.

Schmahmann, J. D. (2019). The cerebellum and cognition. *Neuroscience Letters*, *688*, 62–75. https://doi.org/10.1016/j.neulet.2018.07.005.

Schmidgall, S., & Hays, J. (2023). Meta-SpikePropamine: Learning to learn with synaptic plasticity in spiking neural networks. *Frontiers in Neuroscience*, *17*. https://doi.org/10.3389/fnins.2023.1183321.

Schmitt, L. I., Wimmer, R. D., Nakajima, M., Happ, M., Mofakham, S., & Halassa, M. M. (2017). Thalamic amplification of cortical connectivity sustains attentional control. *Nature*, *545*(7653), 219–223. https://doi.org/10.1038/nature22073.

Schultz, W. (2007). Multiple Dopamine Functions at Different Time Courses. *Annual Review of Neuroscience*, *30*(1), 259–288. https://doi.org/10.1146/annurev.neuro.28.061604.135722.

Schultz, W. (2016a). Dopamine reward prediction error coding. *Dialogues in Clinical Neuroscience*, *18*(1), 23–32. https://doi.org/10.31887/DCNS.2016.18.1/wschultz.

Schultz, W. (2016b). Dopamine reward prediction-error signalling: A two-component response. *Nature Reviews Neuroscience*, *17*(3), 183–195. https://doi.org/10.1038/nrn.2015.26.

Schulz, R. A., Stein, J. A., & Pelc, N. J. (2021). How CT happened: The early development of medical computed tomography. *Journal of Medical Imaging*, *8*(5), 052110. https://doi.org/10.1117/1.JMI.8.5.052110.

Schwalb, J. M., & Hamani, C. (2008). The history and future of deep brain stimulation. *Neurotherapeutics*, *5*, 3–13. https://doi.org/10.1016/j.nurt.2007.11.003.

Seijen, H., & Sutton, R. (2014). True Online TD(lambda). *Proceedings of the 31st International Conference on Machine Learning*, 692–700.

Shannon, R. V. (2012). Advances in Auditory Prostheses. *Current opinion in neurology*, *25*(1), 61–66. https://doi.org/10.1097/WCO.0b013e32834ef878.

Sharma, S., Sharma, S., & Athaiya, A. (2020). Activation Functions in Neural Networks. *International Journal of Engineering Applied Sciences and Technology*, *4*(12), 310–316. https://doi.org/10.33564/IJEAST.2020.v04i12.054.

Shen, G., Zhao, D., Dong, Y., & Zeng, Y. (2023). Brain-inspired neural circuit evolution for spiking neural networks. *Proceedings of the National Academy of Sciences*, *120*(39), e2218173120. https://doi.org/10.1073/pnas.2218173120.

Shima, K., & Tanji, J. (1998). Role for Cingulate Motor Area Cells in Voluntary Movement Selection Based on Reward. *Science*, *282*(5392), 1335–1338. https://doi.org/10.1126/science.282.5392.1335.

Siddiqui, S. A., Krueger, D., LeCun, Y., & Deny, S. (2023). *Blockwise Self-Supervised Learning at Scale*. arXiv: 2302.01647 [cs.CV]. https://arxiv.org/abs/2302.01647.

Skinner, G., & Walmsley, T. (2019). Artificial intelligence and deep learning in video games a brief review. *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, 404–408. https://doi.org/10.1109/CCOMS.2019.8821783.

Smith, J. D., Chapman, W. P., & Redford, J. S. (2010). Stages of Category Learning in Monkeys (Macaca mulatta) and Humans (Homo sapiens). *Journal of experimental psychology. Animal behavior processes*, *36*(1), 39–53. https://doi.org/10.1037/a0016573.

Smith, J. D., Crossley, M. J., Boomer, J., Church, B. A., Beran, M. J., & Ashby, F. G. (2012). Implicit and Explicit Category Learning by Capuchin Monkeys (Cebus apella). *Journal of comparative psychology*, *126*(3), 294–304. https://doi.org/10.1037/a0026031.

Sobal, V., Ibrahim, M., Balestriero, R., Cabannes, V., Bouchacourt, D., Astolfi, P., Cho, K., & LeCun, Y. (2024). *$\mathbb{X}$-sample contrastive loss: Improving contrastive learning with sample similarity graphs.* arXiv: 2407.18134 `[cs.CV]`. https://arxiv.org/abs/2407.18134.

Song, Y., Millidge, B., Salvatori, T., Lukasiewicz, T., Xu, Z., & Bogacz, R. (2024). Inferring neural activity before plasticity as a foundation for learning beyond backpropagation. *Nature Neuroscience*, *27*(2), 348–358. https://doi.org/10.1038/s41593-023-01514-1.

Sonneborn, A., Bartlett, L., Olson, R. J., Milton, R., & Abbas, A. I. (2024). Divergent subregional information processing in mouse prefrontal cortex during working memory. *Communications Biology*, *7*(1), 1235. https://doi.org/10.1038/s42003-024-06926-8.

Soviany, P., Ionescu, R. T., Rota, P., & Sebe, N. (2022). Curriculum Learning: A Survey. *International Journal of Computer Vision*, *130*(6), 1526–1565. https://doi.org/10.1007/s11263-022-01611-x.

Spratling, M. W. (2017). A review of predictive coding algorithms. *Brain and Cognition*, *112*, 92–97. https://doi.org/10.1016/j.bandc.2015.11.003.

Starkweather, C. K., & Uchida, N. (2021). Dopamine signals as temporal difference errors: Recent advances. *Current opinion in neurobiology*, *67*, 95–105. https://doi.org/10.1016/j.conb.2020.08.014.

Stokes, M. G. (2015). 'Activity-silent' working memory in prefrontal cortex: A dynamic coding framework. *Trends in Cognitive Sciences*, *19*(7), 394–405. https://doi.org/10.1016/j.tics.2015.05.004.

Stroud, J. P., Duncan, J., & Lengyel, M. (2024). The computational foundations of dynamic coding in working memory. *Trends in Cognitive Sciences*, *28*(7), 614–627. https://doi.org/10.1016/j.tics.2024.02.011.

Sturgill, J., Hegedus, P., Li, S., Chevy, Q., Siebels, A., Jing, M., Li, Y., Hangya, B., & Kepecs, A. (2020). *Basal forebrain-derived acetylcholine encodes valence-free reinforcement prediction error.* https://doi.org/10.1101/2020.02.17.953141.

Subramoney, A., Bellec, G., Scherr, F., Legenstein, R., & Maass, W. (2021). *Revisiting the role of synaptic plasticity and network dynamics for fast learning in spiking neural networks.* https://doi.org/10.1101/2021.01.25.428153.

Sutskever, I. (2013). *Training recurrent neural networks (doctoral thesis).* http://hdl.handle.net/1807/36012.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, *3*(1), 9–44. https://doi.org/10.1007/BF00115009.

Sutton, R. S. (2022). *A History of Meta-gradient: Gradient Methods for Meta-learning.* arXiv: 2202.09701 `[cs]`. https://arxiv.org/abs/2202.09701.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning, second edition: An Introduction.* MIT Press.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2014). *Intriguing properties of neural networks.* arXiv: 1312.6199 `[cs]`. https://doi.org/10.48550/arXiv.1312.6199.

Talloen, J., Dambre, J., & Vandesompele, A. (2021). *PyTorch-Hebbian: Facilitating local learning in a deep learning framework.* arXiv: 2102.00428 `[cs.LG]`. https://arxiv.org/abs/2102.00428.

Tang, M., Yang, Y., & Amit, Y. (2022). Biologically Plausible Training Mechanisms for Self-Supervised Learning in Deep Networks. *Frontiers in Computational Neuroscience*, *16*, 789253. https://doi.org/10.3389/fncom.2022.789253.

Tavanaei, A., Ghodrati, M., Kheradpisheh, S. R., Masquelier, T., & Maida, A. (2019). Deep learning in spiking neural networks. *Neural Networks*, *111*, 47–63. https://doi.org/10.1016/j.neunet.2018.12.002.

Taylor, K., & Rohrer, D. (2010). The effects of interleaved practice. *Applied Cognitive Psychology*, *24*(6), 837–848. https://doi.org/10.1002/acp.1598.

Teufel, C., & Fletcher, P. C. (2016). The promises and pitfalls of applying computational models to neurological and psychiatric disorders. *Brain*, *139*(10), 2600–2608. https://doi.org/10.1093/brain/aww209.

Thirunavukarasu, A. J., Ting, D. S. J., Elangovan, K., Gutierrez, L., Tan, T. F., & Ting, D. S. W. (2023). Large language models in medicine. *Nature Medicine*, *29*(8), 1930–1940. https://doi.org/10.1038/s41591-023-02448-8.

Thrun, S., & Pratt, L. (1998). Learning to Learn: Introduction and Overview. In S. Thrun & L. Pratt (Editors), *Learning to Learn* (Pages 3–17). Springer US. https://doi.org/10.1007/978-1-4615-5529-2_1.

Tscshantz, A., Millidge, B., Seth, A. K., & Buckley, C. L. (2023). Hybrid predictive coding: Inferring, fast and slow. *PLoS Computational Biology*, *19*(8), e1011280. https://doi.org/10.1371/journal.pcbi.1011280.

Tuite, K., Girotti, M., & Morilak, D. (2022). Activation of the Central Medial Thalamic Afferent to the Orbitofrontal Cortex Contributes to Successful Reversal Learning. *The FASEB Journal*, *36*(S1). https://doi.org/10.1096/fasebj.2022.36.S1.R2678.

Urban, D. J., & Roth, B. L. (2015). DREADDs (Designer Receptors Exclusively Activated by Designer Drugs): Chemogenetic Tools with Therapeutic Utility. *Annual Review of Pharmacology and Toxicology*, *55*(1), 399–417. https://doi.org/10.1146/annurev-pharmtox-010814-124803.

Ursino, M., Cesaretti, N., & Pirazzini, G. (2023). A model of working memory for encoding multiple items and ordered sequences exploiting the theta-gamma code. *Cognitive Neurodynamics*, *17*(2), 489–521. https://doi.org/10.1007/s11571-022-09836-9.

van Kerkoerle, T., Self, M. W., & Roelfsema, P. R. (2017). Layer-specificity in the effects of attention and working memory on activity in primary visual cortex. *Nature Communications*, *8*(1), 13804. https://doi.org/10.1038/ncomms13804.

van den Berg, A. R., Roelfsema, P. R., & Bohte, S. M. (2024). Biologically plausible gated recurrent neural networks for working memory and learning-to-learn. *PLoS ONE*, *19*(12), e0316453. https://doi.org/10.1371/journal.pone.0316453.

Voitov, I., & Mrsic-Flogel, T. D. (2022). Cortical feedback loops bind distributed representations of working memory. *Nature*, *608*(7922), 381–389. https://doi.org/10.1038/s41586-022-05014-3.

Wagemans, J., Feldman, J., Gepshtein, S., Kimchi, R., Pomerantz, J. R., van der Helm, P. A., & van Leeuwen, C. (2012). A Century of Gestalt Psychology in Visual Perception II. Conceptual and Theoretical Foundations. *Psychological bulletin*, *138*(6), 1218–1252. https://doi.org/10.1037/a0029334.

Walsh, K. S., McGovern, D. P., Clark, A., & O'Connell, R. G. (2020). Evaluating the neurophysiological evidence for predictive processing as a model of perception. *Annals of the New York Academy of Sciences*, *1464*(1), 242–268. https://doi.org/10.1111/nyas.14321.

Wang, H. E., Triebkorn, P., Breyton, M., Dollomaja, B., Lemarechal, J.-D., Petkoski, S., Sorrentino, P., Depannemaecker, D., Hashemi, M., & Jirsa, V. K. (2024). Virtual brain twins: From basic neuroscience to clinical use. *National Science Review*, *11*(5), nwae079. https://doi.org/10.1093/nsr/nwae079.

Wang, J. X. (2021). Meta-learning in natural and artificial intelligence. *Current Opinion in Behavioral Sciences*, *38*, 90–95. https://doi.org/10.1016/j.cobeha.2021.01.002.

Wang, J. X., Kurth-Nelson, Z., Kumaran, D., Tirumala, D., Soyer, H., Leibo, J. Z., Hassabis, D., & Botvinick, M. (2018). Prefrontal cortex as a meta-reinforcement learning system. *Nature Neuroscience*, *21*(6), 860–868. https://doi.org/10.1038/s41593-018-0147-8.

Wang, J. X., Kurth-Nelson, Z., Tirumala, D., Soyer, H., Leibo, J. Z., Munos, R., Blundell, C., Kumaran, D., & Botvinick, M. (2017). *Learning to reinforcement learn.* arXiv: 1611.05763 `[cs, stat]`. https://doi.org/10.48550/arXiv.1611.05763.

Wang, Y., Yin, X., Zhang, Z., Li, J., Zhao, W., & Guo, Z. V. (2021). A cortico-basal ganglia-thalamo-cortical channel underlying short-term memory. *Neuron*, *109*(21), 3486–3499. https://doi.org/10.1016/j.neuron.2021.08.002.

Watabe-Uchida, M., Eshel, N., & Uchida, N. (2017). Neural Circuitry of Reward Prediction Error. *Annual Review of Neuroscience*, *40*(1), 373–394. https://doi.org/10.1146/annurev-neuro-072116-031109.

Weiler, R., Brucklacher, M., Pennartz, C. M. A., & Bohté, S. M. (2024). Masked Image Modeling as a Framework for Self-Supervised Learning across Eye Movements. *International Conference on Artificial Neural Networks*, 17–31.

Weinshall, D., & Amir, D. (2020). Theory of Curriculum Learning, with Convex Loss Functions. *Journal of Machine Learning Research*, *21*(222), 1–19.

Weinshall, D., Cohen, G., & Amir, D. (2018). *Curriculum Learning by Transfer Learning: Theory and Experiments with Deep Networks*. arXiv: 1802.03796 [cs.LG]. https://arxiv.org/abs/1802.03796.

Wen, S., Rios, A. S., Lekkala, K., & Itti, L. (2022). *What can we learn from misclassified ImageNet images?* arXiv: 2201.08098 [cs.CV]. https://arxiv.org/abs/2201.08098.

Wert-Carvajal, C., Reneaux, M., Tchumatchenko, T., & Clopath, C. (2022). Dopamine and serotonin interplay for valence-based spatial learning. *Cell Reports*, *39*(2), 110645. https://doi.org/10.1016/j.celrep.2022.110645.

Whittington, J. C. R., & Bogacz, R. (2019). Theories of Error Back-Propagation in the Brain. *Trends in Cognitive Sciences*, *23*(3), 235–250. https://doi.org/10.1016/j.tics.2018.12.005.

Wilhelm, M., Sych, Y., Fomins, A., Alatorre Warren, J. L., Lewis, C., Serratosa Capdevila, L., Boehringer, R., Amadei, E. A., Grewe, B., O'Connor, E. C., Hall, B. J., & Helmchen, F. (2023). Striatum-projecting prefrontal cortex neurons support working memory maintenance. *Nature Communications*, *14*(1), 7016. https://doi.org/10.1038/s41467-023-42777-3.

Williams, R. J., & Peng, J. (1990). An Efficient Gradient-Based Algorithm for On-Line Training of Recurrent Network Trajectories. *Neural Computation*, *2*(4), 490–501. https://doi.org/10.1162/neco.1990.2.4.490.

Wilson, R. C., Shenhav, A., Straccia, M., & Cohen, J. D. (2019). The Eighty Five Percent Rule for optimal learning. *Nature Communications*, *10*(1), 4646. https://doi.org/10.1038/s41467-019-12552-4.

Wu, X., Liu, X., Li, W., & Wu, Q. (2018). Improved expressivity through dendritic neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Editors), *Advances in neural information processing systems* (Volume 31). Curran Associates, Inc.

Xiao, W., Chen, H., Liao, Q., & Poggio, T. (2018). *Biologically-plausible learning algorithms can scale to large datasets*. arXiv: 1811.03567 [cs.LG]. https://arxiv.org/abs/1811.03567.

Yamaguchi, K., Maeda, Y., Sawada, T., Iino, Y., Tajiri, M., Nakazato, R., Ishii, S., Kasai, H., & Yagishita, S. (2022). A behavioural correlate of the synaptic eligibility trace in the nucleus accumbens. *Scientific Reports*, *12*(1), 1921. https://doi.org/10.1038/s41598-022-05637-6.

Yen, C., Lin, C.-L., & Chiang, M.-C. (2023). Exploring the Frontiers of Neuroimaging: A Review of Recent Advances in Understanding Brain Functioning and Disorders. *Life*, *13*(7), 1472. https://doi.org/10.3390/life13071472.

Żakowski, W. (2020). Animal Use in Neurobiological Research. *Neuroscience*, *433*, 1–10. https://doi.org/10.1016/j.neuroscience.2020.02.049.

Zaremba, W., & Sutskever, I. (2015). *Learning to Execute.* arXiv: 1410.4615 [cs.NE]. https://arxiv.org/abs/1410.4615.

Zenke, F. (no date). *Private communication.*

Zhang, M., & Barash, S. (2000). Neuronal switching of sensorimotor transformations for antisaccades. *Nature*, *408*(6815), 971–975. https://doi.org/10.1038/35050097.

Zhang, M., & Barash, S. (2004). Persistent LIP Activity in Memory Antisaccades: Working Memory For a Sensorimotor Transformation. *Journal of Neurophysiology*, *91*(3), 1424–1441.

Zheng, H., Zheng, Z., Hu, R., Xiao, B., Wu, Y., Yu, F., Liu, X., Li, G., & Deng, L. (2024). Temporal dendritic heterogeneity incorporated with spiking neural networks for learning multi-timescale dynamics. *Nature Communications*, *15*(1), 277. https://doi.org/10.1038/s41467-023-44614-z.

Zhou, Z., & Firestone, C. (2019). Humans can decipher adversarial images. *Nature Communications*, *10*(1), 1334. https://doi.org/10.1038/s41467-019-08931-6.