UNIVERSITY OF CALIFORNIA SAN DIEGO

Unsupervised sleep-like processes for enhancing neural networks

A dissertation submitted in partial satisfaction of the
requirements for the degree of Doctor of Philosophy

in

Computer Science

by

Jean Erik Delanois

Committee in charge:

Professor Maxim Bazhenov, Co-Chair
Professor Julian McAuley, Co-Chair
Professor Debashis Sahoo
Professor Rose Yu

2024

The dissertation of Jean Erik Delanois is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2024

Dedication

I dedicate my dissertation to my family.

To my father and mother, Clark and Roberta, for always supporting me and giving me the courage to pursue my passions.

To my sisters, Gabrielle and Rachel, for motivating me to graduate and promising to honor my name by addressing me as "Doctor" forevermore.

Table of Contents

List of Figures

List of Supplemental Figures

Acknowledgements

I would like to acknowledge my advisors, coauthors, and collaborators.

Chapter 1, in full, is a reprint of the material as it appears Elife.

González, O. C., Sokolov, Y., Krishnan, G. P., Delanois, J. E., & Bazhenov, M. (2020). Can sleep protect memories from catastrophic forgetting?. Elife, 9, e51005.

Chapter 2, in full, is a reprint of the material as it appears in PLOS Computational Biology.

Delanois, J. E., Golden, R., Sanda, P., & Bazhenov, M. (2022). Sleep prevents catastrophic forgetting in spiking neural networks by forming a joint synaptic weight representation. PLOS Computational Biology, 18(11), e1010628.

Chapter 3, in full, is a reprint of the material as it appears in ICMLA.

Delanois, J. E., Ahuja, A., Krishnan, G. P., Tadros, T., McAuley, J., & Bazhenov, M. (2023, December). Improving Robustness of Convolutional Networks Through Sleep-Like Replay. In 2023 International Conference on Machine Learning and Applications (ICMLA) (pp. 257-264). IEEE.

Vita

2016	Bachelor of Science, University of Illinois Urbana-Champaign
2017 – 2018	Industry Application Development
2018 – 2019	Computational Bioengineering Research Associate
2022	Master of Science, University of California San Diego
2019 - 2024	Research Assistant, University of California San Diego
2024	Doctor of Philosophy, University of California San Diego

Publications

Delanois J. E., Ahuja A., Krishnan G., Tadros T., McAuley J., Bazhenov M (2023, December). Improving robustness of convolutional networks through sleep-like replay. ICMLA 2023.

Delanois, J. E., Golden, R., Sanda, P., & Bazhenov, M. (2022). Sleep prevents catastrophic forgetting in spiking neural networks by forming a joint synaptic weight representation. PLOS Computational Biology, 18(11), e1010628

González, O. C., Sokolov, Y., Krishnan, G. P., Delanois, J. E., & Bazhenov, M. (2020). Can sleep protect memories from catastrophic forgetting?. Elife, 9, e51005.

FIELDS OF STUDY

Machine Learning, Computational Neuroscience, Computer Science

Abstract of the Dissertation

Unsupervised sleep-like processes for enhancing neural networks

by

Jean Erik Delanois

Doctor of Philosophy in Computer Science

University of California San Diego, 2024

Professor Maxim Bazhenov, Co-Chair
Professor Julian McAuley, Co-Chair

Advancing our understanding of neuroscience and artificial intelligence, this dissertation aims to progress our understanding of memory representation, consolidation, and robustness within neural networks. While the brain serves as a remarkable inspiration for machine learning, our comprehension of its complexities remains limited. Gaining insight in how the brain operates enables mutual progress in both fields simultaneously, one potential avenue is through exploring sleep. Sleep is a significant yet only partially understood phenomena that occurs in biological brains. This critical physiological process is prevalent across species due to its pivotal role for many biologically relevant metabolic and cognitive functions; importantly sleep has been shown to be crucial for memory enhancement and consolidation. Despite the extreme importance of natural sleep, there is no true artificial counterpart in machine learning. This work elucidates the intricate mechanisms by which sleep enhances memory representation through biophysical

modeling and applies these principals to a range of network architectures across the biophysical-artificial spectrum for a variety of tasks. Specifically, sleep mechanisms are conceptualized and illustrated in biophysical Hodgkin-Huxley neural networks capable of realistic wake and sleep activity. Similar sleep-like stages are then applied to map-based spiking neural networks to mitigate catastrophic forgetting in a sequential learning paradigm. Finally, fully bridging the neuroscience / artificial intelligence gap, a sleep based algorithm for artificial convolutional neural networks is proposed which bolsters the resilience of convolutional filters thereby improving model performance in distorted contexts. Collectively, this dissertation sheds light on the role of sleep in shaping memory across diverse neural systems and reimagines the relationship between artificial and biological intelligence.

eLife

Can sleep protect memories from catastrophic forgetting?

Oscar C González[1†], Yury Sokolov[1†], Giri P Krishnan[1], Jean Erik Delanois[1,2], Maxim Bazhenov[1*]

[1]Department of Medicine, University of California, San Diego, La Jolla, United States; [2]Department of Computer Science and Engineering, University of California, San Diego, La Jolla, United States

Abstract Continual learning remains an unsolved problem in artificial neural networks. The brain has evolved mechanisms to prevent catastrophic forgetting of old knowledge during new training. Building upon data suggesting the importance of sleep in learning and memory, we tested a hypothesis that sleep protects old memories from being forgotten after new learning. In the thalamocortical model, training a new memory interfered with previously learned old memories leading to degradation and forgetting of the old memory traces. Simulating sleep after new learning reversed the damage and enhanced old and new memories. We found that when a new memory competed for previously allocated neuronal/synaptic resources, sleep replay changed the synaptic footprint of the old memory to allow overlapping neuronal populations to store multiple memories. Our study predicts that memory storage is dynamic, and sleep enables continual learning by combining consolidation of new memory traces with reconsolidation of old memory traces to minimize interference.

***For correspondence:** mbazhenov@ucsd.edu

[†]These authors contributed equally to this work

Competing interests: The authors declare that no competing interests exist.

Funding: See page 27

Received: 19 January 2020
Accepted: 19 July 2020
Published: 04 August 2020

Reviewing editor: Mark CW van Rossum, University of Nottingham, United Kingdom

Introduction

Animals and humans are capable of continuous, sequential learning. In contrast, modern artificial neural networks suffer from the inability to perform continual learning (***Ratcliff, 1990***; ***French, 1999***; ***Hassabis et al., 2017***; ***Hasselmo, 2017***; ***Kirkpatrick et al., 2017***). Training a new task results in interference and catastrophic forgetting of old memories (***Ratcliff, 1990***; ***McClelland et al., 1995***; ***French, 1999***; ***Hasselmo, 2017***). Several attempts have been made to overcome this problem including (a) explicit retraining of all previously learned memories – interleaved training (***Hasselmo, 2017***), (b) using generative models to reactivate previous inputs (***Kemker and Kanan, 2017***), or (c) artificially 'freezing' subsets of synapses important for the old memories (***Kirkpatrick et al., 2017***). These solutions help prevent new memories from interfering with previously stored old memories, however they either require explicit retraining of all past memories using the original data or have limitations on the types of trainable new memories and network architectures (***Kemker and Kanan, 2017***). How biological systems avoid catastrophic forgetting remains to be understood. In this paper, we propose a mechanism for how sleep modifies network synaptic connectivity to minimize interference of competing memory traces enabling continual learning.

Sleep has been suggested to play an important role in learning and memory (***Paller and Voss, 2004***; ***Walker and Stickgold, 2004***; ***Oudiette et al., 2013***; ***Rasch and Born, 2013***; ***Stickgold, 2013***; ***Weigenand et al., 2016***; ***Wei et al., 2018***). Specifically, the role of stage 2 (N2) and stage 3 (N3) of Non-Rapid Eye Movement (NREM) sleep has been shown to help with the consolidation of newly encoded memories (***Paller and Voss, 2004***; ***Walker and Stickgold, 2004***; ***Rasch and Born, 2013***; ***Stickgold, 2013***). The mechanism by which memory consolidation is influenced by sleep is still debated, however, a number of hypotheses have been put forward. Sleep may enable memory consolidation through repeated reactivation or *replay* of specific memory traces during characteristic

sleep rhythms such as spindles and slow oscillations (**Paller and Voss, 2004**; **Clemens et al., 2005**; **Marshall et al., 2006**; **Oudiette et al., 2013**; **Rasch and Born, 2013**; **Weigenand et al., 2016**; **Ladenbauer et al., 2017**; **Wei et al., 2018**; **Xu et al., 2019**). Memory replay during NREM sleep could help strengthen previously stored memories and map memory traces between brain structures. Previous work using electrical (**Marshall et al., 2004**; **Marshall et al., 2006**; **Ladenbauer et al., 2017**) or auditory (**Ngo et al., 2013**) stimulation showed that increasing neocortical oscillations during NREM sleep resulted in improved consolidation of declarative memories. Similarly, spatial memory consolidation has been shown to improve following cued reactivation of memory traces during NREM sleep (**Paller and Voss, 2004**; **Oudiette et al., 2013**; **Oudiette and Paller, 2013**; **Papalambros et al., 2017**). Our recent computational studies found that sleep dynamics can lead to replay and strengthening of recently learned memory traces (**Wei et al., 2016**; **Wei et al., 2018**; **Wei et al., 2020**). These studies point to the critical role of sleep in memory consolidation.

Can neuroscience inspired ideas help solve the catastrophic forgetting problem in artificial neuronal networks? The most common machine learning training algorithm – backpropagation (**Rumelhart et al., 1986**; **Werbos, 1990**; **Kriegeskorte, 2015**) – is very different from plasticity rules utilized by brain networks. Nevertheless, we have recently seen a number of successful attempts to implement high level principles of biological learning in artificial network designs, including implementation of the ideas from 'Complementary Learning System Theory' (**McClelland et al., 1995**), according to which the hippocampus is responsible for the fast acquisition of new information, while the neocortex would more gradually learn a generalized and distributed representation. These ideas led to interesting attempts of solving the catastrophic forgetting problem in artificial neural networks (**Kemker and Kanan, 2017**). While few attempts have been made to implement sleep in artificial networks, one study suggested that sleep-like activity can increase storage capacity in artificial networks (**Fachechi et al., 2019**). We recently found that implementation of a sleep-like phase in artificial networks trained using backpropagation can dramatically reduce catastrophic forgetting, as well as improve generalization performance and transfer of knowledge (**Krishnan et al., 2019**; **Tadros et al., 2020**). However, despite this progress, we are still lacking a basic understanding of the mechanisms by which sleep replay affects memories, especially when new learning interferes with old knowledge.

The ability to store and retrieve sequentially related information is arguably the foundation of intelligent behavior. It allows us to predict the outcomes of sensory situations, to achieve goals by generating sequences of motor actions, to 'mentally' explore the possible outcomes of different navigational or motor choices, and ultimately to communicate through complex verbal sequences generated by flexibly chaining simpler elemental sequences learned in childhood. In our new study, we trained a network, capable of transitioning between sleep-like and wake-like states, to learn spike sequences in order to identify mechanisms by which sleep allows consolidation of newly encoded memory sequences and prevents damage to old memories. Our study predicts that during a period of sleep, following training of a new memory sequence in awake, both old and new memory traces are spontaneously replayed, preventing forgetting and increasing recall performance. We found that sleep replay results in fine tuning of the synaptic connectivity matrix encoding the interfering memory sequences to allow overlapping populations of neurons to store multiple competing memories.

Results

The network model, used in our study, represents a minimal thalamocortical architecture implementing one cortical layer (consisting of excitatory pyramidal (PY) and inhibitory (IN) neurons) and one thalamic layer (consisting of excitatory thalamic relay (TC) and inhibitory reticular thalamic (RE) neurons) – with all neurons simulated by Hodgkin-Huxley models (**Figure 1A**). These models were built upon neuron models we used in our earlier work (**Krishnan et al., 2016**; **Wei et al., 2016**; **Wei et al., 2018**). This model exhibits two primary dynamical states of the thalamocortical system – awake, characterized by random asynchronous firing of all cortical neurons, and slow-wave sleep (SWS), characterized by slow (<1 Hz) oscillations between Up (active) and Down (silent) states (**Blake and Gerard, 1937**; **Steriade et al., 1993**; **Steriade et al., 2001**). Transitions between sleep and awake (**Figure 1B/C**) were simulated by changing network parameters to model effect of neuromodulators (**Krishnan et al., 2016**). While the thalamic population was part of the network, its role

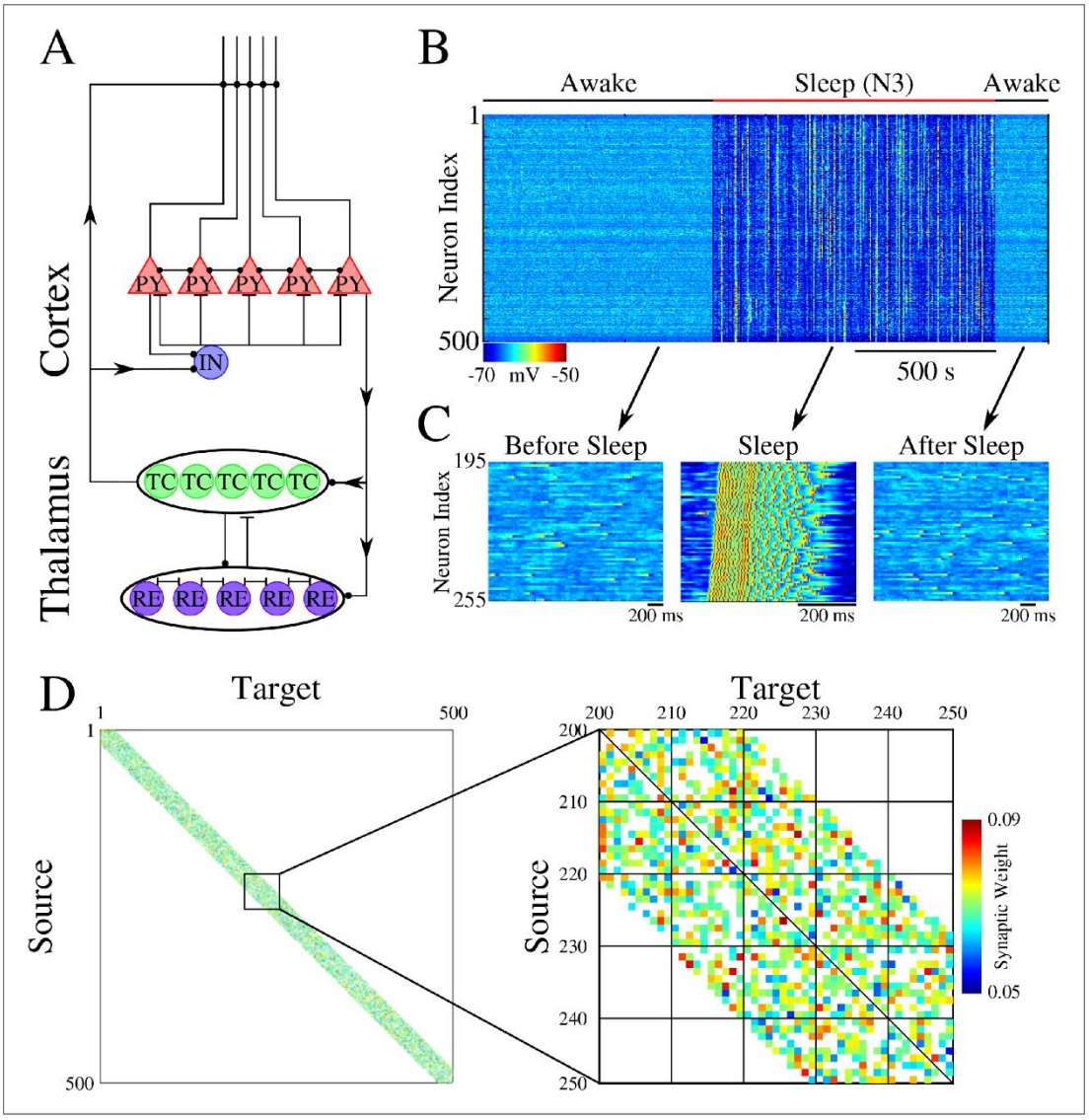


Figure 1. Network architecture and baseline dynamics. (**A**) Basic network architecture (PY: excitatory pyramidal neurons; IN: inhibitory interneurons; TC: excitatory thalamocortical neurons; RE: inhibitory thalamic reticular neurons). Excitatory synapses are represented by lines terminating in a dot, while inhibitory synapses are represented by lines terminating in bars. Arrows indicate the direction of the connection. (**B**) Behavior of a control network exhibiting wake-sleep transitions. Cortical PY neurons are shown. Color represents the voltage of a neuron at a given time during the simulation (dark blue – hyperpolarized potential; light blue / yellow – depolarized potential; red - spike). (**C**) Zoom-in of a subset of neurons from the network in B (time is indicated by arrows). Left and right panels show spontaneous activity during awake-like state before and after sleep, respectively. Middle panel shows example of activity during sleep. (**D**) Left panel shows the initial weighted adjacency matrix for the network in B. The color in this plot represents the strength of the AMPA connections between PY neurons, with white indicating the lack of synaptic connection. Right panel shows the initial weighted adjacency matrix for the subregion indicated on the left.

was limited to help simulate realistic Up and Down state activity (**Bazhenov et al., 2002**), as all synaptic changes occurred in the cortical population. The initial strength of the synaptic connections between cortical PY neurons was Gaussian distributed (**Figure 1D**).

We set probabilistic connectivity (p=0.6) between excitatory cortical neurons within a defined radius ($R_{AMPA(PY-PY)}$=20). Only cortical PY-PY connections were plastic and regulated by spike-timing dependent plasticity (STDP). During initial training, STDP was biased for potentiation to simulate elevated levels of acetylcholine (**Blokland, 1995**; **Shinoe et al., 2005**; **Sugisaki et al., 2016**). During testing/retrieval, STDP was balanced (LTD/LTP = 1). STDP remained balanced during both sleep and interleaved training (except for few selected simulations where we tested effect of unbalancing STDP) to allow side by side comparisons. For details, please see *Methods and Materials*.

Temporally structured sequences of events are a common type of information we learn, and they are believed to be represented in the brain by sequences of neuronal firing. Therefore, in this study we represent each memory pattern as an ordered sequence, S, of activations of populations of cortical neurons (e.g., A→B→. . .), where each 'letter' (e.g., A) labels a population of neurons, so each memory could be labeled by a unique 'word' of such 'letters'. We considered memory patterns represented by non-overlapping populations of neurons as well as memory patterns sharing neurons but with a different activation order, for example, A→B→C vs. C→B→A. This setup can mimic, for example, *in vivo* experiments with a rat learning a track, including: (a) running in one direction on a linear track (**Mehta et al., 1997**) would be equivalent to a sequence training ('A→B→C', 'A→B→C',. . .); (b) forwards and backwards running on a linear track (**Navratilova et al., 2012**) would be equivalent to interleaved sequences training ('A→B→C', 'C→B→A', 'A→B→C',. . .); (c) running on a belt track first only in one direction and then in reverse one (e.g., using Virtual Reality (VR) apparatus) would be equivalent to first learning a sequence ('A→B→C', 'A→B→C',. . .) and then the opposite one ('C→B→A', 'C→B→A',. . .).

In our model, training always occurred in the awake state and no input was delivered to the network in the sleep state. Testing was also done in the awake state; during test sessions, the model was only presented with input to the first group (e.g., A) to test for pattern completion for the trained sequence (e.g., A→B→C→. . .). Performance was calculated based on the distance between the trained pattern (template) and the response during testing. The awake state included multiple testing sessions: before training, after training/before sleep, and after sleep. For details, please see *Methods and Materials*.

The paper is organized as follows. We first consider the scenario of two memory sequences trained at different (non-overlapping) network locations. We show that SWS-like activity after training leads to sequence replay, synaptic weight changes, and performance increases during testing after sleep. Next, we focus on the case of two sequences trained in opposite directions over the same population of neurons. We show that in such a case training a new sequence in awake would 'erase' an old memory. However, if a sleep phase is implemented before complete destruction of the old memory, both memory sequences are spontaneously replayed during sleep. As a result of replay, each sequence allocates its own subset of neurons/synapses, and performance increases for both sequences during testing after sleep. We complete the study with a detailed analysis of synaptic weight changes and replay dynamics during the sleep state to identify mechanisms of memory consolidation and performance increase. In supplementary figures, we compare sleep replay with interleaved training and show that sleep achieves similar or better performance but without explicit access to the training data.

Training of spatially separated memory sequences does not lead to interference

First, we trained two memory patterns, S1 and S2, sequentially (first S1 and then S2) in spatially distinct regions of the network as shown in **Figure 2A**. Each memory sequence was represented by the spatio-temporal pattern of 5 sequentially activated groups of 10 neurons per group. A 5 ms delay was included between stimulations of subsequent groups within a sequence. S1 was trained in the population of cortical neurons 200–249 (**Figure 2B**, top). Training S1 resulted in an increase of synaptic weights between participating neurons (**Figure 2D**, left) and an increase in performance on sequence completion (**Figure 2B/C**, top). When the strength of the synapses in the direction of S1 increased, synapses in the opposite direction showed a reduction consistent with the STDP rule (see *Methods and Materials*). The second sequence, S2, was trained for an equal amount of time as S1

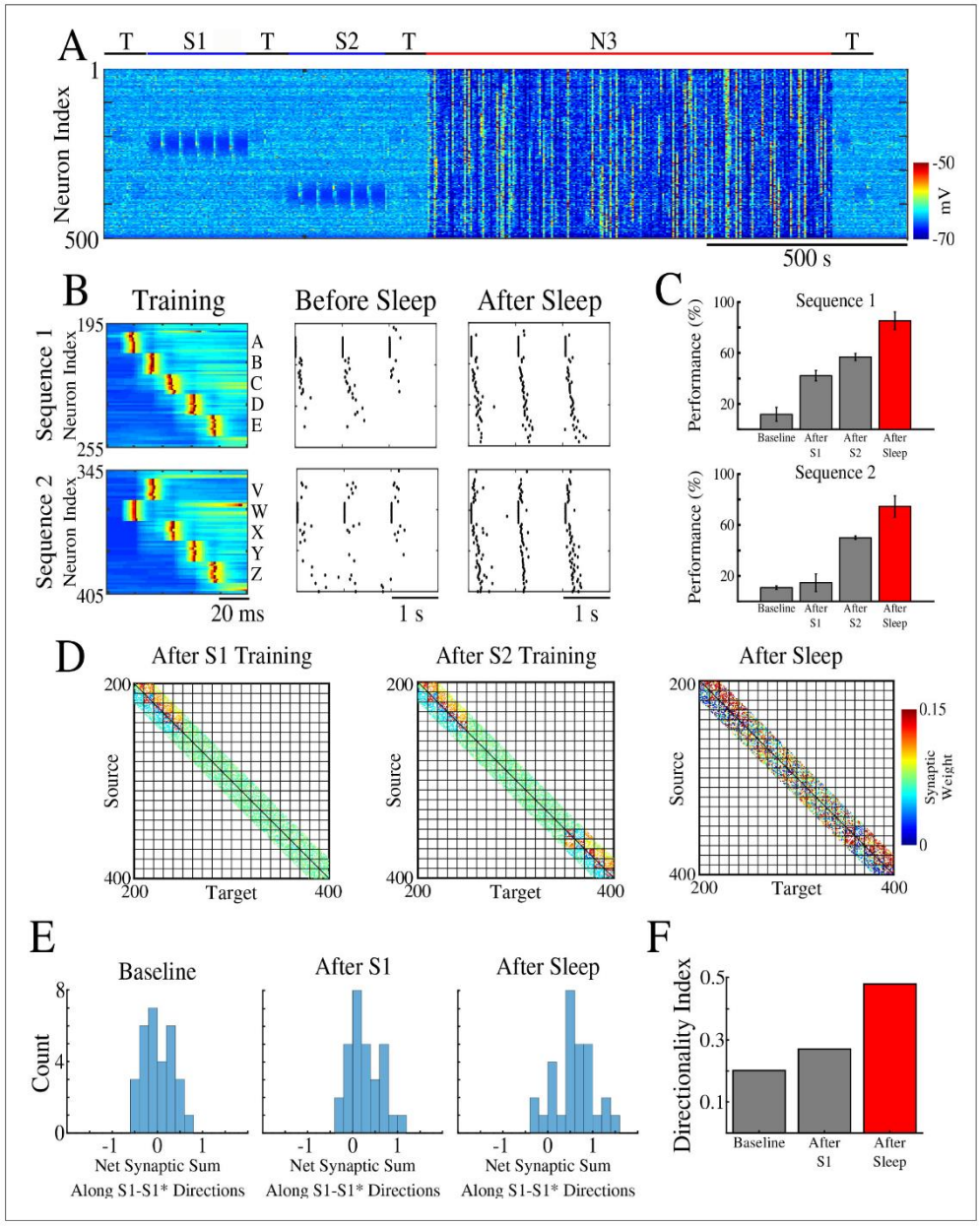


Figure 2. Two spatially separated memory sequences show no interference during training and both are strengthened by subsequent sleep. (**A**) Network activity during periods of testing (T), training of two spatially separated memory sequences (S1/S2), and sleep (N3). Cortical PY neurons are shown. Color indicates voltage of neurons at a given time. (**B**) Left panels show an example of training sequence 1 (S1, top) and sequence 2 (S2, bottom). Middle panels show examples of testing both sequences prior to sleep. Right panels show examples of testing after sleep. Note, after sleep,

Figure 2 continued on next page

Figure 2 continued

both sequences show better completion. (**C**) Performance of S1 and S2 completion before any training (baseline), after S1 training, after S2 training, and after sleep (red). (**D**) Synaptic weight matrices show changes of synaptic weights in the regions trained for S1 and S2. Left panel shows weights after training S1; middle panel shows weights after training S2; right panel shows weights after sleep. Color indicates strength of AMPA synaptic connections. (**E**) Distributions of the net sum of synaptic weights each neuron receives from all the neurons belonging to its left neighboring group (S1 direction) vs its right neighboring group (opposite direction, defined as S1* direction below) within a trained region at baseline (left), after S1 training (middle) and after sleep (right). (**F**) Synaptic weight-based directionality index before/after training (gray bars) and after sleep (red bar).
The online version of this article includes the following figure supplement(s) for figure 2:

Figure supplement 1. Sleep replay improves performance for complex non-linear sequences.

but in a different population of neurons 350–399 (W-V-X-Y-Z, ***Figure 2B***, bottom). Training of S2 also resulted in synaptic weight changes (***Figure 2D***, middle) and improvement in performance (***Figure 2B/C***, bottom). Importantly, training of S2 did not interfere with the weight changes encoding S1 because both sequences involved spatially distinct populations of neurons (compare ***Figure 2D***, left and middle). It should be noted that though testing resulted in reactivation of memory traces, there was little change in synaptic weights during testing periods because of a relatively small number of pre/post spike events. (Simulations where STDP was explicitly turned off during all testing periods exhibited similar results to those presented here.)

We next calculated the net sum of synaptic weights each neuron received from all neurons belonging to its left vs right neighboring populations (e.g., total input to a neuron B_i, belonging to group B, that it received from all the neurons in group A vs all the neurons in group C) and we analyzed the difference of these net weights. The initial distribution was symmetric reflecting the initial state of the network (***Figure 2E***, left). After training, it became asymmetric, indicating stronger input from the left groups (i.e., total input to B_i from all the neurons in group A was larger than that from all the neurons in group C) (***Figure 2E***, middle). These results are consistent with *in vivo* recordings from a rat running in one direction on a linear track (***Mehta et al., 1997***), where this phenomenon was called 'receptive field backwards expansion', i.e., neurons representing locations along the track became asymmetrically coupled such that activity in one group of neurons (one location) led to activation of the next group of neurons (new location) even before the corresponding input occurred (before the animal moved to the new location).

After successful training of both sequences, the network went through a period of sleep (N3 in ***Figure 2A***) when no stimulation was applied. After sleep, synaptic weights for both memory sequences revealed strong increases in the direction of their respective activation patterns and further decreases in the opposing directions (***Figure 2D***, right). In line with our previous work (***Wei et al., 2018***), these changes were a result of sequence replay during the Up states of slow oscillation (see next section for details). Synaptic strengthening increased the performance on sequence completion after sleep (***Figure 2B***, right; 2C, red bar). Analysis of the net synaptic input to each neuron from its left vs right neighboring groups, revealed further shift of the synaptic weight distribution (***Figure 2E***, right). This predicts that SWS following linear track training would lead to further receptive field backwards expansion in the cortical neurons. To quantify this asymmetry we calculated a 'directionality index', *I*, for synaptic weights (similar to ***Navratilova et al., 2012*** but using synaptic weights), based on synaptic input to each neuron from its left vs right neighboring populations (*'Directionality Index'=0* if all the neurons receive the same input from its left vs right neighboring groups and *'Directionality Index'=1* if all the neurons receive input from one 'side' only; see *Methods and Materials* for details). This analysis showed an increase in the directionality index from naive to trained cortical networks and further increase after sleep (***Figure 2F***). Note, that the backwards expansion of the place fields was reset between sessions in CA1 (***Mehta et al., 1997***), but not in CA3 (***Roth et al., 2012***), where the backward shift gradually diminished across days, possibly as memories became hippocampus independent (see *Discussion*).

The goal of this study was to reveal basic mechanisms of replay and therefore we focus on the 'simple' linear (e.g., S1) memory sequences. Our results, however, can be generalized to much more complex non-linear sequences (see ***Figure 2—figure supplement 1***). In simulations from ***Figure 2—figure supplement 1***, training a sequence in awake was not long enough to ensure reliable pattern completion, however, performance was significantly improved after replay during SWS.

Sleep replay improves pattern completion performance for memory sequences

Why do SWS dynamics lead to improvement in memory performance? The hypothesis is that memory patterns trained in awake are spontaneously replayed during sleep. With this in mind, we next analyzed the network firing patterns during Up states of the slow oscillation to identify replay. We focused our analysis on pairs of neurons (as opposed to the longer sequences) because (a) having different elementary units of a sequence (neuronal pairs) replayed independently would still be sufficient to strengthen the entire sequence; (b) *in vivo* data suggest that memory sequence replay often involves random subsets of the entire sequence (e.g., ***Euston et al., 2007***; ***Roumis and Frank, 2015***; ***Joo and Frank, 2018***; ***Swanson et al., 2020***); (c) we want to compare results in this section to the analysis of the overlapping opposite sequences in the following sections, however, we could not reliably detect replay of the full sequences in the latter case possibly because of highly overlapping spiking between sequences.

For each synapse in direction S1 (we refer to it below as S1 synapse) and each Up state, we (a) calculated the time delay between nearest pre/post spikes; (b) transformed this time delay through an STDP-like function to obtain a value characterizing its effect on synaptic weight; and (c) calculated the total net effect of all such spike events. This gave us a net weight change for a given synapse during a given Up state. If we observed a net weight increase, we labeled this S1 synapse as being preferentially replayed during a given Up state. Finally, we counted all the Up states where a given synapse was replayed as defined above. This procedure is similar to off-line STDP, however, instead of weight change over entire sleep, we obtained the number of Up states where a synapse in the direction of S1 was (preferentially) replayed.

Figure 3A shows, for each synapse in the direction of S1, the total change of its synaptic strength across entire sleep (Y-axis) vs number of Up states when that synapse was replayed (X-axis). As expected, it shows a strong positive correlation. Synaptic weight changes became negative when the number of Up states where an S1 synapse was replayed dropped below half of the total number of Up states (blue vertical line in ***Figure 3A***). In ***Figure 3B*** we plotted only those S1 synapses which were replayed reliably – for more that 66% of all Up states (dotted line in ***Figure 3A***). We found such synapses between all neuronal groups (gray boxes in ***Figure 3B***) as well as between neurons within groups.

In ***Figure 3C***, we illustrated all the synapses identified in the analysis in ***Figure 3B***, that is, synapses that were replayed reliably (in more than 66% of all Up states) in direction of S1. We also colored in blue neurons receiving at least one of these synapses as identified in ***Figure 3B***. We concluded that there were multiple direct and indirect synaptic pathways connecting the first (A) and last (E) groups of neurons that were replayed reliably during sleep. These synapses increased their strength which explains reliable memory recall during testing after sleep.

Sequential training of overlapping memory sequences results in interference

We next tested whether our network model shows interference during awake when a new sequence (S1*) (***Figure 4A***) is trained in the same population of neurons as the earlier old sequence (S1). S1* included the same exact groups of neurons as S1, but the order of activation was reversed, that is, the stimulation order was E-D-C-B-A (***Figure 4B***). S2 was once again trained in a spatially distinct region of the network (***Figure 4A/B***). Testing for sequence completion was performed immediately after each training period. This protocol can represent two somewhat different training scenarios: (a) two competing memory traces (S1 and S1*) are trained sequentially before sleep; (b) the first (old) memory S1 is trained and then consolidated during sleep followed by training of the second (new) memory S1* followed by another episode of sleep. We explicitly tested both scenarios and they behaved similarly, so in the following we discuss the simpler case of two sequentially trained memories followed by sleep. This setup can simulate *in vivo* experiments with a rat running on a belt in a VR apparatus, first in one direction only (learning S1) and then in the opposite direction (learning S1*). An example of the second scenario is presented in ***Figure 5—figure supplement 1*** and discussed below.

In the model, training S1 increased performance of S1 completion (***Figure 4C***, top/left). It also led to decrease in performance for S1* below its baseline level in the 'naive' network (***Figure 4C***,

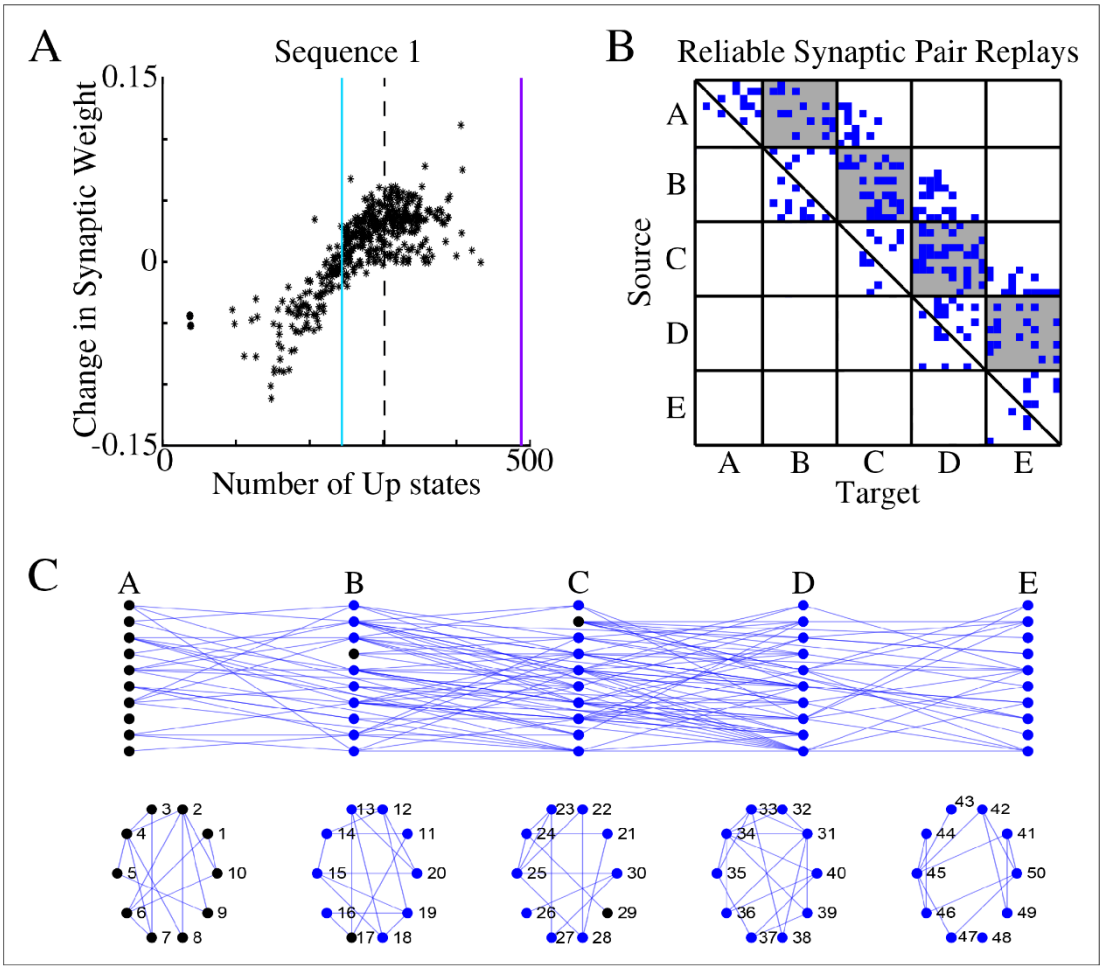


Figure 3. Sleep replay strengthens synapses to improve memory recall. (**A**) Change in synaptic weight over entire sleep period as a function of the number of Up states where a given synapse was replayed. Each star represents a synapse in the direction of S1. Dashed line indicates the threshold (66% of Up states) used to identify synapses that are replayed reliably for analysis in B; purple line indicates the maximum number of Up states; blue line demarcates the 50% mark of the total number of Up states. (**B**) Thresholded connectivity matrix indicating synaptic connections (blue) showing reliable replays in the trained region. Grey boxes highlight between group connections. (**C**) Network's graph showing between group (top) and within group (bottom) connections. Edges shown here are those synapses which revealed reliable replays of S1 as shown in B. Nodes are colored blue if they receive at least one of the synapses identified in panel B.

bottom/left). (Note that even a naive network displayed some above zero probability to complete a sequence depending on the initial strength of synapses and spontaneous network activity). Training S2 led to an increase in S2 performance (S1 performance also increased, most-likely due to the random reactivation of S1 in awake). Subsequent training of S1* resulted in both a significant increase in S1* performance and a significant reduction of S1 performance (***Figure 4C***). To evaluate the impact of S1* training on S1 performance, we varied the duration of S1* (later memory) training (***Figure 4D***). Increasing the duration of S1* training correlated with a reduction of S1 performance

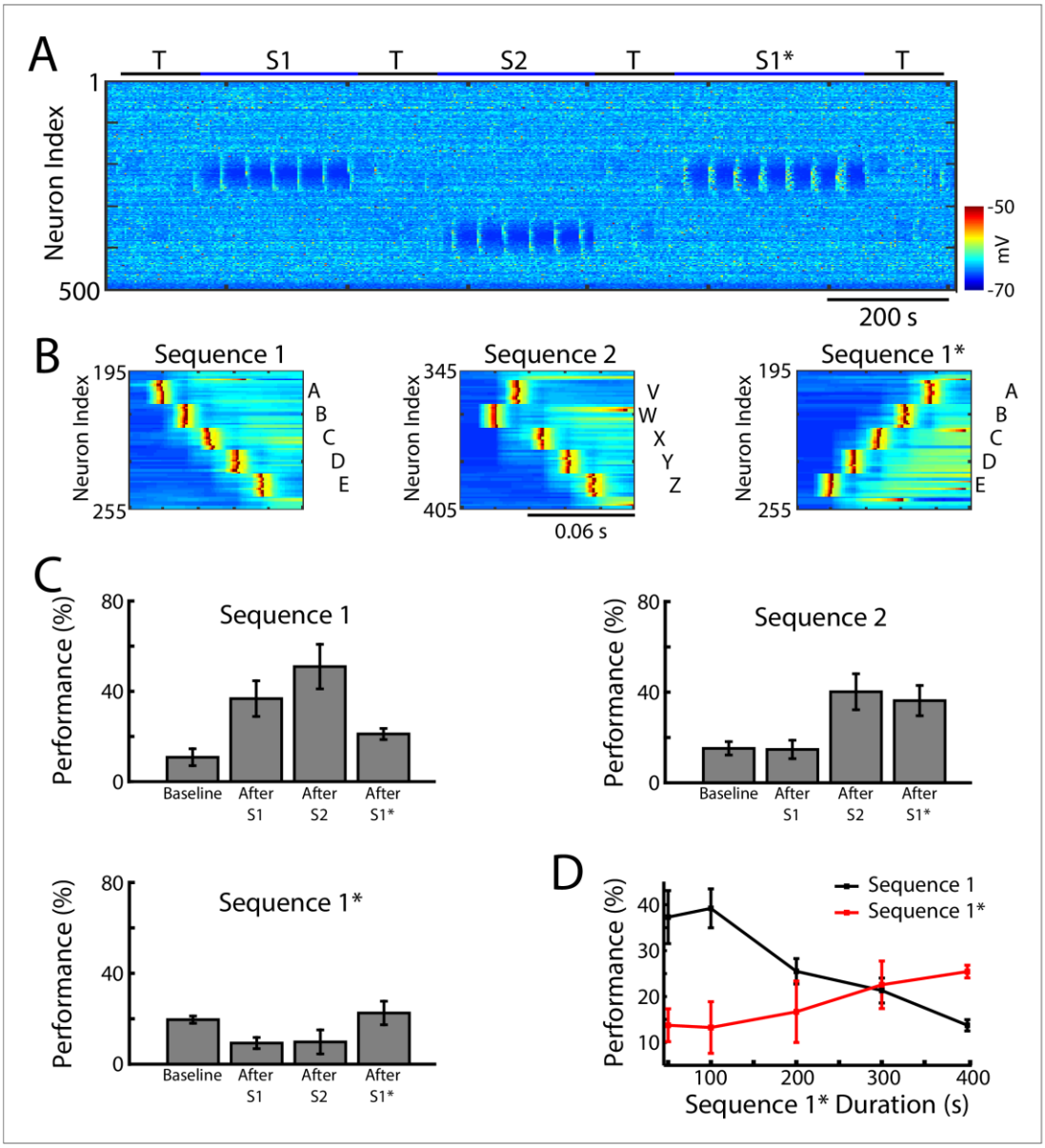


Figure 4. Training of overlapping memory sequences results in catastrophic interference. (**A**) Network activity (PY neurons) during training and testing periods for three memory sequences in awake-like state. Note, sequence 1 (S1) and sequence 1* (S1*) are trained over the same population of neurons. Color indicates the voltage of the neurons at a given time. (**B**) Examples of sequence training protocol for S1 (left), S2 (middle), and S1* (right). (**C**) Performances for the three sequences at baseline, and after S1, S2 and S1* training. Training of S1* leads to reduction of S1 performance. (**D**) Performance of S1 (black) and S1* (red) as a function of S1* training duration. Note that longer S1* training increases degradation of S1 performance.

Figure 4 continued on next page

Figure 4 continued

The online version of this article includes the following figure supplement(s) for figure 4:

Figure supplement 1. Interleaved training of the old and new memory sequences prevents the old sequence from forgetting and improves performance for both memories.

up to the point when S1 performance was reduced to its baseline level (***Figures 4D*** and 400 sec training duration of S1*). This suggests that sequential training of two memories competing for the same population of neurons results in memory interference and catastrophic forgetting of the earlier memory sequence.

The model predicts that in experiments with a rat running on a belt in a VR apparatus, training the backward direction after training the forward one would 'erase' the effect of the forward training. While we are not aware of such experiments, studies done with a rat running forward and backward on a liner track (***Navratilova et al., 2012***), which would be equivalent to interleaved training S1→ S1*→ S1→ S1*…, revealed that, in the hippocampus, spatial sequences of opposite direction are rapidly orthogonalized, largely on the basis of differential head direction system input, to accommodate both trainings. Thus, at each location, some neurons had their receptive field expanded in one direction and others in the opposite direction (***Navratilova et al., 2012***). To compare our model with these data, we tested interleaved training of S1 and S1* (***Figure 4—figure supplement 1***) and found performance increase for both sequences. Importantly, in agreement with *in vivo* data, different neurons became specific for S1 vs S1* as reflected in the overall increase of the directionality index (***Figure 4—figure supplement 1F***). In the next section we test if sleep can achieve the same goal.

Sleep prevents interference and leads to performance improvement for overlapping memories

So far we found that when a single sequence was trained, it replayed spontaneously during sleep resulting in improvement in performance (***Figures 2*** and ***3***). For two opposite sequences trained in the same network location we found competition and interference during sequential training in awake (***Figure 4***). However, when the same two sequences were trained using alternating protocol (interleaved training), both increased in performance (***Figure 4—figure supplement 1C***). We next tested the effect of SWS following sequential training of two opposite sequences in awake. Two outcomes are possible: (a) the stronger sequence could dominate replay and eventually suppress the weaker one, or (b) both sequences can be replayed during sleep and increase in performance after sleep. To test these possibilities, we simulated SWS (N3) after the sequences S1/S2/S1* were trained sequentially in the awake state (S1→ S1→…→ S2→ S2→…→ S1*→ S1*→…) (***Figure 5A***), as described in the previous sections (***Figures 2*** and ***4***). We stopped training the new memory S1* before the old memory trace S1 was completely erased (300 sec of S1* training, see ***Figure 4D***). Since we biased STDP towards LTP during awake, both memories S1 and S1* showed above baseline performance after training.

We found that sleep improves sequence completion performance for all three memories, including competing memory traces – S1 and S1*. ***Figure 5B*** shows raster plots of the spiking activity before vs after sleep, which revealed significant improvements in sequence completion. These results are summarized in (***Figure 5C***). Thus, we predict that sleep replay is not only able to reverse the damage caused to the old memory (S1) following S1* training, but it can enhance S1 performance at the same time as it enhances performance of S1*.

As for a single sequence, we next calculated the net sum of synaptic weights each neuron received from all the neurons belonging to its left vs right neighboring groups, and we analyzed the difference of these net weights. The initial distribution was symmetric reflecting the initial state of the network (***Figure 5D***, left). After S1 training, the distribution became asymmetric, indicating stronger input from the left (***Figure 5D***, middle/left). Training the opposite sequence, S1*, reversed the process and the distribution became more symmetric again, however, it also became wider with some neurons in each population preferring sequence S1 (i.e., for some group B neurons, B_i, input from group A was stronger than input from group C) and others preferring S1* (i.e., for other group B neurons, B_j, input from group C was stronger than input from group A) (***Figure 5D***, middle/right).

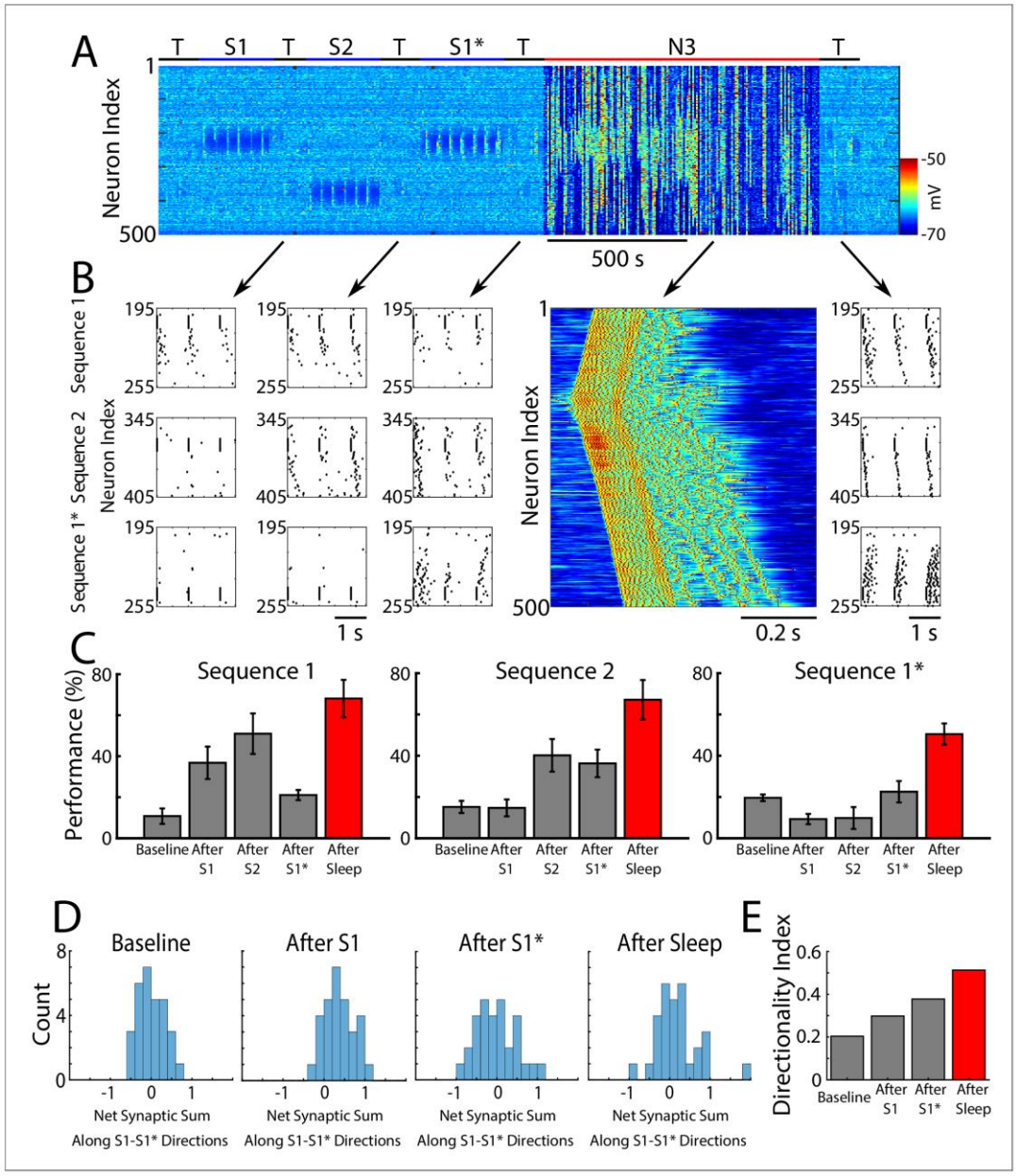


Figure 5. Sleep prevents the old memory sequence from forgetting and improves performance for all memories. (**A**) Network activity (PY neurons) during sequential training of sequencies S1/S2/S1* (blue bars) followed by N3 sleep (red bar). No stimulation was applied during sleep. (**B**) Examples of testing for each trained memory at different times. The top row corresponds to the testing of S1, middle is testing of S2, and bottom is testing of S1*. Heatmap shows characteristic cortical Up state during SWS. (**C**) Testing of S1, S2, and S1* shows damage to S1 after training S1*, and increase in

Figure 5 continued on next page

Figure 5 continued

performance for all three sequences after sleep (red bars). (**D**) Distributions of the net sum of synaptic weights each neuron receives from all the neurons belonging to its left vs right neighboring groups within a trained region at baseline (left), after training S1 (middle/left), after training S1* (middle/right), and after sleep (right). Wider distribution indicates presence of neurons that are strongly biased to one sequence or the other. (**E**) Synaptic weight-based directionality index before/after training (gray bars) and after sleep (red bar).
The online version of this article includes the following figure supplement(s) for figure 5:

Figure supplement 1. Training of a new memory that interferes with previously consolidated old memory leads to forgetting that can be reversed by subsequent sleep.

After SWS, the width of the distribution further increased indicating that sleep, similar to interleaved training, changes the network connectivity to develop neurons which become strongly specific for one sequence or another (***Figure 5D***, right). The synaptic weight-based directionality index that summarizes these changes (see above and *Methods and Materials* for details) also increased after sleep (***Figure 5E***).

Our study predicts that in experiments with a rat running on a belt in a VR apparatus, training the backward direction after training the forward one can damage (erase) the effect of forward training, however, SWS following training can reverse the damage. Additionally, similar to interleaved training (***Navratilova et al., 2012***), directionality index should increase after SWS.

As we mentioned previously, the training protocol we have focused on in this study was of two memories trained sequentially before sleep. We have also tested the scenario where the first (old) memory is trained and consolidated during sleep before the second (new) memory is trained and then consolidated during a second period of sleep (***Figure 5—figure supplement 1***). The main results from both training protocols remain the same. Thus, performance for S1 improved after first episode of sleep (initial consolidation) (***Figure 5—figure supplement 1B,C***). Training new memory S1* in the same population of neurons damaged S1 and led to improvement of S1*. Consistent with empirical results on proactive interference (***McDevitt et al., 2015***), training S1* took longer in that scenario to achieve a high level of performance. Note, that even longer training of S1* further improved its performance but could also completely erase S1 (***Figure 5—figure supplement 1D***). Finally, both S1 and S1* showed an improvement after a subsequent episode of sleep (***Figure 5—figure supplement 1B,C***). Thus, the training paradigm 'S1→ sleep→ S1*→ sleep' shows qualitatively similar results to the 'S1→ S1*→ sleep' paradigm. This result is also consistent with the 'Complementary Learning Systems Theory' prediction that the old memories interfering with new learning have to be replayed during new phase of memory consolidation to avoid forgetting (***McClelland et al., 2020***).

Competing memories are replayed spontaneously during Up states of slow oscillation

In this section we focus our analysis on the competing sequences S1 and S1*. We asked the following questions: (a) What kind of network dynamics during Up states of SWS allows for replay and improvement of both memory traces S1 and S1*? (b) Do the same neurons participate in replay of both sequences or do different subsets of neurons uniquely represent each memory? (c) Do both memory sequences replay during the same Up state or do different Up states become biased for replay of one memory or the other?

We performed spike timing analysis similar to what we did for S1 alone (***Figure 3***), but we now analyzed separately synaptic connections in direction of S1 and S1*. ***Figure 6A*** plots, for each synapse in direction of S1 (left) and S1* (right), the net change in synaptic strength across the entire sleep period vs total number of Up states (slow-waves) where that synapse was preferentially replayed. As before, we found a strong positive correlation. We next plotted only those synapses which replayed reliably – more that 66% of all Up states (***Figure 6B***). We found that such synapses exist between all neuronal groups and for both sequences (in ***Figure 6B*** blue color indicates synapses in the direction of S1 and red in the direction of S1*). This analysis revealed two important properties. First, after sleep, each pair of neurons preferentially supported only one sequence, S1 or S1* (note that the connectivity matrix in ***Figure 6B*** is strictly asymmetric). Second, individual neurons can be divided into two groups - those participating reliably in only one sequence replay (either S1 or

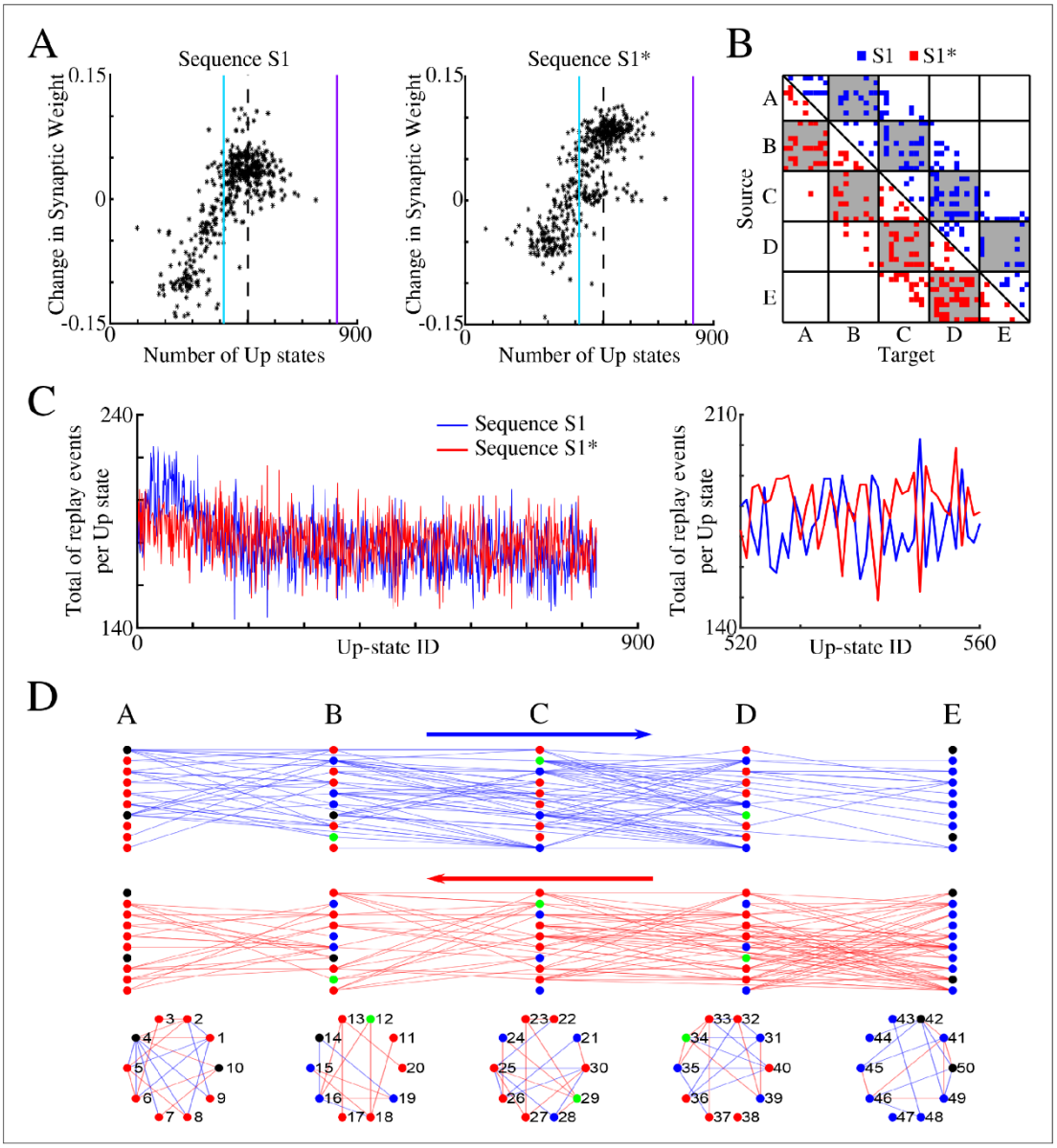


Figure 6. Sleep promotes replay of both overlapping memory sequences during each Up state. (**A**) Change in synaptic weight over entire sleep period as a function of the number of Up states where a given synapse was preferentially replayed. Each star represents a synapse in the direction of S1 (left) or S1* (right). Dashed line indicates the threshold (66% of Up states) used to identify synapses that are replayed reliably for analysis in (B); purple line indicates the maximum number of Up states; blue line demarcates the 50% mark of the total number of Up states. (**B**) Thresholded connectivity matrix indicating synaptic connections showing reliable replays for S1 (blue) or S1* (red). Grey boxes highlight between group connections. (**C**) Number of replay events for inter-group synapses per Up state across all Up states (left) and a subset of Up states (right) for S1 (blue) and S1* (red). Note that

Figure 6 continued on next page

Figure 6 continued

both sequences show similar high number of replays across all Up states, suggesting that both sequences are replayed during each Up state. (**D**) Network's graphs showing between group (top/middle) and within group (bottom) connections after sleep. Edges shown here are those which revealed reliable replays of S1 (blue) and S1* (red) as shown in **B** (right). Nodes are colored blue (red) if more than 50% of their incoming connections show reliable replay in direction of S1 (S1*). Green nodes indicate neurons with high in-degrees, receiving the same number of 'replayed' synapses from left and right, and black indicates that none of these conditions are met.

S1*) and those participating in both sequences replays (see ***Figure 6B***, where some target neurons (X-axis) receive input from source neurons (Y-axis) in only one network 'direction', left (blue) or right (red), and others receive input from both 'directions').

To confirm that both memories are replayed within the same Up state (i.e., some synapses replay S1 and others replay S1* during a given Up state), we counted, for each Up state, the total number of individual replay events across all synapses that were identified to replay S1 and S1* reliably (***Figure 6C***). This revealed fluctuations from one Up state to another, but the count remained high for both S1 and S1* confirming our prediction that partial replays of both sequences occur during the same Up state, that is, any given Up state participates in replay of both memories. Still, zoom-in to the replay count diagram (***Figure 6C***, right) revealed an antiphase oscillation, that is, one Up state would replay more S1 synapses, while another one (commonly next one) would replay more S1* synapses. Note, our model predicts that partial sequences (specifically spike doubles) of both memories can be replayed during *the same Up state* and not that both are replayed *simultaneously* (at the same exact time). Comparing replays during first vs second half of an Up state, we found that more replay events happened during the first half of any given Up state (particularly near the Down to Up transition) compared to the second half (not shown). This result is consistent with electrophysiological data suggesting that memory replay is strongest at the Down to Up state transition (***Johnson et al., 2010***).

Finally, in ***Figure 6D***, we plotted all the synapses identified by the analysis in ***Figure 6B***, that is, those involved in reliable (in more than 66% of all Up states) replay during sleep: top plot shows synapses in S1 direction (in blue) and bottom one shows synapses in S1* direction (in red). For each neuron we compared the number of such synapses it received from its left (S1 direction) vs right (S1* direction) neighboring population (e.g., for a neuron in group B, we compared if it received more synapses demonstrating reliable replay from group A or from group C). We then colored in blue (red) neurons receiving more synapses demonstrating reliable replay from its left (right) neighbors (***Figure 6D***). In green we colored neurons receiving the same number of 'replayed' synapses from left and right. While we found that many neurons (blue or red) participated reliably in only one sequence replay, S1 or S1*, a few neurons (green) participated equally in replay of both sequences, creating 'network hubs'.

Sleep replay leads to competition between synapses

In order to further understand how sleep replay affects S1 and S1* memory traces to allow enhancement of both memories, we next analyzed the dynamics of individual synaptic weights within the population of neurons containing the overlapping memory sequences (i.e. neurons 200–249). ***Figure 7A*** shows distributions of synaptic weights for synapses in the direction of S1 (top row) and in the direction of S1* (bottom row) before (blue) and after (red) specific events. Different columns correspond to different events, i.e. after S1 training (***Figure 7A***, left), after S1* training (***Figure 7A***, middle), after sleep (***Figure 7C***, right). Prior to any training, synaptic weights in the direction of either memory sequence were Gaussian distributed (***Figure 7A***, blue histogram, left). After S1 training, the weights for S1 strengthened (shifted to the right), while the weights for S1* weakened (shifted to the left). As expected, this trend was reversed when S1* was trained (***Figure 7A***, middle). After sleep, for each sequence (S1 or S1*) there was a subset of synapses that were further strengthened, while the rest of synapses were weakened (***Figure 7A***, right). This suggests that sleep promotes competition between synapses, so that specific subsets of synapses uniquely representing each memory trace can reach the upper bound to maximize recall performance while other synapses would become extinct to minimize interference.

Because of the random 'anatomical' connectivity, the cortical network model included two classes of synapses: *recurrent/bidirectional*, when a pair of neurons (e.g., n1 and n2) are connected by

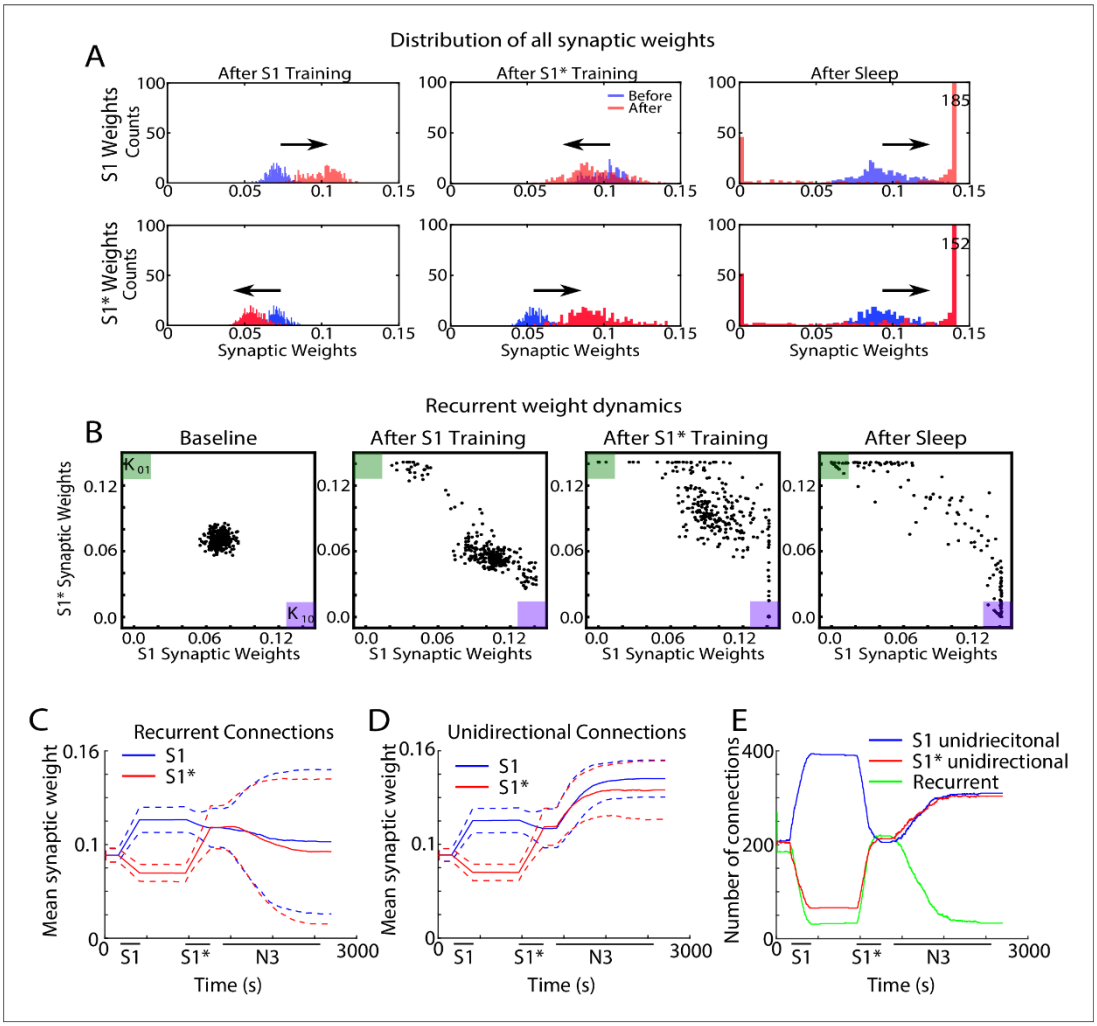


Figure 7. Sleep promotes unidirectional synaptic connectivity with different subsets of synapses becoming specific to the old or new memory sequences. (**A**) Dynamics of synaptic weight distributions from the trained region. Top row shows strength of synapses in direction of S1. Bottom row shows strength of synapses in direction of S1*. Blue shows the starting points for weights, and red shows new weights after different specific events, for example, S1 training, S1* training, sleep. (**B**) Scatter plots show synaptic weights for all reciprocally connected pairs of neurons before and after training (left/middle) and after sleep (right). For each pair of neurons (e.g., n1-n2), the X-coordinate shows the strength of $W_{n1\rightarrow n2}$ synapse and the Y-coordinate shows the strength of $W_{n2\rightarrow n1}$ synapse. The green (K_{01}) and purple (K_{10}) boxes show the locations in the scatter plot representing synaptic pairs with strong preference for S1* (green) or S1 (purple). (**C**) The evolution of the mean synaptic strength (solid line) and the standard deviation (dashed line) of recurrent connections in S1 (blue) and S1* (red) direction. Note the large standard deviation after sleep indicating strong synaptic weight separation, so each recurrent neuronal pair supports preferentially either S1 or S1*. (**D**) The evolution of the mean synaptic weight (solid line) and the standard deviation (dashed line) of unidirectional connections in S1 (blue) and S1* (red) direction. Note the overall increase in synaptic strength after sleep. (**E**) The number of functionally recurrent and unidirectional connections in the trained region of the network as a function of time, obtained after thresholding the network connectivity matrix with threshold 0.065 (which is smaller than the initial mean synaptic strength). Note the decrease of functionally recurrent connections and increase of functionally unidirectional connections after sleep.

Figure 7 continued on next page

Figure 7 continued

The online version of this article includes the following figure supplement(s) for figure 7:

Figure supplement 1. Interleaved training revealed synaptic weight dynamics that are similar to sleep but result in less segregation of synaptic weights.

Figure supplement 2. Synaptic plasticity that is biased towards LTP or LTD also results in memory orthogonalization during sleep .

opposite synapses (n1→ n2 *and* n2→ n1) and *unidirectional* (n1→ n2 *or* n2→ n1). In the following we looked separately at these two classes. We also compared synaptic weights dynamics during sleep (***Figure 7***) vs interleaved training (***Figure 7—figure supplement 1***).

In the scatter plots of synaptic weights for the recurrent synapses (***Figure 7B***), for each pair of neurons (e.g., n1-n2), we plotted a point with the X-coordinate representing the weight of n1→ n2 synapse and the Y-coordinate representing the weight of n2→ n1 synapse. Any point with X- (Y-) coordinate close to zero would, therefore, indicate a pair of neurons with functionally unidirectional coupling in S1* (S1) direction. The initial Gaussian distribution of weights (***Figure 7B***, left) was pushed towards the bottom right corner of the plot (K_{10}, purple box), indicating increases in S1 weights and relative decrease of S1* weights in response to S1 training (***Figure 7B***, middle/left). It should be noted that a small subset of synaptic weights increased in the direction of S1* during S1 training. Analysis of this population of synaptic weights revealed that these connections were comprised solely of 'within group' connections. It is important to note that these synapses did not impair the consolidation of the trained memory but instead helped to increase activity *within each group* regardless of which sequence was recalled.

Training of S1* caused an upward/left shift representing strengthening of S1* weights and weakening of S1 weights (***Figure 7B***, middle/right). For very long S1* training (not shown) almost all the weights would be pushed to the upper left corner (K_{01}, green box). Sleep appears to have taken most of the weights located in the center of the plot (i.e., strongly bidirectional synapses) and separated them by pushing them to the opposite corners (K_{01}, green box, and K_{10}, purple box) (***Figure 7B***, right). In doing so, sleep effectively converted recurrent connections into unidirectional connections which preferentially contributed to one memory sequence or another. It should be noted that interleaved training resulted in similar separation of weights such that some previously recurrent synapses became functionally unidirectional (***Figure 7—figure supplement 1A, B***). Interleaved training, however, retained more recurrent weights than sleep likely contributing to the smaller improvement in performance during post-interleaved training testing (***Figure 4—figure supplement 1C***).

Sleep-dependent synaptic weight dynamics are further illustrated in ***Figure 7*** panels C-E. The mean strength of all recurrent connections in the trained region decreased slightly during sleep (***Figure 7C***), however the standard deviation increased significantly (see dashed lines in ***Figure 7C***). The last reflected strong asymmetry of the connection strength for recurrent pairs after sleep, again indicating that sleep effectively converts recurrent connections into unidirectional ones. Indeed, the mean strength of all unidirectional connections increased during sleep (***Figure 7D***, blue and red lines). We next counted the total number of functionally recurrent and unidirectional connections after training and after sleep (***Figure 7E***). In this analysis if one branch of a recurrent pair reduced in strength below the threshold, it was counted as unidirectional. After sleep, the number of recurrent connections dropped to just about 15% of what it was after training. Interleaved training resulted in similar but smaller changes to unidirectional and bidirectional connections (***Figure 7—figure supplement 1C, D, E***). Together these results suggest that sleep decreases the density of recurrent connections and increases the density of unidirectional connections, both by increasing the *strength* of anatomical unidirectional connections and by *converting* anatomical recurrent connections to functionally unidirectional connections. This allows the assignment of individual neurons to unique memories, that is, orthogonalization of memory representations, so that multiple memories could replay without interference during the same Up states of slow oscillations and can be recalled successfully after sleep.

LTP or LTD biased synaptic plasticity still leads to orthogonalization of memory representations during sleep

In all previous simulations, LTP and LTD were balanced during sleep and interleaved training. To test that the orthogonalization of the memory traces during sleep is independent of the specific balance of LTP/LTD (A_+/A_-, see *Methods and Materials*), we performed additional simulations biasing the LTP/LTD ratio during sleep towards either LTD (***Figure 7—figure supplement 2A***; A_+/A_-=0.0019/0.002) or LTP (***Figure 7—figure supplement 2B***; A_+/A_-=0.0021/0.002). We found that in both cases, sleep resulted in the orthogonalization of memory representations. Scatter plots of bidirectional synaptic connections (the same analysis as in ***Figure 7B***) revealed that sleep formed strongly memory specific configurations of weights by pushing some of the recurrent connections to either the top left (S1* preferential) or bottom right (S1 preferential) corners of the plot. The red lines on these plots depict the threshold used to identify neuronal pairs that are either strongly preferential for S1 (bottom right corner) or S1* (top left corner). The number of synapses above these thresholds were quantified in the bar plots below showing that sleep increases the density of the memory specific connections between neurons regardless of the LTP/LTD ratio (***Figure 7—figure supplement 2***, bottom panels). The vector field plots (***Figure 7—figure supplement 2***, middle panels) provide a summary of the average synaptic weight dynamics during training (left and middle plots) and during sleep (right plot). It revealed convergence towards the corners (note arrows pointing to the corners during sleep phase) which represent cell pairs being strongly enrolled either to sequence S1 or sequence S1* encoding.

It should be noted that because our model does not have homeostatic mechanisms to regulate 'average' synaptic strength during sleep, the case of LTD biased sleep revealed a net reduction of synaptic strengths, while the LTP biased condition showed a net increase. For LTD biased sleep, many recurrent synapses decreased the strength while a fraction of synapses kept or even increased the strength. These synapses became memory specific after sleep. This observation may be in line with ideas from Tononi and colleagues showing net reductions of synaptic weights during sleep (***Tononi and Cirelli, 2014***) however, more analysis of the model including additional homeostatic rules is need to make this conclusion based on model simulations.

Neurons participating in sleep replay are the same as those responding earlier during memory recall

In the previous sections, we found that for overlapping memories sleep leads to segregation of the entire population of neurons into two subsets based on (a) asymmetric synaptic input from left/right neighboring groups (e.g., subset B_i of neurons from group B receives stronger total synaptic input from group A compare with total input from group C; subset B_j of neurons from group B receives stronger input from C than from A) (***Figure 5D,E***); (b) preference to participate reliably in only one specific sequence replay during sleep (e.g., subset B_k of neurons from group B receives more synapses demonstrating reliable replay from group A than from group C; this is reversed for subset B_l of neurons from group B) (***Figure 6D***). Here we tested if these groups of neurons, identified by synaptic strength and replay, overlap. We also compared them to the subset of neurons responding earliest within each group during memory recall.

Instead of stimulating only groups A or E, here we stimulated independently every single group - A, B, C, D, E (***Figure 8A***). We then obtained the response delay for each neuron in groups B, C, D when its respective left vs right neighboring groups were stimulated, and we calculated the difference of delays. Thus, for example, we measured a difference of response delays for each neuron in group B when either group A or group C was stimulated. This analysis is similar to what was done in (***Navratilova et al., 2012***), where the difference of place cell responses at a specific location on a linear track was calculated when a rat was approaching that location from one direction vs the other. ***Figure 8B*** shows the distribution of delays at different times. As expected, it became asymmetric after S1 training (e.g., in group B more neurons responded earlier upon stimulation of group A vs stimulation of group C), symmetric again after S1* training, and finally symmetric but wider after sleep. The last suggests that sleep increases segregation of neurons into two groups specific to each memory based on response delay (e.g., in group B some neurons, B_n, responded earlier upon stimulation of group A vs stimulation of group C; while other neurons, B_m, responded earlier upon stimulation of group C vs stimulation of group A). Indeed, the directionality index based on delays

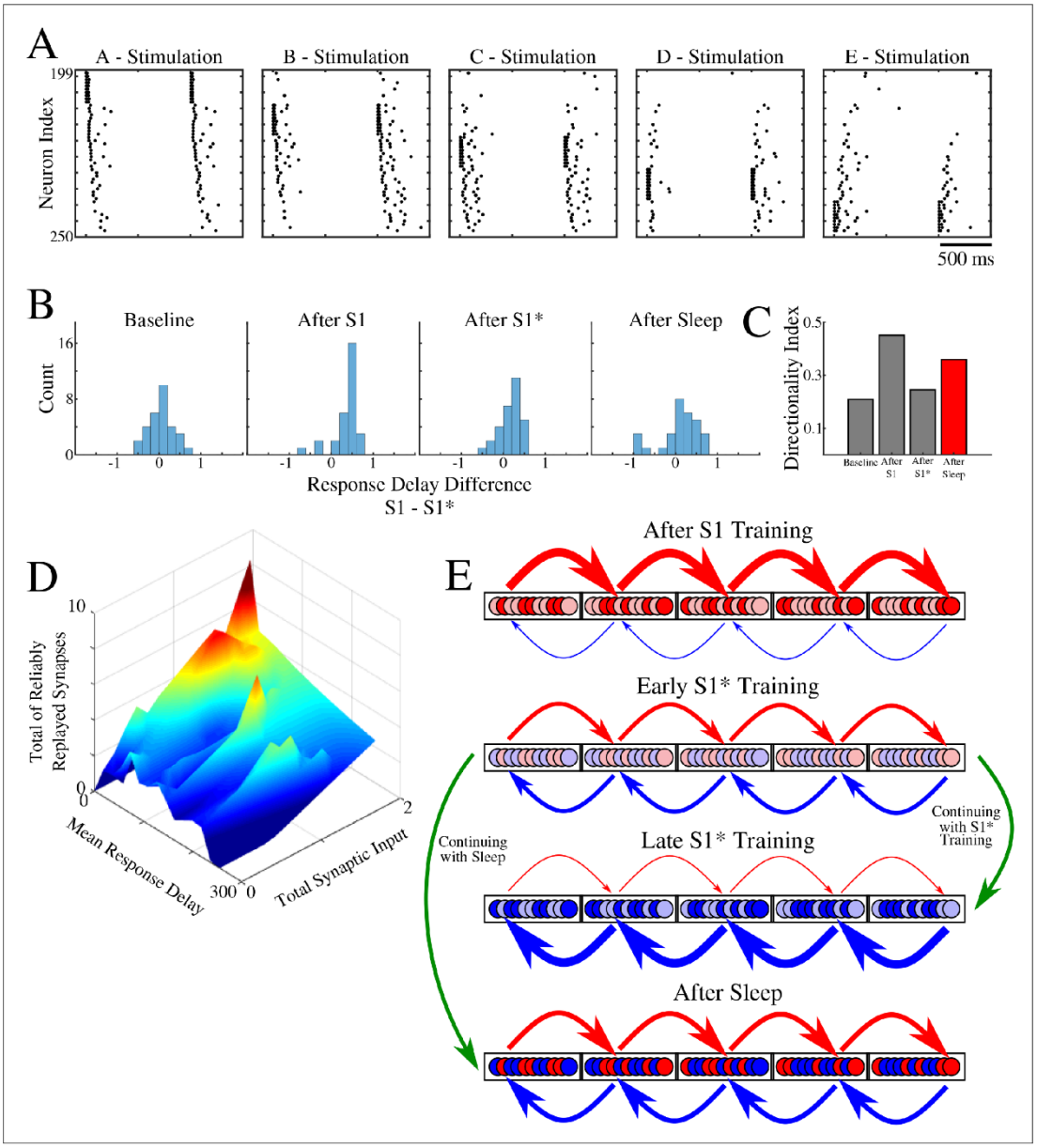


Figure 8. Population of neurons participating in reliable replay during sleep overlaps with the early responders during memory recall. (**A**) Characteristic examples of the network activity showing spiking events during stimulations of each individual 'letter' of a memory sequence in awake. (**B**) Distributions of the differences in response delay for all neurons from the trained region when the respective left vs right neighboring groups are stimulated, as shown in **A**. (**C**) Response delay-based directionality index before/after training of each sequence (S1/S1*) in gray, and after sleep (red). (**D**) 3-D surface plot showing, for each neuron from a trained region, the number of incoming synapses demonstrating reliable replays during sleep (z-axis), mean

Figure 8 continued on next page

Figure 8 continued

response delay during testing (as in panel A) after sleep (y-axis), and total synaptic input to a neuron after sleep (x-axis). Note that neurons receiving highest total synaptic inputs in specific network direction after sleep are also those who respond with shortest delay during testing recall in that direction after sleep and also those who receive the highest number of synapses demonstrating reliable replay during sleep. (**E**) Simplified cartoon of the network connectivity after different training phases followed by sleep. Arrows indicate connections between neurons (nodes) with blue arrows being connections strong for S1 and red for S1*. Blue and red nodes represent neurons that contribute (weakly - light colors; strongly - dark colors) to recall of S1 and S1*, respectively. Top, network configuration after S1 training – all nodes and connections are allocated to S1. Middle/Top, network configuration after initial S1* training – nodes/connections start to learn S1* and 'unlearn' S1. The information about the old memory S1 is still available. Middle/Bottom, network configuration after continuing S1* training – all nodes/connections are allocated to S1*. All information about S1 is lost. Bottom, network configuration when initial S1* training is followed by sleep – orthogonalization of memory traces, some nodes/connection are allocated to S1 and others to S1*.

(***Figure 8C***) revealed an increase after S1 training, drop after S1* training, and increase again after sleep.

In ***Figure 8D***, we summarized our results by putting together three main characteristics we discussed in this study: Mean response delay of a neuron during stimulation of its neighboring group, Total synaptic input a neuron receives from that neighboring group, and Number of connections to a neuron from that neighboring group that are replayed reliably during sleep. We found a strong correlation between these three measures, that is, the neurons who responded with a shortest delay during a given sequence recall after sleep are the same neurons who received strongest synaptic input in that sequence direction after sleep and were involved in most of that sequence replays during sleep.

Together, our study proposes the following network connectivity dynamics during learning and sleep (***Figure 8E***). Initial training allocates all available neuronal/synaptic resources to a single memory (S1) (***Figure 8E***, top); some neurons contribute stronger than others (light vs dart colors in ***Figure 8E***; based on ***Figure 7B***). Subsequent training of a competing memory (S1*) progressively erases the initial memory trace by reallocating synaptic resources to the new memory; an initial segregation of neurons is formed (***Figure 8E***, middle/top). Continuing training of a competing memory (S1*) leads to complete and irreversible damage to the old memory (S1) – catastrophic forgetting (***Figure 8E***, middle/bottom). A sleep phase implemented before the old memory is erased allows replay of both old and new memory traces; this divides resources between competing memories leading to the formation of the orthogonal memory representations which allows the co-existence of multiple memories within overlapping populations of neurons (***Figure 8E***, bottom).

Discussion

We report here that sleep can reverse catastrophic forgetting of previously learned (old) memories after damage by new training. Sleep is able to accomplish this task through spontaneous reactivation (replay) of both old and new memory traces, leading to reorganization and fine-tuning of synaptic connectivity. As a result, sleep creates unique orthogonal representations of the competing memories that allow their co-existence without interference within overlapping ensembles of neurons. Thus, if without competition, a memory is represented by the entire available population of neurons and synapses, in the presence of competition, its representation is reduced to a *subset* of neurons/synapses which selectively encode a given memory trace. Our study predicts that memory representations in the brain are dynamic; after each new episode of training followed by sleep, the synaptic representations of the old memories, sharing resources with the new task, may change to achieve an optimal separation among the memory traces occupying overlapping ensembles of neurons. Our study suggests that sleep, by being able to directly reactivate memory traces encoded in synaptic weight patterns, provides a powerful mechanism to prevent catastrophic forgetting and enable continual learning.

Catastrophic forgetting and continual learning

The work on catastrophic forgetting and interference in connectionist networks was pioneered by ***Mccloskey and Cohen, 1989*** and ***Ratcliff, 1990***. Catastrophic interference is observed when a previously trained network is required to learn new data, e.g., a new set of patterns. When learning new

data, the network can suddenly erase the memory of the old, previously learned inputs (**French, 1999**; **Hasselmo, 2017**; **Kirkpatrick et al., 2017**). Catastrophic interference is thought to be related to the so-called 'plasticity-stability' problem. This problem comes from the difficulty of creating a network with connections which are plastic enough to learn new data, while stable enough to prevent damage to the old memories. Due to the inherent trade-off between plasticity and memory stability, catastrophic interference and forgetting remains to be a difficult problem to overcome in connectionist networks (**French, 1999**; **Hasselmo, 2017**; **Kirkpatrick et al., 2017**).

A number of attempts have been made to overcome catastrophic interference (**French, 1999**; **Hasselmo, 2017**; **Kirkpatrick et al., 2017**). Early attempts included changes to the backpropagation algorithm, implementations of a 'sharpening algorithm' in which a decrease in the overlap of internal representations was achieved by making hidden-layer representations sparse, or changes to the internal structure of the network (**French, 1999**; **Hasselmo, 2017**; **Kirkpatrick et al., 2017**). These attempts were able to reduce the severity of catastrophic interference in specific cases but could not provide a complete and generic solution to the problem. Another popular method for preventing interference and forgetting is to explicitly retrain or rehearse all the previously learned inputs while training the network on the new data – interleaved training (**Hasselmo, 2017**). This idea recently led to a number of successful algorithms to constrain the catastrophic forgetting problem, including interleaved training focusing on the previously known items overlapping with new training data (**McClelland et al., 2020**), generative algorithms to generate previously experienced stimuli during the next training period (**Zz and Hoiem, 2018**; **van de Ven and Tolias, 2018**) and generative models of the hippocampus and cortex to generate examples from a distribution of previously learned tasks in order to retrain (replay) these tasks during a sleep phase (**Kemker and Kanan, 2017**).

In agreement with these previous studies, we show that interleaved training can prevent catastrophic forgetting resulted from sequential training of the overlapping spike patterns. This method, however, does not necessarily achieve optimal separation between old and new overlapping memory traces. Indeed, interleaved training requires repetitive activation of the entire memory patterns, so if different memory patterns compete for synaptic resources (as for the opposite sequences studied here) each phase of interleaved training will enhance one memory trace but damage the others. This is in contrast to replay during sleep when only memory specific subsets of neurons and synapses may be involved in each replay episode. Another primary concern with interleaved training is that it becomes increasingly difficult/cumbersome to retrain all the memories as the number of stored memories continues to increase and access to the earlier training data may no longer be available. As previously mentioned, biological systems have evolved a mechanism to prevent this form of forgetting without the need to explicitly retrain the network on all previously encoded memories. Studying how biological systems overcome catastrophic forgetting can provide insights into novel techniques to combat this problem in artificial neural networks.

Sleep and memory consolidation

Though a variety of sleep functions remain to be understood, there is growing evidence for the role of sleep in consolidation of newly encoded memories (**Paller and Voss, 2004**; **Walker and Stickgold, 2004**; **Oudiette et al., 2013**; **Rasch and Born, 2013**; **Stickgold, 2013**; **Weigenand et al., 2016**; **Wei et al., 2018**). The mechanism by which memory consolidation is influenced by sleep is still debated, however a number of hypotheses have been put forward. One such hypothesis is the 'Active System Consolidation Hypothesis' (**Rasch and Born, 2013**). Central to this hypothesis is the idea of repeated memory reactivation (**Wilson and McNaughton, 1994**; **Skaggs and McNaughton, 1996**; **Paller and Voss, 2004**; **Mednick et al., 2013**; **Oudiette et al., 2013**; **Oudiette and Paller, 2013**; **Rasch and Born, 2013**; **Stickgold, 2013**; **Weigenand et al., 2016**). Although NREM sleep was shown to be particularly important for reactivation of declarative (hippocampus-dependent) memories (**Marshall et al., 2006**; **Mednick et al., 2013**), human studies suggest that NREM sleep may be also involved in the consolidation of the procedural (hippocampus-independent) memories. This includes, for example simple motor tasks (**Fogel and Smith, 2006**), or finger-sequence tapping tasks (**Walker et al., 2002**; **Laventure et al., 2016**). Selective deprivation of NREM sleep, but not REM sleep, reduced memory improvement for the rotor pursuit task (**Smith and MacNeill, 1994**). Following a period of motor task learning, the duration of NREM sleep (**Fogel and Smith, 2006**) and the number of sleep spindles (**Morin et al., 2008**) increased. The amount of performance increase in the finger tapping task correlated with the amount of NREM sleep (**Walker et al., 2002**), spindle

density (*Nishida and Walker, 2007*) and delta power (*Tamaki et al., 2013*). In a recent animal study (*Ramanathan et al., 2015*), consolidation of the procedural (skilled upper-limb) memory depended on bursts of spindle activity and slow oscillations during NREM sleep.

Model predictions

The model of awake training and sleep consolidation presented in our new study was designed to simulate learning and consolidation of procedural memory tasks. Indeed, in our model, training a new task directly impacts cortical synaptic connectivity that may be already allocated to other (old) memories. We found that as long as damage to the old memory is not sufficient to completely erase its synaptic footprint, sleep can enable replay of both old and newer memory traces and reverse the damage while improving performance. Thus, to avoid irreversible damage, new learning in our model is assumed to be slow which may correspond to learning a procedural task, for example, new motor skill, over multiple days allowing sleep to recover old memory traces that are damaged by each new episodes of learning.

Nevertheless, we suggest that our model predictions, at least at the synaptic level, are not limited to a specific type of memory (declarative vs procedural) or specific type of sleep (NREM vs REM). Replay during REM sleep (*Louie and Wilson, 2001*) may trigger synaptic weight dynamics similar to that we described here. Though REM is characterized by less synchronized spiking activity, the occurrence of memory replay during REM is supported by place cell recordings (*Louie and Wilson, 2001*) and electroencephalography studies in humans (*Atienza and Cantero, 2001*). While synchronized activity is helpful for replay and may allow (because of high spike precision) for replay to occur at compressed time scales, as observed during NREM sleep (*Euston et al., 2007*), the crucial component of replay is the defined spike ordering which may be happening during REM sleep even when the overall network synchronization is low. Indeed, we observed similar synaptic weight dynamics and orthogonalization of memory representations when periodic Up/Down state oscillations were replaced by continuous REM-like spiking activity. While our model lacks hippocampal input, we showed previously (*Wei et al., 2016*; *Sanda et al., 2019*) that sharp wave-ripple (SWR) like input to the cortical network would trigger replay of previously learned cortical sequences during SWS. This suggests, in agreement with (*Skelin et al., 2019*), that replay driven by hippocampal inputs may reorganize the cortical synaptic connectivity in a matter similar to spontaneous replay we described here.

Our model predicts the possibility of the partial sequence replays, that is, when short snippets of a sequence are replayed independently, within the cortex. Furthermore, we showed that reliable partial replays of overlapping memory traces can occur during the same cortical Up state. That is to say, during an Up state rather than replaying the entire sequence A-B-C-D-E, we observed replay of individual transitions (e.g. A-B, D-E, C-D). We can speculate that for strongly overlapping sequences, as we modeled in this study, such partial replay would allow to replay snippets of both sequences with less interference during the same Up state. Indeed, recent data (*Ghandour et al., 2019*) have shown evidence for partial memory replay during NREM sleep (also see *Swanson et al., 2020*).

Importantly, our model of sleep consolidation predicts that the critical synaptic weight information from previous episodes of learning is still preserved after new training even if recall performance for the older task is significantly reduced. Because of this, spontaneous activity during sleep combined with unsupervised plasticity can trigger reactivation of the previously learned memory patterns and modify synaptic weights reversing damage from the new learning. It further suggests that the apparent loss of performance for older tasks in the artificial neuronal networks after new training – catastrophic forgetting – may not imply irreversible loss of information as it is generally assumed. Indeed, our recent work (*Krishnan et al., 2019*) revealed that simulating a sleep-like phase in feed-forward artificial networks trained using backpropagation can provide a solution for the catastrophic forgetting problem in agreement with our results from the biophysical model presented here. Few changes to the network properties, normally associated with transition to sleep, were critical to accomplish this goal: relative hyperpolarization of the pyramidal neurons and increasing strength of excitatory synaptic connectivity. Both are associated with known effects of neuromodulators during wake-sleep transitions (*McCormick, 1992*) and were previously implemented in the thalamocortical model (*Krishnan et al., 2016*) that we used in our new study. Interestingly, these changes would make neurons relatively less excitable and, at the same time, increase contribution of the strongest synapses, effectively enhancing the dynamical range for the trained synaptic patterns and reducing

contribution of synaptic noise; together this would promote replay of the previously learned memories.

The 'Sleep Homeostasis Hypothesis' (**Tononi and Cirelli, 2014**) suggests that homeostatic mechanisms active during sleep should result in a net synaptic depression to renormalize synaptic weights and to stabilize network dynamics. In our model, LTP and LTD were generally balanced during sleep and no homeostatic mechanisms were implemented to control net synaptic dynamics. However, when synaptic plasticity during sleep was explicitly biased towards LTD, sleep was still able to selectively increase a subset of synaptic weights, thus making them memory specific, while reducing the strength of other synapses. We predict that this mechanism may aid in increasing the memory capacity of the network by only strengthening the minimal number of connections required for the preservation of memories and resetting other synapses towards baseline strength during sleep. The network would then be able to use these synapses to encode new memories thus potentially facilitating continual learning without the consequence of retroactive interference.

Comparison to experimental data and model limitations

There are evidences that memory replay during SWS occurs predominantly near the Down to Up state transitions (**Johnson et al., 2010**). This observation comes from *in vivo* studies in which multiple brain regions, including the hippocampus and cortex, are in continual communication during SWS. It has been shown that sharp-wave ripples tend to occur at the Down to Up state transition (**Sanda et al., 2019**; **Skelin et al., 2019**), which may explain the predomiance of the hippocampus driven replay at the beginning of cortical Up states. We did not explicitly model the hippocampus or hippocampal inputs in our study. Rather, we assumed that memory traces are already embedded to the cortical connectivity matrix either because of the earlier hippocampal-dependent consolidation or because these memories are hippocampal-independent (as for procedural memories). We found that such cortical memory traces also tend to replay more during the initial phase of an Up state, possibly because of the higher firing rate, but replay continues throughout the entire Up state duration. This predicts that hippocampal dependent replay of new memories, that are not yet encoded in the cortex, may occur earlier in the Up state compared to the spontaneous replay of the old memory traces, which may occur later in the Up state.

Our results are consistent with *in vivo* experiments with rats running on a linear track and we make several specific predictions for future experiments. Specifically, our model predicts: (a) running in one direction on a linear track would lead to backwards receptive field expansion (confirmed for hippocampus [**Mehta et al., 1997**]); (b) forwards and backwards running on a linear track would lead to developing asymmetric receptive fields for different neurons (confirmed for hippocampus [**Navratilova et al., 2012**]); (c) running on a belt track in a VR apparatus first only in one direction and then in reverse one could damage the learning associated with first task; (d) SWS implemented after training would reverse damage and further enhance task specificity of neurons.

It is important to note that (**Mehta et al., 1997**) found that backwards expansion of the place fields was reset between sessions. Later, (**Roth et al., 2012**) found that the resetting of the backwards expanded place fields between sessions was a phenomenon specific to the CA1 and place fields did not reset in CA3. These results suggest that synapses in CA3 vs CA1 may have different plasticity properties. Furthermore, the neocortex may have entirely different synaptic dynamics since its goal is long term storage as opposed to temporal memory encoding. With successive sleep periods, cortical memories become hippocampal independent (**Lehmann and McNamara, 2011**) and this may explain why resetting of the place fields was observed in CA1 (**Mehta et al., 1997**). Our study predicts that the cortical (such as associate cortex) representations of the sequence memories undergo a similar form of backwards expansion as it was observed in CA1. This form of backwards expansion, however, persists and even increases after sleep.

The phenomena of backwards memory replay and decrease in number of memory replays over time have been observed in rat hippocampus for recent memories. Within the hippocampus, backwards replay is predominantly observed during a post-task awake resting period (**Foster and Wilson, 2006**). The studies of hippocampal replay (**O'Neill et al., 2008**; **Giri et al., 2019**) found decreases in replay of familiar sequences over time, which may occur because of the hippocampal SWRs inducing persistent synaptic depression within the hippocampus (**Norimoto et al., 2018**). We did not observe backwards replay; rather, forward replay in the model persisted during sleep. However, we believe there is no definitive evidence for either backwards replay or decrease in memory

replays in the cortex. The opposite, in fact, may be true. Cortical replay of recently formed memories results in a tagging of synapses involved in consolidation of those memories by increasing their synaptic efficacy (***Langille, 2019***). These tagged synapses may likely be reactivated throughout sleep thereby resulting in more cortical replay during both NREM and REM sleep (***Diekelmann and Born, 2010***; ***Langille, 2019***).

To summarize, our study predicts that sleep could prevent catastrophic forgetting and reverse memory damage through replay of old and new memory traces. By selectively replaying new and competing old memories, sleep dynamics not only achieve consolidation of new memories but also provide a mechanism for reorganizing the synaptic connectivity responsible for previously learned memories – re-consolidation of old memory traces – to maximize separation between memory representations. By assigning different subsets of neurons and synapses to primarily represent different memory traces, sleep effectively orthogonalizes memory representations to allow for overlapping populations of neurons to store competing memories and to enable continual learning.

Materials and methods

Thalamocortical network model

Network architecture

The thalamocortical network model used in this study has been previously described in detail (***Krishnan et al., 2016***; ***Wei et al., 2016***; ***Wei et al., 2018***) and the code is available in (https://github.com/o2gonzalez/sequenceLearningSleepCode; copy archived at https://github.com/elifesciences-publications/sequenceLearningSleepCode; ***González, 2020b***). Briefly, the network was comprised of a thalamus which contained 100 thalamocortical relay neurons (TC) and 100 reticular neurons (RE), and a cortex containing 500 pyramidal neurons (PY) and 100 inhibitory interneurons (IN). The model contained only local network connectivity as described in ***Figure 1***. Excitatory synaptic connections were mediated by AMPA and NMDA connections, while inhibitory synapses were mediated through $GABA_A$ and $GABA_B$. Starting with the thalamus, TC neurons formed AMPA connections onto RE neurons with a connection radius of 8 ($R_{AMPA(TC-RE)}$=8). RE neurons then projected inhibitory $GABA_A$ and $GABA_B$ connections onto TC neurons with $R_{GABA-A(RE-TC)}$=8 and $R_{GABA-B(RE-TC)}$=8. Inhibitory connections between RE-RE neurons were mediated by $GABA_A$ connections with $R_{GABA-A(RE-RE)}$=5. Within the cortex, PY neurons formed AMPA and NMDA connections onto PYs and INs with $R_{AMPA(PY-PY)}$=20, $R_{NMDA(PY-PY)}$=5, $R_{AMPA(PY-IN)}$=1, and $R_{NMDA(PY-IN)}$=1. PY-PY AMPA connections had a 60% connection probability, while all other connections were deterministic. Cortical inhibitory IN-PY connections were mediated by $GABA_A$ with $R_{GABA-A(IN-PY)}$=5. Finally, connections between thalamus and cortex were mediated by AMPA connections with $R_{AMPA(TC-PY)}$=15, $R_{AMPA(TC-IN)}$=3, $R_{AMPA(PY-TC)}$=10, and $R_{AMPA(PY-RE)}$=8.

Wake - Sleep transition

To model the transitions between wake and sleep states the model included synaptic and intrinsic mechanisms which reflect the changes in neuromodulatory tone during these different arousal states as previously described in ***Krishnan et al., 2016***. We included the effects of acetylcholine (ACh), histamine (HA), and GABA. ACh modulated potassium leak currents in all neuron types and excitatory AMPA connections within the cortex only. HA modulated the activation of the hyperpolarization-activated mixed cation current in TC neurons only, and GABA modulated the strength of inhibitory GABAergic synapses in both cortex and thalamus. As compared to the awake state, the levels of ACh and HA were reduced during NREM slow wave sleep, while the level of GABA was increased. This was done to reflect experimental observations of changes in the relative concentrations of ACh, HA, and GABA during different sleep stages (***Vanini et al., 2012***).

Intrinsic currents

All neurons were modeled with Hodgkin-Huxley kinetics. Cortical PY and IN neurons contained dendritic and axo-somatic compartments as previously described (***Wei et al., 2018***). The membrane potential dynamics were modeled by the following equation:

$$C_m\frac{dV_D}{dt} = -I_D^{Na} - I_D^{NaP} - I_D^{Km} - I_D^{KCa} - ACh_{gkl}I_D^{KL} - I_D^{HVA} - I_D^{L} - g(V_D - V_S) - I^{syn},$$

$$g(V_D - V_S) = -I_S^{Na} - I_S^{NaP} - I_S^{K},$$

where C_m is the membrane capacitance, $V_{D,S}$ are the dendritic and axo-somatic membrane voltages respectively, I^{Na} is the fast sodium (Na^+) current, I^{NaP} is the persistent Na^+ current, I^{Km} is the slow voltage-dependent non-inactivating potassium (K^+) current, I^{KCa} is the slow calcium (Ca^{2+})-dependent K^+ current, ACh_{gkl} represents the change in K^+ leak current I^{KL} which is dependent on the level of acetylcholine (ACh) during the different stages of wake and sleep, I^{HVA} is the high-threshold Ca^{2+} current, I^L is the chloride (Cl^-) leak current, g is the conductance between the dendritic and axo-somatic compartments, and I^{syn} is the total synaptic current input to the neuron (see next section for details). IN neurons contained all intrinsic currents present in PY with the exception of the I^{NaP}. All intrinsic ionic currents (I^j) were modeled in a similar form:

$$I^j = g_j m^M h^N (V - E_j)$$

where g_j is the maximal conductance, m (activation) and h (inactivation) are the gating variables, V is the voltage of the corresponding compartment, and E_j is the reversal potential of the ionic current. The gating variable dynamics are described as follows:

$$\frac{dx}{dt} = -\frac{x - x_\infty}{\tau_x},$$

$$\tau_x = \frac{(1/(\alpha_x + \beta_x))}{Q_T},$$

$$x_\infty = \frac{\alpha_x}{(\alpha_x + \beta_x)},$$

where $x = m$ or h, τ is the time constant, Q_T is the temperature related term, $Q_T = Q^{((T-23)/10)} = 2.9529$, with $Q = 2.3$ and $T = 36$.

Thalamic neurons (TC and RE) were modeled as single compartment neurons with membrane potential dynamics mediated by the following equation:

$$C_m\frac{dV_D}{dt} = -I^{Na} - I^{K} - ACh_{gkl}I^{KL} - I^{T} - I^{h} - I^{L} - I^{syn},$$

where I^{Na} is the fast Na^+ current, I^K is the fast K^+ current, I^{KL} is the K^+ leak current, I^T is the low-threshold Ca^{2+} current, I^h is the hyperpolarization-activated mixed cation current, I^L is the Cl^- leak current, and I^{syn} is the total synaptic current input to the neurons (see next section for details). The I^h was only expressed in the TC neurons and not the RE neurons. The influence of histamine (HA) on I^h was implemented as a shift in the activation curve by HA_{gh} as described by:

$$m_\infty = \frac{1}{1 + exp\left(\frac{V + 75 + HA_{gh}}{5.5}\right)}.$$

A detailed description of the individual currents can be found in our previous studies (***Krishnan et al., 2016***; ***Wei et al., 2018***).

Synaptic currents and spike-timing dependent plasticity (STDP)

AMPA, NMDA, and $GABA_A$ synaptic current equations were described in detail in our previous studies (***Krishnan et al., 2016***; ***Wei et al., 2018***). The effects of ACh on $GABA_A$ and AMPA synaptic currents in our model are described by the following equations:

$$I_{syn}^{GABA} = \gamma_{GABA_A}\, g_{syn}\, [O](V - E_{syn}),$$

$$I_{syn}^{AMPA} = ACh_{AMPA}\, g_{syn}\, [O](V - E_{syn}),$$

where g_{syn} is the maximal conductance at the synapse, $[O]$ is the fraction of open channels, and E_{syn} is the channel reversal potential ($E_{GABA\text{-}A}$ = -70 mV, E_{AMPA} = 0 mV, and E_{NMDA} = 0 mv). Parameter γ_{GABA_A} modulates the GABA synaptic currents for IN-PY, RE-RE, and RE-TC connections. For IN neurons γ_{GABA_A} was 0.22 and 0.44 for awake and N3 sleep, respectively; γ_{GABA_A} for RE was 0.6 and 1.2 for awake and N3 sleep. ACh_{AMPA} defines the influence of ACh levels on AMPA synaptic currents for PY-PY, TC-PY, and TC-IN. ACh_{AMPA} for PY was 0.133 and 0.4332 for awake and N3 sleep. ACh_{AMPA} for TC is 0.6 and1.2 for awake and N3 sleep.

Spontaneous miniature excitatory post-synaptic potentials (EPSPs) and inhibitory post-synaptic potentials (IPSPs) were implemented for PY-PY, PY-IN, and IN-PY connections. The synaptic dynamics were similar to regular post-synaptic potentials (PSPs) described above and their arrival times were modeled by a Poisson process with time-dependent mean rate, with next release time $t_{release}$ given by:

$$t_{release} = (2/(1 + exp(-(t - t_0)/\upsilon)) - 1)/250\,,$$

where t_0 is the time of the last presynaptic spike. The maximal conductances for miniature PSPs were $g_{mini(PY-PY)}^{AMPA} = 0.03\ \mu S$, $g_{mini(PY-IN)}^{AMPA} = 0.02\ \mu S$, and $g_{mini(IN-PY)}^{GABA} = 0.02\ \mu S$. υ is the mini PSP frequency: $\upsilon_{mini(PY-PY)}^{AMPA} = 30$, $\upsilon_{mini(PY-IN)}^{AMPA} = 30$, and $\upsilon_{mini(IN-PY)}^{GABA} = 30$. Short-term depression of intracortical AMPA synapses was included. The maximal synaptic conductance was multiplied by a depression variable ($D \leq 1$), which represents the amount of available 'synaptic resources' as described in **Bazhenov et al., 2002**. This short-term depression was modeled as follows:

$$D = 1 - (1 - D_i(1 - U))exp\left(-\frac{t - t_i}{\tau}\right)$$

where D_i is the value of D immediately before the i_{th} event, $(t - t_i)$ is the time after the i_{th} event, $U = 0.073$ is the fraction of synaptic resources used per action potential, and $\tau = 700ms$ is time constant of recovery of synaptic resources.

Potentiation and depression of synaptic weights between PY neurons were regulated by spike-timing dependent plasticity (STDP). The changes in synaptic strength (g_{AMPA}) and the amplitude of miniature EPSPs (A_{mEPSP}) have been described previously (**Wei et al., 2018**):

$$g_{AMPA} \leftarrow g_{AMPA} + g_{max}\, F(\Delta t),$$

$$A_{mEPSP} \leftarrow A_{mEPSP} + fA_{PY-PY}\, F(\Delta t),$$

where g_{max} is the maximal conductance of g_{AMPA}, and f = 0.01 represents the slower change of STDP on A_{mEPSP} as compared to g_{AMPA}. The STDP function F is dependent on the relative timing (Δt) of the pre- and post-synaptic spikes and is defined by:

$$F(\Delta t) = \begin{cases} A_+\, e^{-|\Delta t|/\tau_+}, & if\ \Delta t > 0 \\ -A_-\, e^{-|\Delta t|/\tau_-}, & if\ \Delta t < 0 \end{cases}$$

where $A_{+/-}$ set the maximum amplitude of synaptic change. $A_{+/-}$ = 0.002 and $\tau_{+/-}$ = 20 ms. A_- was reduced to 0.001 during training to reflect the effects of changes in acetylcholine concentration during focused attention on synaptic depression during task learning observed experimentally (**Bokland, 1995**; **Shinoe et al., 2005**; **Sugisaki et al., 2016**).

Sequence training and testing

Training and testing of memory sequences was performed similar to our previous study (**Wei et al., 2018**). Briefly, trained sequences were comprised of 5 groups of 10 sequential PY neurons. Each group of 10 were sequentially activated by a 10 ms DC pulse with 5 ms delay between subsequent group pulses. This activation scheme was applied every 1 s throughout the duration of the training period. Sequence 1 (S1) consisted of PY groups (in order of activation): A(200-209), B(210-219), C (220-229), D(230-239), E(240-249). Sequence 2 (S2) consisted of PY groups (in order of activation): W

(360-369), V(350-359), X(370-379), Y(380-389), Z(390-399) and can be referred as non-linear due to the non-spatially sequential activations of group W, V, and X. Sequence 1* (S1*) was trained over the same population of neurons trained on S1 but in the reverse activation order (i.e. E-D-C-B-A). During testing, the network was presented with only the activation of the first group of a given sequence (A for S1, W for S2, and E for S1*). Performance was measured based on the network's ability to recall/complete the remainder of the sequence (i.e. A-B-C-D-E for S1) within a 350 ms time window. Similar to training, test activation pulses were applied every 1 s throughout the testing period. Training and testing of the sequences occurred sequentially as opposed to simultaneously as in our previous study (***Wei et al., 2018***).

Data analysis

All analyses were performed with standard MatLab and Python functions. Data are presented as mean ± standard error of the mean (SEM) unless otherwise stated. For each experiment a total of 10 simulations with different random seeds were used for statistical analysis.

Sequence performance measure

A detailed description of the performance measure used during testing can be found in ***Wei et al., 2018*** and the code is available in (https://github.com/o2gonzalez/sequencePerformanceAnalysis; copy archived at https://github.com/elifesciences-publications/sequencePerformanceAnalysis; ***González, 2020a***). Briefly, the performance of the network on recalling a given sequence following activation of the first group of that sequence (see *Methods and Materials*: *Sequence training and testing*) was measured by the percent of successful sequence recalls. We first detected all spikes within the predefined 350 ms time window for all 5 groups of neurons in a sequence. The firing rate of each group was then smoothed by convolving the average instantaneous firing rate of the group's 10 neurons with a Gaussian kernel with window size of 50 ms. We then sorted the peaks of the smoothed firing rates during the 350 ms window to determine the ordering of group activations. Next, we applied a string match (SM) method to determine the similarity between the detected sequences and an ideal sequence (ie. A-B-C-D-E for S1). SM was calculated using the following equation:

$$SM = 2 * N - \sum_{i=1}^{N} |L(S_{test}, S_{sub}[i]) - i|,$$

where N is the sequence length of S_{test}, S_{test} is the test sequence generated by the network during testing, S_{sub} is a subset of the ideal sequence that only contains the same elements of S_{test}, and $L(S_{test}, S_{sub}[i])$ is the location of the element $S_{sub}[i]$ in sequence S_{test}. SM was then normalized by double the length of the ideal sequence. Finally, the performance was calculated as the percent of recalled sequences with SM$\geq$Th, where Th is the selected threshold (here, Th = 0.8) indicating that the recalled sequence must be at least 80% similar to the ideal sequence to be counted as a successful recall as previously done in ***Wei et al., 2018***.

Sequence replay during N3 sleep

To find out whether a trained sequence is replayed in the trained region of the network during the Up state of a slow-wave in N3 sleep, we first identified the beginning and the end of each Up state by considering sorted spike times of neurons in each group. For each group, the time instances of consecutive spikes that occur within a 15 ms window were considered as candidate members of an Up state, where the window size was determined to decrease the chance of two spikes of the same neuron within the window. To eliminate spontaneous spiking activity of a group that satisfies the above condition but is not part of an Up state, we additionally required that the period between two upstate was at least 300 ms, which corresponds to a cortical Down state. The values for window durations reported above were identified to maximize the performance of the Up state search algorithm.

Once all Up states were determined, we defined the time instances when groups were active in each Up state. A group was defined as active if the number of neurons from the group that spikes during 15 ms exceeded the activation threshold, and the instance when the group is active was defined as the average over spike times of a subgroup of neurons with the size equals to the

activation threshold within the 15 ms window. In our study the activation threshold was selected to be half of a group size (i.e. five neurons). Using sorted time instances when groups are active, we counted the number of times a possible transition between arbitrary groups, and if all four transitions of a sequence were observed sequentially in the right order then we counted that as a replay of the sequence.

Analysis of total sequence specific synaptic input

For every neuron from a group we computed the total synaptic weight 'from left' and 'to right', by considering the sum of all weights of synapses projecting to the neuron from neurons in preceding group, with respect to propagation of activity within a memory sequence if such a group exists, and the sum of all weights of synaptic connections from the neuron to the following group, if there is such a group. We omitted all synaptic connections within the group to which the neuron, for which the total synaptic weight is computed, belongs.

Weight directionality index

To see how learning recruits neurons in encoding one of the competing sequences, we looked at the evolution of deviation from the center of unit square in a two dimensional subspace of total synaptic input from left and right neighboring neuronal groups. For this, we first found the total synaptic input from both sides, embedded it into a unit square, and computed Euclidean distance from the center of the square.

$$Weight\ directionality\ index = \sqrt{(li-0.5)^2+(ri-0.5)^2},$$

where li (ri) is the total synaptic input to a neuron from its left (right) neighboring neuronal group.

Delay directionality index

To see whether neurons respond preferentially to one of the sequences, we evaluated signed (***Figure 8B***) and unsigned (***Figure 8C, D***) delay directionality indices, which are defined as follows. For each neuron, we found its response delays, t_{S1} and t_{S1*}, after corresponding left and right neighboring neuronal groups were stimulated, respectively. Using these quantities, we computed the indices as

$$signed\ directionality\ index = \frac{\Delta t_{S1*} - \Delta t_{S1}}{\Delta t_{S1*} + \Delta t_{S1}},$$

$$unsigned\ directionality\ index = \frac{|\Delta t_{S1*} - \Delta t_{S1}|}{\Delta t_{S1*} + \Delta t_{S1}}.$$

Acknowledgements

This work was supported by the Lifelong Learning Machines program from DARPA/MTO (HR0011-18-2-0021) and ONR (MURI: N00014-16-1-2829).

Additional information

Funding

Funder	Grant reference number	Author
Defense Advanced Research Projects Agency	HR0011-18-2-0021	Maxim Bazhenov
Office of Naval Research	MURI: N00014-16-1-2829	Maxim Bazhenov

The funders had no role in study design, data collection and interpretation, or the decision to submit the work for publication.

Author contributions

Oscar C González, Yury Sokolov, Conceptualization, Data curation, Software, Formal analysis, Investigation, Visualization, Methodology, Writing - original draft, Writing - review and editing; Giri P Krishnan, Conceptualization, Software, Methodology, Writing - original draft, Writing - review and editing; Jean Erik Delanois, Data curation, Software, Formal analysis, Investigation, Visualization, Methodology, Writing - original draft, Writing - review and editing; Maxim Bazhenov, Conceptualization, Supervision, Funding acquisition, Writing - original draft, Project administration, Writing - review and editing

Author ORCIDs

Oscar C González https://orcid.org/0000-0003-1302-1911
Yury Sokolov https://orcid.org/0000-0002-4590-3321
Giri P Krishnan http://orcid.org/0000-0002-3931-7633
Jean Erik Delanois https://orcid.org/0000-0002-8680-3239
Maxim Bazhenov https://orcid.org/0000-0002-1936-0570

Decision letter and Author response

Decision letter https://doi.org/10.7554/eLife.51005.sa1
Author response https://doi.org/10.7554/eLife.51005.sa2

Additional files

Supplementary files

- Transparent reporting form

Data availability

Computational models were used exclusively in this study. The model is fully described in the Methods section and code has been deposited to https://github.com/o2gonzalez/sequenceLearningSleepCode (copy archived at https://github.com/elifesciences-publications/sequenceLearningSleepCode).

References

Atienza M, Cantero JL. 2001. Complex sound processing during human REM sleep by recovering information from long-term memory as revealed by the mismatch negativity (MMN). *Brain Research* **901**:151–160. DOI: https://doi.org/10.1016/S0006-8993(01)02340-X, PMID: 11368962

Bazhenov M, Timofeev I, Steriade M, Sejnowski TJ. 2002. Model of thalamocortical slow-wave sleep oscillations and transitions to activated states. *The Journal of Neuroscience* **22**:8691–8704. DOI: https://doi.org/10.1523/JNEUROSCI.22-19-08691.2002, PMID: 12351744

Blake H, Gerard RW. 1937. Brain potentials during sleep. *American Journal of Physiology-Legacy Content* **119**: 692–703. DOI: https://doi.org/10.1152/ajplegacy.1937.119.4.692

Blokland A. 1995. Acetylcholine: a neurotransmitter for learning and memory? *Brain Research Reviews* **21**:285–300. DOI: https://doi.org/10.1016/0165-0173(95)00016-X, PMID: 8806017

Clemens Z, Fabó D, Halász P. 2005. Overnight verbal memory retention correlates with the number of sleep spindles. *Neuroscience* **132**:529–535. DOI: https://doi.org/10.1016/j.neuroscience.2005.01.011, PMID: 15802203

Diekelmann S, Born J. 2010. The memory function of sleep. *Nature Reviews Neuroscience* **11**:114–126. DOI: https://doi.org/10.1038/nrn2762, PMID: 20046194

Euston DR, Tatsuno M, McNaughton BL. 2007. Fast-forward playback of recent memory sequences in prefrontal cortex during sleep. *Science* **318**:1147–1150. DOI: https://doi.org/10.1126/science.1148979, PMID: 18006749

Fachechi A, Agliari E, Barra A. 2019. Dreaming neural networks: forgetting spurious memories and reinforcing pure ones. *Neural Networks* **112**:24–40. DOI: https://doi.org/10.1016/j.neunet.2019.01.006, PMID: 30735914

Fogel SM, Smith CT. 2006. Learning-dependent changes in sleep spindles and stage 2 sleep. *Journal of Sleep Research* **15**:250–255. DOI: https://doi.org/10.1111/j.1365-2869.2006.00522.x, PMID: 16911026

Foster DJ, Wilson MA. 2006. Reverse replay of behavioural sequences in hippocampal place cells during the awake state. *Nature* **440**:680–683. DOI: https://doi.org/10.1038/nature04587, PMID: 16474382

French RM. 1999. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences* **3**:128–135. DOI: https://doi.org/10.1016/S1364-6613(99)01294-2, PMID: 10322466

Ghandour K, Ohkawa N, Fung CCA, Asai H, Saitoh Y, Takekawa T, Okubo-Suzuki R, Soya S, Nishizono H, Matsuo M, Osanai M, Sato M, Ohkura M, Nakai J, Hayashi Y, Sakurai T, Kitamura T, Fukai T, Inokuchi K. 2019. Orchestrated ensemble activities constitute a hippocampal memory engram. *Nature Communications* **10**:2637. DOI: https://doi.org/10.1038/s41467-019-10683-2, PMID: 31201332

Giri B, Miyawaki H, Mizuseki K, Cheng S, Diba K. 2019. Hippocampal reactivation extends for several hours following novel experience. *The Journal of Neuroscience* **39**:866–875. DOI: https://doi.org/10.1523/JNEUROSCI.1950-18.2018, PMID: 30530857

González OC. 2020a. sequence Performancen Analysis. *GitHub*. 094c4be. https://github.com/o2gonzalez/sequencePerformanceAnalysis

González OC. 2020b. *sequence Learning SleepCode. GitHub*. a1eaace. https://github.com/o2gonzalez/sequenceLearningSleepCode

Hassabis D, Kumaran D, Summerfield C, Botvinick M. 2017. Neuroscience-Inspired artificial intelligence. *Neuron* **95**:245–258. DOI: https://doi.org/10.1016/j.neuron.2017.06.011, PMID: 28728020

Hasselmo ME. 2017. Avoiding catastrophic forgetting. *Trends in Cognitive Sciences* **21**:407–408. DOI: https://doi.org/10.1016/j.tics.2017.04.001, PMID: 28442279

Johnson LA, Euston DR, Tatsuno M, McNaughton BL. 2010. Stored-trace reactivation in rat prefrontal cortex is correlated with down-to-up state fluctuation density. *Journal of Neuroscience* **30**:2650–2661. DOI: https://doi.org/10.1523/JNEUROSCI.1617-09.2010, PMID: 20164349

Joo HR, Frank LM. 2018. The hippocampal sharp wave-ripple in memory retrieval for immediate use and consolidation. *Nature Reviews Neuroscience* **19**:744–757. DOI: https://doi.org/10.1038/s41583-018-0077-1, PMID: 30356103

Kemker R, Kanan C. 2017. FearNet: brain-inspired model for incremental learning. *arXiv*. https://arxiv.org/abs/1711.10563.

Kirkpatrick J, Pascanu R, Rabinowitz N, Veness J, Desjardins G, Rusu AA, Milan K, Quan J, Ramalho T, Grabska-Barwinska A, Hassabis D, Clopath C, Kumaran D, Hadsell R. 2017. Overcoming catastrophic forgetting in neural networks. *PNAS* **114**:3521–3526. DOI: https://doi.org/10.1073/pnas.1611835114, PMID: 28292907

Kriegeskorte N. 2015. Deep neural networks: a new framework for modeling biological vision and brain information processing. *Annual Review of Vision Science* **1**:417–446. DOI: https://doi.org/10.1146/annurev-vision-082114-035447, PMID: 28532370

Krishnan GP, Chauvette S, Shamie I, Soltani S, Timofeev I, Cash SS, Halgren E, Bazhenov M. 2016. Cellular and neurochemical basis of sleep stages in the thalamocortical network. *eLife* **5**:e18607. DOI: https://doi.org/10.7554/eLife.18607, PMID: 27849520

Krishnan GP, Tadros T, Ramyaa R, Bazhenov M. 2019. Biologically inspired sleep algorithm for artificial neural networks. *arXiv*. https://arxiv.org/abs/1908.02240.

Ladenbauer J, Ladenbauer J, Külzow N, de Boor R, Avramova E, Grittner U, Flöel A. 2017. Promoting sleep oscillations and their functional coupling by transcranial stimulation enhances memory consolidation in mild cognitive impairment. *The Journal of Neuroscience* **37**:7111–7124. DOI: https://doi.org/10.1523/JNEUROSCI.0260-17.2017, PMID: 28637840

Langille JJ. 2019. Remembering to forget: a dual role for sleep oscillations in memory consolidation and forgetting. *Frontiers in Cellular Neuroscience* **13**:71. DOI: https://doi.org/10.3389/fncel.2019.00071, PMID: 30930746

Laventure S, Fogel S, Lungu O, Albouy G, Sévigny-Dupont P, Vien C, Sayour C, Carrier J, Benali H, Doyon J. 2016. NREM2 and sleep spindles are instrumental to the consolidation of motor sequence memories. *PLOS Biology* **14**:e1002429. DOI: https://doi.org/10.1371/journal.pbio.1002429, PMID: 27032084

Lehmann H, McNamara KC. 2011. Repeatedly reactivated memories become more resistant to hippocampal damage. *Learning & Memory* **18**:132–135. DOI: https://doi.org/10.1101/lm.2000811, PMID: 21325434

Louie K, Wilson MA. 2001. Temporally structured replay of awake hippocampal ensemble activity during rapid eye movement sleep. *Neuron* **29**:145–156. DOI: https://doi.org/10.1016/S0896-6273(01)00186-6, PMID: 11182087

Marshall L, Mölle M, Hallschmid M, Born J. 2004. Transcranial direct current stimulation during sleep improves declarative memory. *Journal of Neuroscience* **24**:9985–9992. DOI: https://doi.org/10.1523/JNEUROSCI.2725-04.2004, PMID: 15525784

Marshall L, Helgadóttir H, Mölle M, Born J. 2006. Boosting slow oscillations during sleep potentiates memory. *Nature* **444**:610–613. DOI: https://doi.org/10.1038/nature05278, PMID: 17086200

McClelland JL, McNaughton BL, O'Reilly RC. 1995. Why there are complementary learning systems in the Hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological Review* **102**:419–457. DOI: https://doi.org/10.1037/0033-295X.102.3.419, PMID: 7624455

McClelland JL, McNaughton BL, Lampinen AK. 2020. Integration of new information in memory: new insights from a complementary learning systems perspective. *Philosophical Transactions of the Royal Society B: Biological Sciences* **375**:20190637. DOI: https://doi.org/10.1098/rstb.2019.0637

Mccloskey M, Cohen NJ. 1989. Catastrophic interference in connectionist networks: the sequential learning problem. *The Psychology of Learning and Motivation* **24**:109–165. DOI: https://doi.org/10.1016/S0079-7421(08)60536-8

McCormick DA. 1992. Neurotransmitter actions in the thalamus and cerebral cortex and their role in neuromodulation of thalamocortical activity. *Progress in Neurobiology* **39**:337–388. DOI: https://doi.org/10.1016/0301-0082(92)90012-4, PMID: 1354387

McDevitt EA, Duggan KA, Mednick SC. 2015. REM sleep rescues learning from interference. *Neurobiology of Learning and Memory* **122**:51–62. DOI: https://doi.org/10.1016/j.nlm.2014.11.015, PMID: 25498222

Mednick SC, McDevitt EA, Walsh JK, Wamsley E, Paulus M, Kanady JC, Drummond SP. 2013. The critical role of sleep spindles in hippocampal-dependent memory: a pharmacology study. *Journal of Neuroscience* **33**:4494–4504. DOI: https://doi.org/10.1523/JNEUROSCI.3127-12.2013, PMID: 23467365

Mehta MR, Barnes CA, McNaughton BL. 1997. Experience-dependent, asymmetric expansion of hippocampal place fields. *PNAS* **94**:8918–8921. DOI: https://doi.org/10.1073/pnas.94.16.8918, PMID: 9238078

Morin A, Doyon J, Dostie V, Barakat M, Hadj Tahar A, Korman M, Benali H, Karni A, Ungerleider LG, Carrier J. 2008. Motor sequence learning increases sleep spindles and fast frequencies in post-training sleep. *Sleep* **31**:1149–1156. PMID: 18714787

Navratilova Z, Hoang LT, Schwindel CD, Tatsuno M, McNaughton BL. 2012. Experience-dependent firing rate remapping generates directional selectivity in hippocampal place cells. *Frontiers in Neural Circuits* **6**:6. DOI: https://doi.org/10.3389/fncir.2012.00006, PMID: 22363267

Ngo HV, Martinetz T, Born J, Mölle M. 2013. Auditory closed-loop stimulation of the sleep slow oscillation enhances memory. *Neuron* **78**:545–553. DOI: https://doi.org/10.1016/j.neuron.2013.03.006, PMID: 23583623

Nishida M, Walker MP. 2007. Daytime naps, motor memory consolidation and regionally specific sleep spindles. *PLOS ONE* **2**:e341. DOI: https://doi.org/10.1371/journal.pone.0000341, PMID: 17406665

Norimoto H, Makino K, Gao M, Shikano Y, Okamoto K, Ishikawa T, Sasaki T, Hioki H, Fujisawa S, Ikegaya Y. 2018. Hippocampal ripples down-regulate synapses. *Science* **359**:1524–1527. DOI: https://doi.org/10.1126/science.aao0702, PMID: 29439023

O'Neill J, Senior TJ, Allen K, Huxter JR, Csicsvari J. 2008. Reactivation of experience-dependent cell assembly patterns in the Hippocampus. *Nature Neuroscience* **11**:209–215. DOI: https://doi.org/10.1038/nn2037, PMID: 18193040

Oudiette D, Antony JW, Creery JD, Paller KA. 2013. The role of memory reactivation during wakefulness and sleep in determining which memories endure. *Journal of Neuroscience* **33**:6672–6678. DOI: https://doi.org/10.1523/JNEUROSCI.5497-12.2013, PMID: 23575863

Oudiette D, Paller KA. 2013. Upgrading the sleeping brain with targeted memory reactivation. *Trends in Cognitive Sciences* **17**:142–149. DOI: https://doi.org/10.1016/j.tics.2013.01.006, PMID: 23433937

Paller KA, Voss JL. 2004. Memory reactivation and consolidation during sleep. *Learning & Memory* **11**:664–670. DOI: https://doi.org/10.1101/lm.75704, PMID: 15576883

Papalambros NA, Santostasi G, Malkani RG, Braun R, Weintraub S, Paller KA, Zee PC. 2017. Acoustic enhancement of sleep slow oscillations and concomitant memory improvement in older adults. *Frontiers in Human Neuroscience* **11**:109. DOI: https://doi.org/10.3389/fnhum.2017.00109, PMID: 28337134

Ramanathan DS, Gulati T, Ganguly K. 2015. Sleep-Dependent reactivation of ensembles in motor cortex promotes skill consolidation. *PLOS Biology* **13**:e1002263. DOI: https://doi.org/10.1371/journal.pbio.1002263, PMID: 26382320

Rasch B, Born J. 2013. About sleep's role in memory. *Physiological Reviews* **93**:681–766. DOI: https://doi.org/10.1152/physrev.00032.2012, PMID: 23589831

Ratcliff R. 1990. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological Review* **97**:285–308. DOI: https://doi.org/10.1037/0033-295X.97.2.285, PMID: 2186426

Roth ED, Yu X, Rao G, Knierim JJ. 2012. Functional differences in the backward shifts of CA1 and CA3 place fields in novel and familiar environments. *PLOS ONE* **7**:e36035. DOI: https://doi.org/10.1371/journal.pone.0036035, PMID: 22558316

Roumis DK, Frank LM. 2015. Hippocampal sharp-wave ripples in waking and sleeping states. *Current Opinion in Neurobiology* **35**:6–12. DOI: https://doi.org/10.1016/j.conb.2015.05.001, PMID: 26011627

Rumelhart DE, Hinton GE, Williams RJ. 1986. Learning representations by back-propagating errors. *Nature* **323**:533–536. DOI: https://doi.org/10.1038/323533a0

Sanda P, Malerba P, Jiang X, Krishnan GP, Cash S, Halgren E, Bazhenov M. 2019. Interaction of hippocampal ripples and cortical slow waves leads to coordinated Large-Scale sleep rhythm. *bioRxiv*. DOI: https://doi.org/10.1101/568881

Shinoe T, Matsui M, Taketo MM, Manabe T. 2005. Modulation of synaptic plasticity by physiological activation of M1 muscarinic acetylcholine receptors in the mouse Hippocampus. *Journal of Neuroscience* **25**:11194–11200. DOI: https://doi.org/10.1523/JNEUROSCI.2338-05.2005, PMID: 16319319

Skaggs WE, McNaughton BL. 1996. Replay of neuronal firing sequences in rat Hippocampus during sleep following spatial experience. *Science* **271**:1870–1873. DOI: https://doi.org/10.1126/science.271.5257.1870, PMID: 8596957

Skelin I, Kilianski S, McNaughton BL. 2019. Hippocampal coupling with cortical and subcortical structures in the context of memory consolidation. *Neurobiology of Learning and Memory* **160**:21–31. DOI: https://doi.org/10.1016/j.nlm.2018.04.004, PMID: 29660400

Smith C, MacNeill C. 1994. Impaired motor memory for a pursuit rotor task following stage 2 sleep loss in college students. *Journal of Sleep Research* **3**:206–213. DOI: https://doi.org/10.1111/j.1365-2869.1994.tb00133.x, PMID: 10607127

Steriade M, Nunez A, Amzica F. 1993. Intracellular analysis of relations between the slow. *The Neurosci* **13**:3266–3283. DOI: https://doi.org/10.1523/JNEUROSCI.13-08-03266.1993

Steriade M, Timofeev I, Grenier F. 2001. Natural waking and sleep states: a view from inside neocortical neurons. *Journal of Neurophysiology* **85**:1969–1985. DOI: https://doi.org/10.1152/jn.2001.85.5.1969, PMID: 11353014

Stickgold R. 2013. Parsing the role of sleep in memory processing. *Current Opinion in Neurobiology* **23**:847–853. DOI: https://doi.org/10.1016/j.conb.2013.04.002, PMID: 23618558

Sugisaki E, Fukushima Y, Fujii S, Yamazaki Y, Aihara T. 2016. The effect of coactivation of muscarinic and nicotinic acetylcholine receptors on LTD in the hippocampal CA1 network. *Brain Research* **1649**:44–52. DOI: https://doi.org/10.1016/j.brainres.2016.08.024, PMID: 27545666

Swanson RA, Levenstein D, McClain K, Tingley D, Buzsáki G. 2020. Variable specificity of memory trace reactivation during hippocampal sharp wave ripples. *Current Opinion in Behavioral Sciences* **32**:126–135. DOI: https://doi.org/10.1016/j.cobeha.2020.02.008

Tadros T, Krishnan GP, Ramyaa R, Bazhenov M. 2020. Biologically inspired sleep algorithm for increased generalization and adversarial robustness in deep neural networks. ICLR 2020.

Tamaki M, Huang TR, Yotsumoto Y, Hämäläinen M, Lin FH, Náñez JE, Watanabe T, Sasaki Y. 2013. Enhanced spontaneous oscillations in the supplementary motor area are associated with sleep-dependent offline learning of finger-tapping motor-sequence task. *Journal of Neuroscience* **33**:13894–13902. DOI: https://doi.org/10.1523/JNEUROSCI.1198-13.2013, PMID: 23966709

Tononi G, Cirelli C. 2014. Sleep and the price of plasticity: from synaptic and cellular homeostasis to memory consolidation and integration. *Neuron* **81**:12–34. DOI: https://doi.org/10.1016/j.neuron.2013.12.025, PMID: 24411729

van de Ven GM, Tolias AS. 2018. Generative replay with feedback connections as a general strategy for continual learning. *arXiv*. https://arxiv.org/abs/1809.10635.

Vanini G, Lydic R, Baghdoyan HA. 2012. GABA-to-ACh ratio in basal forebrain and cerebral cortex varies significantly during sleep. *Sleep* **35**:1325–1334. DOI: https://doi.org/10.5665/sleep.2106, PMID: 23024430

Walker MP, Brakefield T, Morgan A, Hobson JA, Stickgold R. 2002. Practice with sleep makes perfect: sleep-dependent motor skill learning. *Neuron* **35**:205–211. DOI: https://doi.org/10.1016/s0896-6273(02)00746-8, PMID: 12123620

Walker MP, Stickgold R. 2004. Sleep-dependent learning and memory consolidation. *Neuron* **44**:121–133. DOI: https://doi.org/10.1016/j.neuron.2004.08.031, PMID: 15450165

Wei Y, Krishnan GP, Bazhenov M. 2016. Synaptic mechanisms of memory consolidation during sleep slow oscillations. *The Journal of Neuroscience* **36**:4231–4247. DOI: https://doi.org/10.1523/JNEUROSCI.3648-15.2016, PMID: 27076422

Wei Y, Krishnan GP, Komarov M, Bazhenov M. 2018. Differential roles of sleep spindles and sleep slow oscillations in memory consolidation. *PLOS Computational Biology* **14**:e1006322. DOI: https://doi.org/10.1371/journal.pcbi.1006322, PMID: 29985966

Wei Y, Krishnan GP, Marshall L, Martinetz T, Bazhenov M. 2020. Stimulation augments spike sequence replay and memory consolidation during Slow-Wave sleep. *The Journal of Neuroscience* **40**:811–824. DOI: https://doi.org/10.1523/JNEUROSCI.1427-19.2019, PMID: 31792151

Weigenand A, Mölle M, Werner F, Martinetz T, Marshall L. 2016. Timing matters: open-loop stimulation does not improve overnight consolidation of word pairs in humans. *European Journal of Neuroscience* **44**:2357–2368. DOI: https://doi.org/10.1111/ejn.13334, PMID: 27422437

Werbos PJ. 1990. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE* **78**:1550–1560. DOI: https://doi.org/10.1109/5.58337

Wilson MA, McNaughton BL. 1994. Reactivation of hippocampal ensemble memories during sleep. *Science* **265**:676–679. DOI: https://doi.org/10.1126/science.8036517, PMID: 8036517

Xu W, de Carvalho F, Jackson A. 2019. Sequential neural activity in primary motor cortex during sleep. *The Journal of Neuroscience* **39**:3698–3712. DOI: https://doi.org/10.1523/JNEUROSCI.1408-18.2019, PMID: 30842250

Zz L, Hoiem D. 2018. Learning without forgetting. *Ieee T Pattern Anal* **40**:2935–2947. DOI: https://doi.org/10.1109/TPAMI.2017.2773081

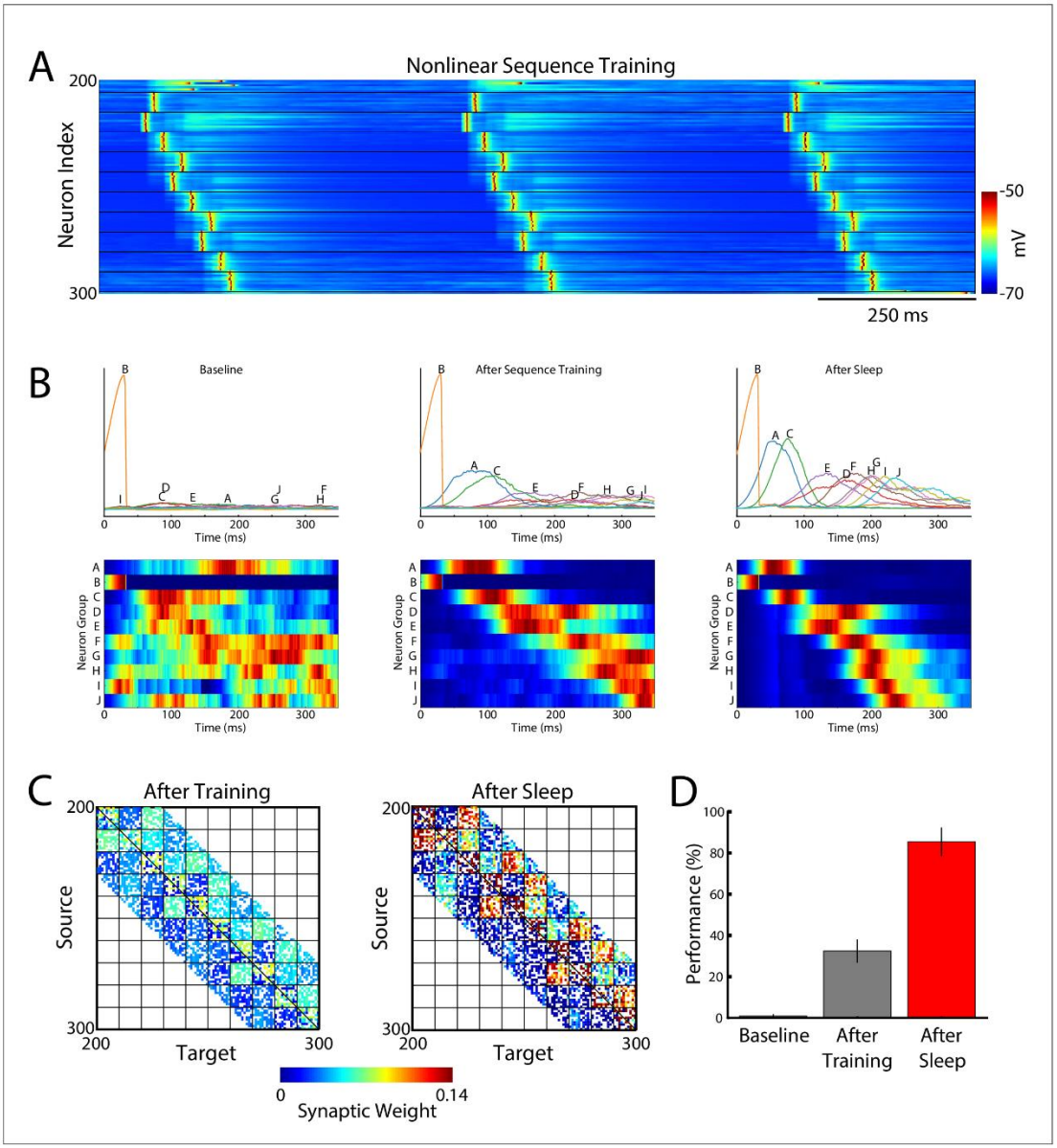


Figure 2—figure supplement 1. Sleep replay improves performance for complex non-linear sequences. (**A**) Example of the training protocol used for training a long non-linear sequence - BACEDFHGIJ. (**B**) Average group activations during baseline testing (left), after sequence training (middle), and after sleep (right). Top panels show average group firing rates during testing periods. Letters above each line indicate the group in the sequence. Sleep results in increase of the firing rates (higher peaks) and sharpening of the response times (narrower distribution) as compared to "before sleep". Bottom panels show normalized average group responses during testing periods. Sleep leads to an improvement and tuning of the responses such

Figure 2—figure supplement 1 continued on next page

Figure 2—figure supplement 1 continued

that testing after sleep results in the correct ordering of group activations and faster completion of the sequence. (**C**) Synaptic weight matrices in the trained region of the network before (left) and after (right) sleep. Color indicates synaptic strength. (**D**) Performance of the sequence completion at baseline, after training, and after sleep (red).

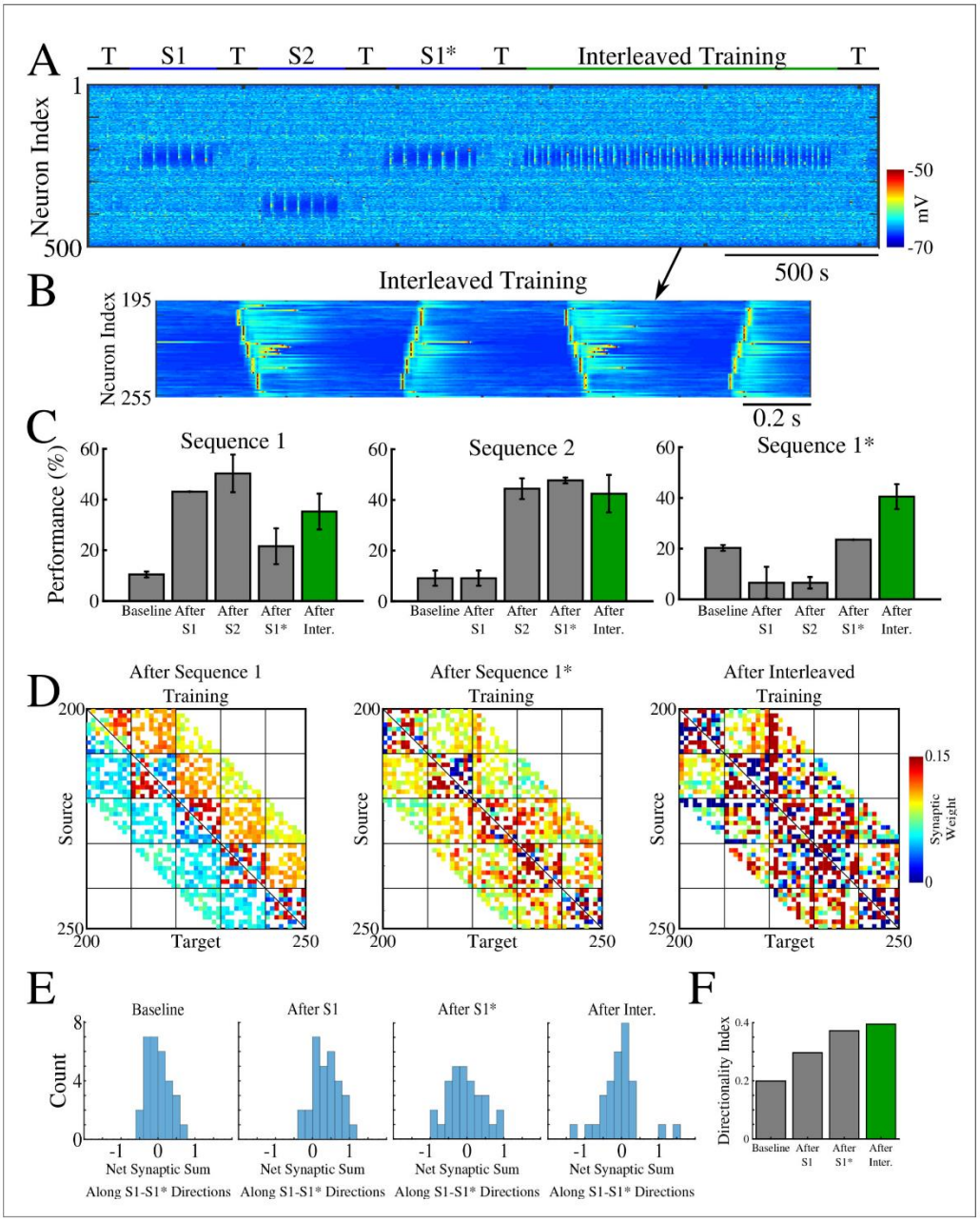


Figure 4—figure supplement 1. Interleaved training of the old and new memory sequences prevents the old sequence from forgetting and improves performance for both memories. (**A**) Network activity during sequential training of memory sequences S1→ S2→ S1* (blue bars) followed by interleaved

Figure 4—figure supplement 1 continued on next page

Figure 4—figure supplement 1 continued

training of S1/S1* (green bar). (**B**) Example of stimulation protocol used for interleaved training of S1/S1*. (**C**) Testing of S1, S2, and S1* shows increase in performance of S1 and S1* after interleaved training (green). (**D**) Weighted adjacency matrices showing changes after initial sequential training of S1 and S1*, and after interleaved S1/S1* training. (**E**) Distributions of the net sum of synaptic weights each neuron receives from all the neurons belonging to its left vs right neighboring groups within a trained region at baseline (left), after training S1 (middle/left), after training S1* (middle/right), and after interleaved training (right). (**F**) Synaptic weight-based directionality index before/after training both sequences (gray bars) and after interleaved training (green bar).

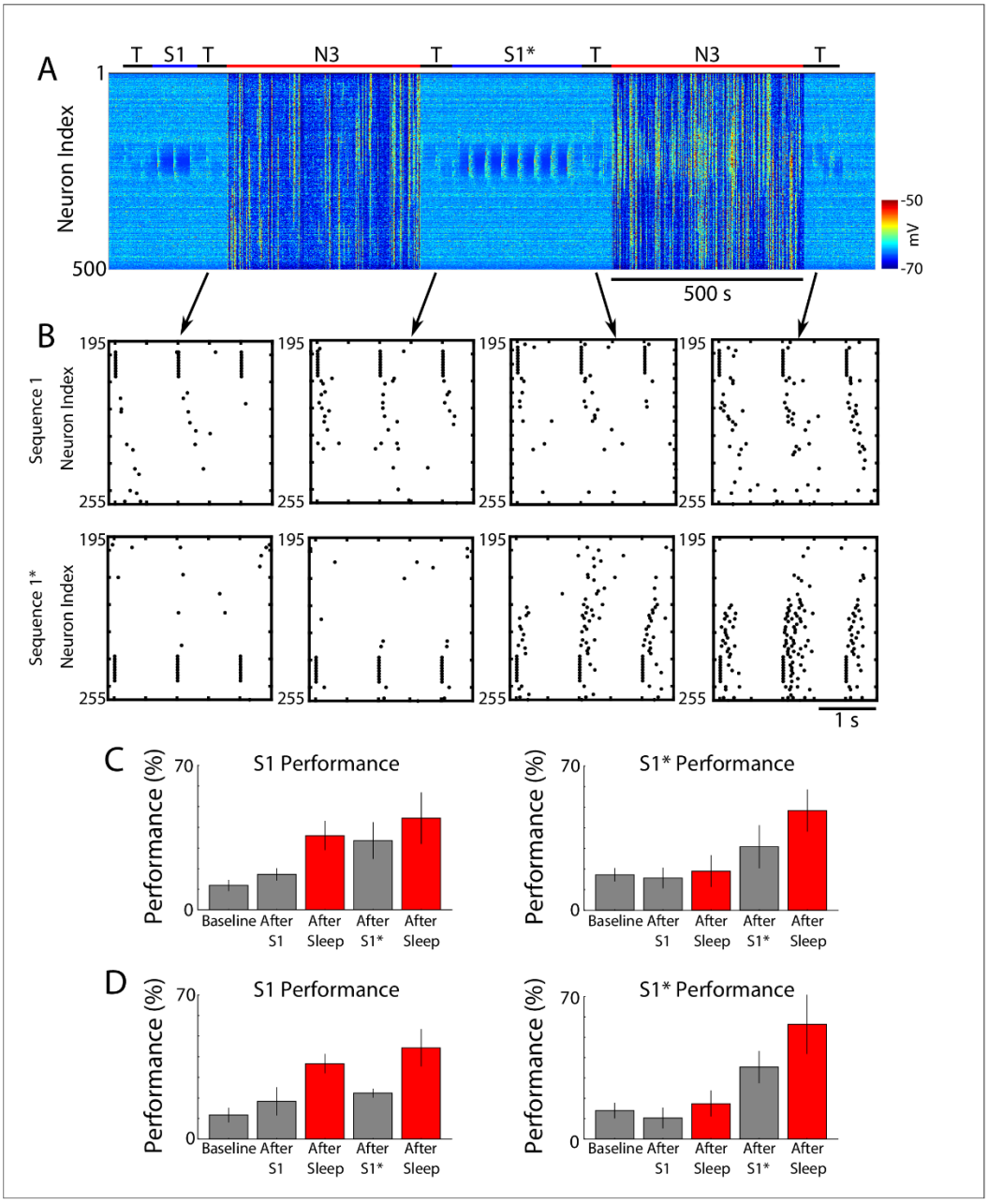


Figure 5—figure supplement 1. Training of a new memory that interferes with previously consolidated old memory leads to forgetting that can be reversed by subsequent sleep. (**A**) Network activity (PY neurons) during training of S1 (150 s), S1* (350 s) (blue bars) and N3 sleep (red bars). No

Figure 5—figure supplement 1 continued on next page

Figure 5—figure supplement 1 continued

stimulation was applied during sleep. (**B**) Examples of testing periods for each trained memory at different times. The top row corresponds to the testing of sequence 1 (S1) and bottom is testing of sequence 1* (S1*). Arrows indicate time of specific testing period. (**C**) Performance of S1 (S1*) at baseline, after each training period and after each sleep period (red) for the network shown in A and B. (**D**) Performance of S1 (S1*) at baseline, after each training period and after each sleep period (red) for a network with longer (450 s) S1* training.

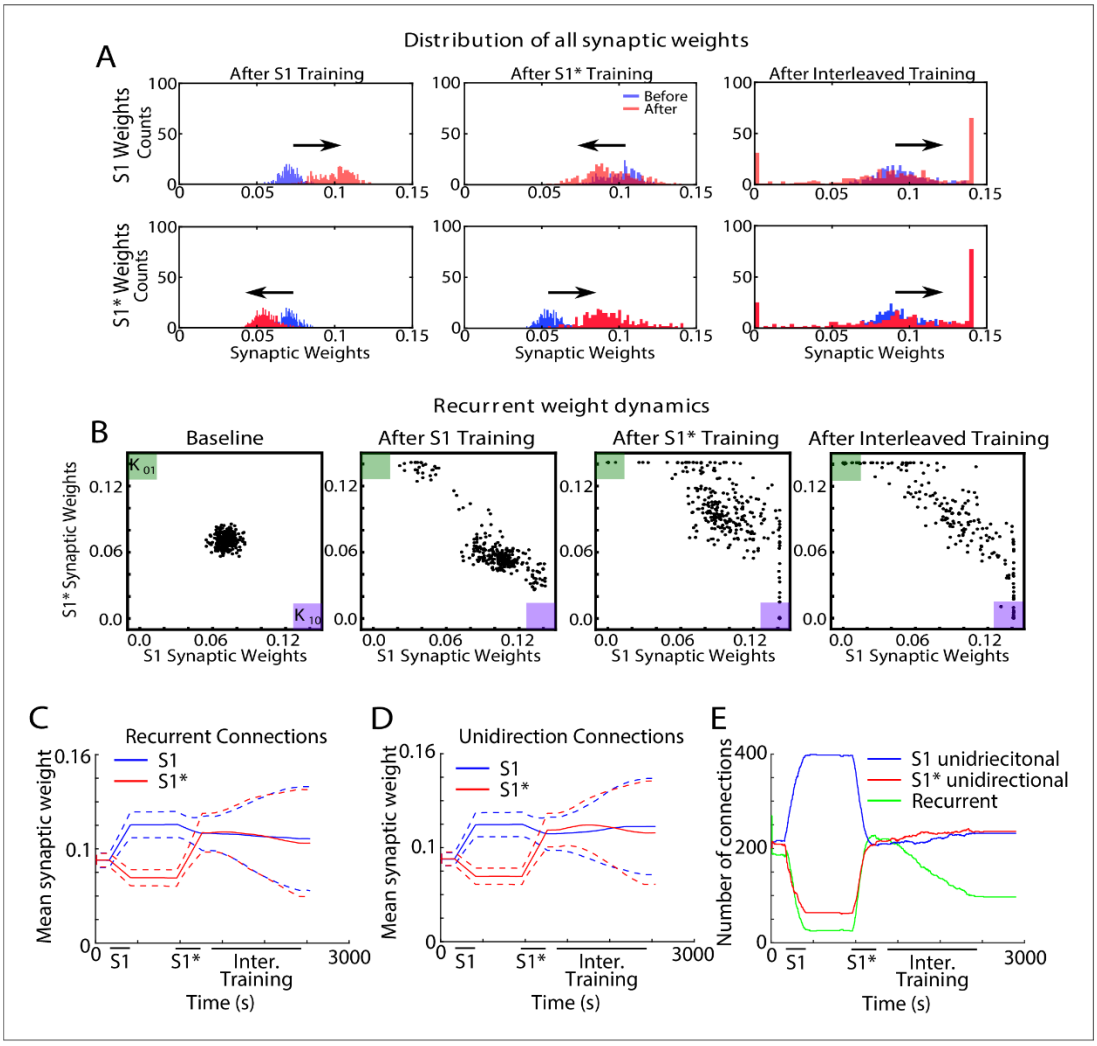


Figure 7—figure supplement 1. Interleaved training revealed synaptic weight dynamics that are similar to sleep but result in less segregation of synaptic weights. (**A**) Dynamics of synaptic weight distributions from the trained region. Top row shows strength of synapses in direction of S1. Bottom row shows strength of synapses in direction of S1*. Blue shows the starting points for weights, and red shows new weights after different specific events, for example, S1 training, S1* training, interleaved training. (**B**) Scatter plots show synaptic weights for all pairs of neurons contributing to both S1 and S1* before and after training (left/middle) and after interleaved training (right). For each pair of neurons (e.g., n1-n2), the X-coordinate shows the strength of $W_{n1\rightarrow n2}$ synapse and the Y-coordinate shows the strength of $W_{n2\rightarrow n1}$ synapse. The green (K_{01}) and purple (K_{10}) boxes show the locations in the scatter plot representing synaptic pairs with strong preference for S1* (green) or S1 (purple). (**C**) The evolution of the mean synaptic strength (solid line) and the standard deviation (dashed line) of recurrent connections in S1 (blue) and S1* (red) direction. (**D**) The evolution of the mean synaptic weight (solid line) and the standard deviation (dashed line) of unidirectional connections in S1 (blue) and S1* (red) direction. Note the lack of overall increase in synaptic strength after interleaved training as compared to sleep (***Figure 7C***). (**E**) The number of functionally recurrent and unidirectional connections in the trained region of the network as a function of time obtained after thresholding connectivity matrix with threshold 0.065 (which is smaller than the initial mean synaptic strength).

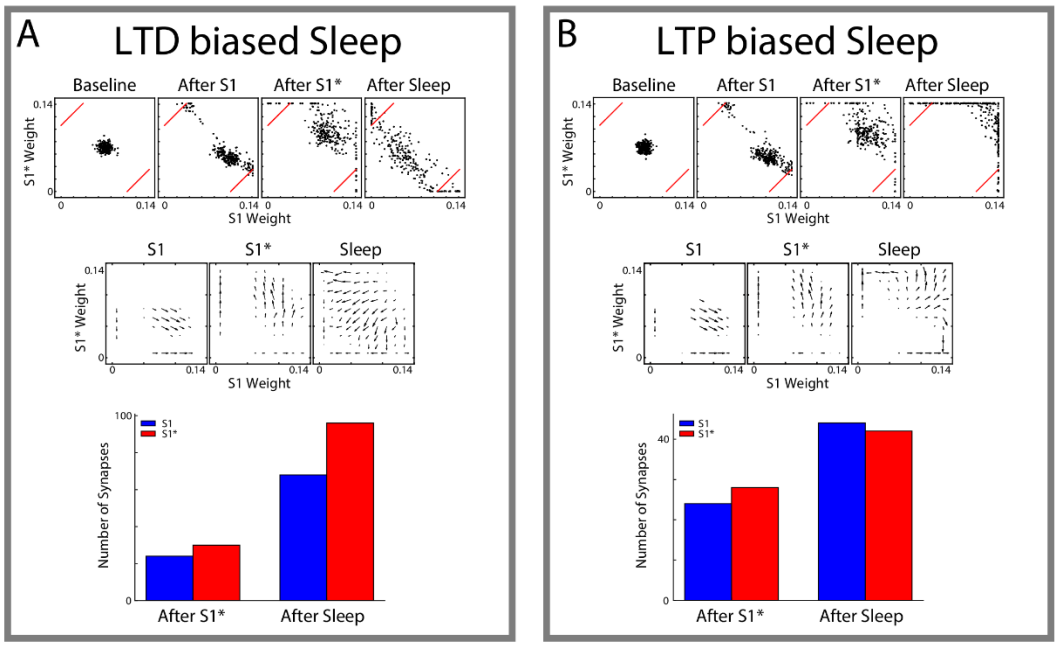


Figure 7—figure supplement 2. Synaptic plasticity that is biased towards LTP or LTD also results in memory orthogonalization during sleep . Synaptic dynamics for LTP/LTD ratio biased towards LTD (A_+/A_-=0.0019/0.002) (**A**) or LTP (A_+/A_-=0.0021/0.002) (**B**). Top, Scatter plots showing synaptic weights for all reciprocally connected pairs of neurons before and after training (left/middle) and after sleep (right). For each pair of neurons (e.g., n1-n2), the X-coordinate shows the strength of $W_{n1\rightarrow n2}$ synapse and the Y-coordinate shows the strength of $W_{n2\rightarrow n1}$ synapse. The red lines indicate the thresholds used to determine synapses which are preferentially strong for S1 (bottom right) or S1* (top left). Middle, Vector fields summarizing the average synaptic weights dynamics of the scatter plots in the top panel. Arrows point in the direction of the average movement of synaptic weights and length of the arrows indicates the amplitude of the movement. Bottom, Total number of synapses which are preferentially strong for S1 (blue) or S1* (red) after training S1*/before sleep (left) and after sleep (right). Thresholds for determining preference for either sequence are indicated by red lines in the scatter plots.

Chapter 1, in full, is a reprint of the material as it appears Elife.

González, O. C., Sokolov, Y., Krishnan, G. P., Delanois, J. E., & Bazhenov, M. (2020). Can sleep protect memories from catastrophic forgetting?. Elife, 9, e51005.

PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Sleep prevents catastrophic forgetting in spiking neural networks by forming a joint synaptic weight representation

Ryan Golden[1,2☯], **Jean Erik Delanois**[2,3☯], **Pavel Sanda**[4], **Maxim Bazhenov**[1,2*]

1 Neurosciences Graduate Program, University of California, San Diego, La Jolla, California, United States of America, **2** Department of Medicine, University of California, San Diego, La Jolla, California, United States of America, **3** Department of Computer Science and Engineering, University of California, San Diego, La Jolla, California, United States of America, **4** Institute of Computer Science of the Czech Academy of Sciences, Prague, Czech Republic

☯ These authors contributed equally to this work.
* mbazhenov@ucsd.edu

Abstract

Artificial neural networks overwrite previously learned tasks when trained sequentially, a phenomenon known as catastrophic forgetting. In contrast, the brain learns continuously, and typically learns best when new training is interleaved with periods of sleep for memory consolidation. Here we used spiking network to study mechanisms behind catastrophic forgetting and the role of sleep in preventing it. The network could be trained to learn a complex foraging task but exhibited catastrophic forgetting when trained sequentially on different tasks. In synaptic weight space, new task training moved the synaptic weight configuration away from the manifold representing old task leading to forgetting. Interleaving new task training with periods of off-line reactivation, mimicking biological sleep, mitigated catastrophic forgetting by constraining the network synaptic weight state to the previously learned manifold, while allowing the weight configuration to converge towards the intersection of the manifolds representing old and new tasks. The study reveals a possible strategy of synaptic weights dynamics the brain applies during sleep to prevent forgetting and optimize learning.

Author summary

Artificial neural networks can achieve superhuman performance in many domains. Despite these advances, these networks fail in sequential learning; they achieve optimal performance on newer tasks at the expense of performance on previously learned tasks. Humans and animals on the other hand have a remarkable ability to learn continuously and incorporate new data into their corpus of existing knowledge. Sleep has been hypothesized to play an important role in memory and learning by enabling spontaneous reactivation of previously learned memory patterns. Here we use a spiking neural network model, simulating sensory processing and reinforcement learning in animal brain, to demonstrate that interleaving new task training with sleep-like activity optimizes the

Citation: Golden R, Delanois JE, Sanda P, Bazhenov M (2022) Sleep prevents catastrophic forgetting in spiking neural networks by forming a joint synaptic weight representation. PLoS Comput Biol 18(11): e1010628. https://doi.org/10.1371/journal.pcbi.1010628

Editor: Daniel Bush, University College London, UNITED KINGDOM

Received: April 22, 2022

Accepted: October 3, 2022

Published: November 18, 2022

Peer Review History: PLOS recognizes the benefits of transparency in the peer review process; therefore, we enable the publication of all of the content of peer review and author responses alongside final, published articles. The editorial history of this article is available here: https://doi.org/10.1371/journal.pcbi.1010628

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: This study was supported by ONR (N00014-16-1-2829 to MB), Lifelong Learning Machines program from DARPA/MTO (HR0011-18-2-0021 to MB), NSF (EFRI BRAID 2223839 to MB), and NIH (1RF1MH117155 to MB; 1R01MH125557 to MB; 1R01NS109553 to MB). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

network's memory representation in synaptic weight space to prevent forgetting old memories. Sleep makes this possible by replaying old memory traces without the explicit usage of the old task data.

Introduction

Humans are capable of continuously learning to perform novel tasks throughout life without interfering with their ability to perform previous tasks. Conversely, while modern artificial neural networks (ANNs) are capable of learning to perform complicated tasks, ANNs have difficulty learning multiple tasks sequentially [1–3]. Sequential training commonly results in catastrophic forgetting, a phenomenon which occurs when training on the new task completely overwrites the synaptic weights learned during the previous task, leaving the ANN incapable of performing a previous task [1–4]. Attempts to solve catastrophic forgetting have drawn on insights from the study of neurobiological learning, leading to the growth of neuroscience-inspired artificial intelligence (AI) [5–8]. While proposed approaches are capable of mitigating catastrophic forgetting in certain circumstances, a general solution which can achieve human level performance for continual learning is still an open question [9].

Historically, an interleaved training paradigm, where multiple tasks are presented within a common training dataset, has been employed to circumvent the issue of catastrophic forgetting [4,10,11]. In fact, interleaved training was originally construed to be an approximation to what the brain may be doing during sleep to consolidate memories; spontaneously reactivating memories from multiple interfering tasks in an interleaved manner [11]. Unfortunately, explicit use of interleaved training, in contrast to memory consolidation during biological sleep, imposes the stringent constraint that the original training data be perpetually stored for later use and combined with new data to retrain the network [1,2,4,11]. Thus, the challenge is to understand how the biological brain enables memory reactivation during sleep without access to past training data.

Parallel to the growth of neuroscience-inspired ANNs, there has been increasing investigation of spiking neural networks (SNNs) which attempt to provide a more realistic model of brain functioning by taking into account the underlying neural dynamics and by using biologically plausible local learning rules [12–15]. A potential advantage of the SNNs, that was explored in our new study, is that local learning rules combined with spike-based communication allow previously learned memory traces to reactivate spontaneously and modify synaptic weights without interference during off-line processing–sleep. Indeed, a common hypothesis, supported by a vast range of neuroscience data, is that the consolidation of memories during sleep occurs through synaptic changes enabled by reactivation of the neuron ensembles engaged during learning [16–20]. It has been suggested that Rapid Eye Movement (REM) sleep supports the consolidation of non-declarative or procedural memories, while non-REM sleep supports the consolidation of declarative memories [16,21–23].

Here we used a multi-layer SNN with reinforcement learning to investigate whether interleaving periods of new task training with periods of sleep-like autonomous activity, can circumvent catastrophic forgetting. The network can be trained to learn one of two complementary complex foraging tasks involving pattern discrimination but exhibits catastrophic forgetting when trained on the tasks sequentially. Significantly, we show that catastrophic forgetting can be prevented by periodically interrupting reinforcement learning on a new task with sleep-like phases. From the perspective of synaptic weight space, while new task training alone moves the synaptic weight configuration away from the old task's manifold–a subspace of synaptic weight space that guarantees high performance on that task—and towards

the new task manifold, interleaving new task training with sleep replay allows the synaptic weights to stay near the old task manifold and still move towards its intersection with the manifold representing the new task, i.e., converge to the intersection of these manifolds. Our study predicts that sleep prevents catastrophic forgetting in the brain by forming joint synaptic weight representations suitable for storing multiple memories.

Results

Human and animal brains are complex and although there are many differences between species, critical common elements can still be identified from insects to humans. From an anatomic perspective, this includes largely the sequential processing of sensory information, from raw low level representations on the sensory periphery to high level representations deeper in the brain followed by decision making networks controlling the motor circuits. From a functional perspective, this includes local synaptic plasticity, combination of different plasticity rules and sleep-wake cycle that was shown to be critical for memory and learning in variety of species from insects [24–26] to vertebrates [16]. In this new study we model a basic brain neural circuit including many of these anatomical and functional elements. While our model is extremely simplified, it captures critical processing steps found, e.g., in insect olfactory system where odor information is sent from olfactory receptors to the mushroom bodies and then to the motor circuits. In vertebrates, visual information is sent from the retina to early visual cortex and then to decision making layers in associative cortices to drive motor output. Many of these steps are plastic, in particular decision making circuits utilize spike timing dependent plasticity (STDP) in insects [27] and vertebrates [28,29].

Fig 1A illustrates a feedforward spiking neural network (see also *Methods*: *Network Structure* for details) simulating the basic steps from sensory input to motor output. Excitatory synapses between the input (I) and hidden (H) layers were subjected to unsupervised learning (implemented as non-rewarded STDP) [28,29] while those between the H and output (O) layers were subjected to reinforcement learning (implemented using rewarded STDP) [30–33] (see *Methods*: *Synaptic plasticity* for details). Unsupervised plasticity allowed neurons in layer H to learn different particle patterns at various spatial locations of the input layer I, while rewarded STDP allowed the neurons in layer O to learn motor decisions based on the type of the particle patterns detected in the input layer [14]. While inspired by the processing steps of a biological brain, this structure also mimics basic elements of the feedforward artificial neural networks (ANNs), including convolutional layer (from I to H) and fully connected layer (from H to O) [34].

Complementary complex foraging tasks can be robustly learned

We trained the network on one of two complementary complex foraging tasks. In either task, the network learned to discriminate between rewarded and punished particle patterns in order to acquire as much reward as possible. We consider pattern discriminability (ratio of rewarded vs punished particles consumed) as a measure of performance, with chance performance being 0.5. All reported results are based on at least 10 trials with different random network initialization.

The paradigm for Task 1 is shown in Fig 1B. First, during an unsupervised learning period, all 4 types of 2-particle patterns (horizontal, vertical, positive diagonal, and negative diagonal) were present in the environment with equal densities. This was a period, equivalent to a developmental critical period in the brain (or training convolutional layers in ANN), when the network learned the environmental statistics and formed, in layer H, high level representations of all possible patterns found at the different visual field locations (see Fig 2 for details).

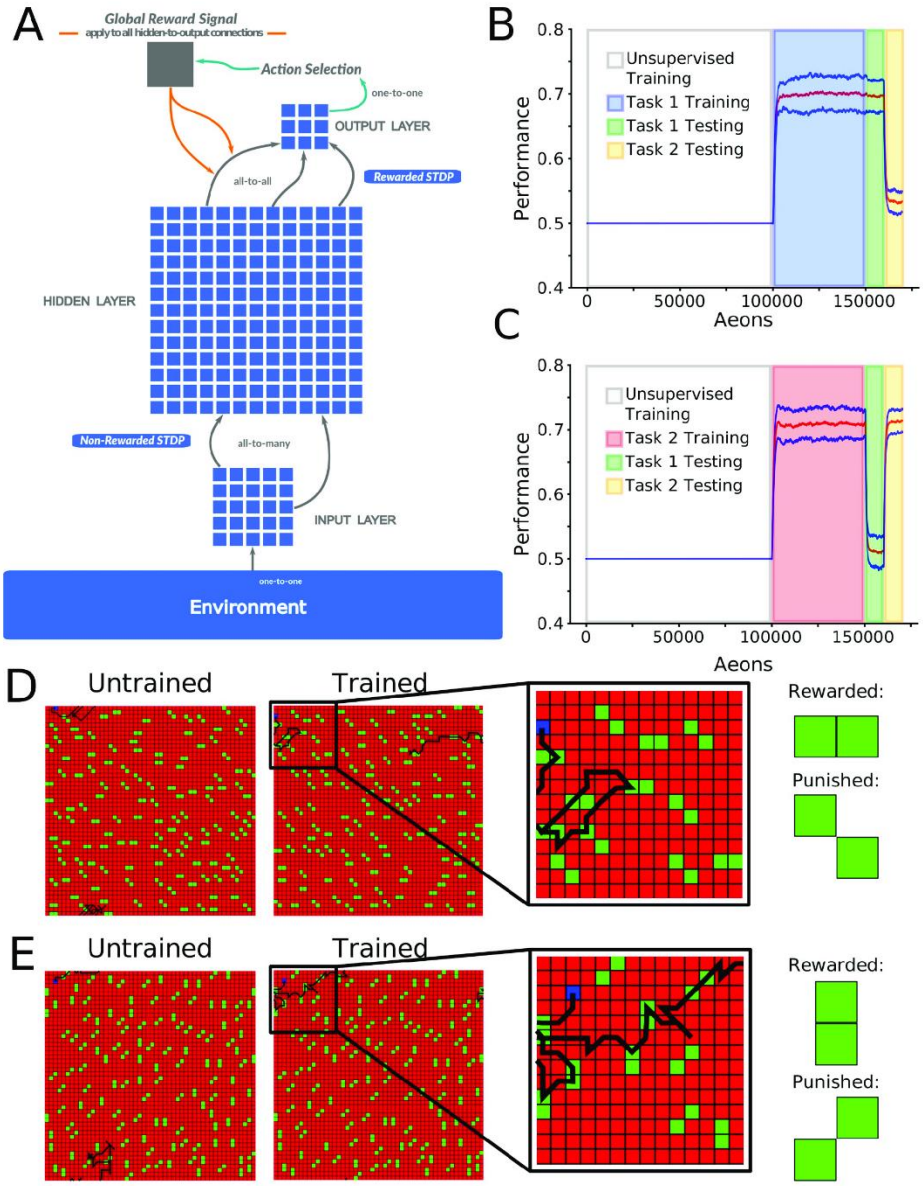


Fig 1. Network architecture and foraging task structure. (A) The network had three layers of neurons with a feed-forward connectivity scheme. Input from virtual environment was simulated as a set of excitatory inputs to the input layer neurons ("visual field"- 7x7 subspace of 50x50 environment) representing the position of food particles in an egocentric reference frame relative to the virtual agent. Each hidden layer neuron received an excitatory synapse from 9 randomly selected input layer neurons. Each output layer neuron received one excitatory and one inhibitory synapse from each hidden layer neuron. The most active neuron in the output layer (size 3x3) determined the direction of

movement. **(B)** Mean performance (redline) and standard deviation (blue lines) over time: unsupervised training (white), Task 1 training (blue), and Task 1 (green) and Task 2 (yellow) testing. The y-axis represents the agent's performance, or the probability of acquiring rewarded as opposed to punished particle patterns. The x-axis is time in aeons (1 aeon = 100 movement cycles). **(C)** The same as shown in (B) except now for: unsupervised training (white), Task 2 training (red), and Task 1 (green) and Task 2 (yellow) testing. **(D)** Examples of trajectories through the environment at the beginning (left) and at the end (middle-left) of training on Task 1, with a zoom in on the trajectory at the end of training (middle-right), and the values of the task-relevant food particles (right). **(E)**. The same as shown in (D) except for Task 2.

https://doi.org/10.1371/journal.pcbi.1010628.g001

Unsupervised training was followed by a reinforcement learning period, equivalent to task specific training in the brain (or training a specific set of classes in an ANN), during which the synapses between layers I and H were frozen while synapses from H to O were updated using a

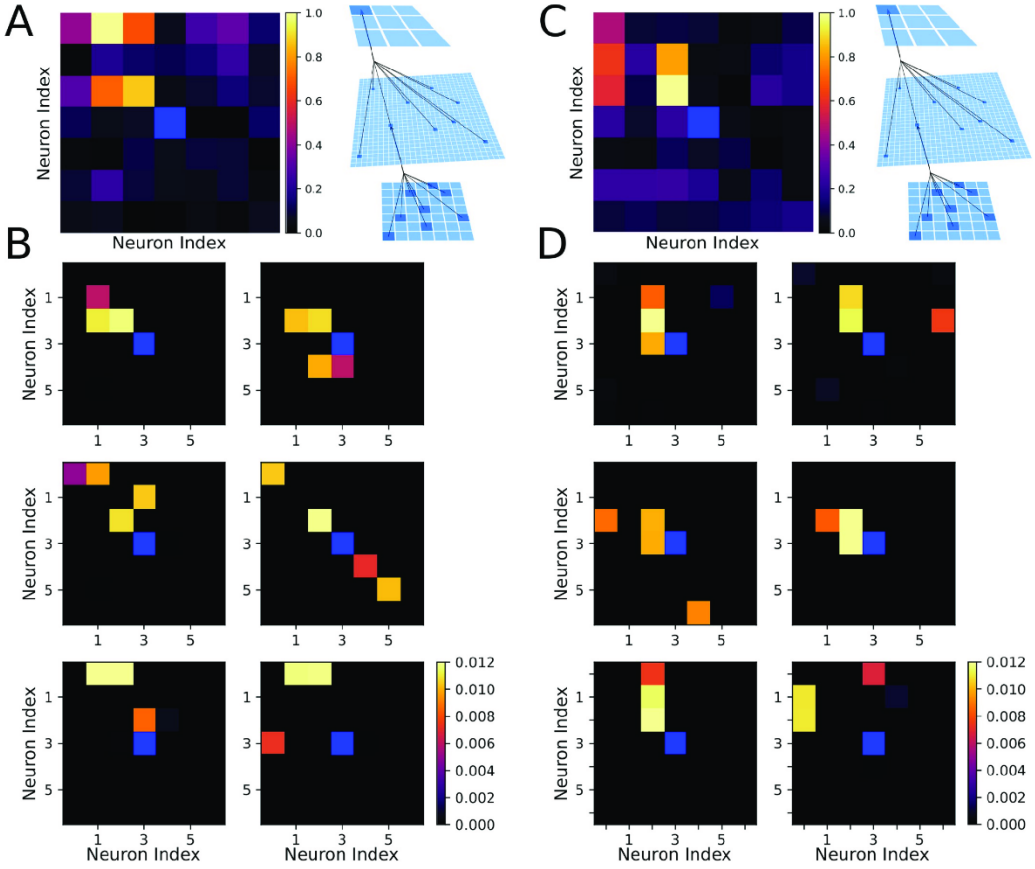


Fig 2. Receptive fields of output and hidden layer neurons determine the agent behavior. (A) Left, Receptive field of the output layer neuron controlling movement to the upper-left direction following training on Task 1. This neuron can be seen to selectively respond to horizontal orientations in the upper-left quadrant of the visual field. Right, Schematic of connections between layers. **(B)** Examples of receptive fields of hidden layer neurons which synapse strongly onto the output neuron from (A) after training on Task 1. **(C)** The same as shown in (A) except following training on Task 2. The upper-left decision neuron can be seen to selectively respond to vertical orientations in the upper-left quadrant of the visual field. **(D)** The same as shown in (B) except following training on Task 2.

https://doi.org/10.1371/journal.pcbi.1010628.g002

Sleep prevents catastrophic forgetting of the old task during new task training

We next tested whether the model exhibits catastrophic forgetting by training sequentially on Task 1 (old task) followed by Task 2 (new task) (Fig 3A). Following Task 2 training, mean performance across ten trials on Task 1 was down to no better than chance (0.52 ± 0.02), while performance on Task 2 improved to 0.69 ± 0.03 (Fig 3A and 3B). Thus, sequential training on a complementary task caused the network to undergo catastrophic forgetting of the task trained earlier, remembering only the most recent task.

Interleaved training was proposed as a solution for catastrophic forgetting [4,10,11]. In the next experiment, after training on Task 1, we simulated interleaved T1/T2 training (Interleaved$_{T1,T2}$) when we alternated short presentations of Task 1 and Task 2 every 100 movement cycles (Fig 3C). Sample network activity from this period can be seen to closely resemble single task training (S1C Fig). Following interleaved training, the network achieved a mean performance of 0.68 ±0.03 on Task 1 and a performance of 0.65 ± 0.04 on Task 2 across trials. Therefore, interleaved training allowed the network to learn new Task 2 without forgetting previously learned Task 1. However, while interleaved training made it possible to learn both tasks, it imposes the stringent constraint that all the original training data (in our case explicit access to the Task 1 environment) be stored for later use and combined with new data to retrain the network [1,2,4,11].

Sleep is believed to be an off-line processing period when recent memories are replayed to avoid damage from new learning. We previously showed that sleep replay improves memory in a thalamocortical network [38–40] and when a network was trained to learn interfering tasks sequentially, sleep prevented the old task memory from catastrophic forgetting [41]. Can we implement a sleep like phase to our model to protect an old task and still accomplish new task learning without explicit re-training of the old task? In vivo, activity of the neocortical neurons during REM sleep is low-synchronized and similar to baseline awake activity [42]. Therefore, to simulate REM sleep-like activity in the model, the rewarded STDP rule was replaced by unsupervised STDP, the input layer was silenced while hidden layer neurons were artificially stimulated by Poisson distributed spike trains in order to maintain spiking rates similar to that during task training (see *Methods*: *Simulated Sleep* for details). Sample network activity recorded during this sleep phase is visualized in the raster plots shown in S1D Fig.

Again, we first trained the network on Task 1. Next, we implemented a training phase consisted of alternating periods of training on Task 2 (new task) lasting 100 movement cycles and periods of "sleep" of the same duration (we will refer to this training phase as Interleaved$_{S,T2}$) (Fig 3E). Importantly, no training on Task 1 was performed at any time during Interleaved$_{S,T2}$. Following Interleaved$_{S,T2}$, the network achieved a mean performance across ten trials of 0.68 ± 0.05 on Task 2 and retained a performance of 0.70 ± 0.03 on Task 1 (Fig 3E and 3F), comparable to single Task 1 (0.70 ± 002) or Task 2 (0.69 ± 0.03) performances (Fig 1B and 1C) and exceeding those achieved through Interleaved$_{T1,T2}$ training (Fig 3C and 3D).

We interpret these results as follows (see below for detailed synaptic connectivity analysis). Each episode of new Task 2 training improves Task 2 performance but damages synaptic connectivity responsible for old Task 1. If continuous Task 2 training is long enough, the damage to Task 1 becomes irreversible. Having a sleep phase after a short period of Task 2 training enables spontaneous forward replay between hidden and output layers (H->O) that preferentially benefits the strongest synapses. Thus, if Task 1 synapses are still strong enough to maintain replay, they are replayed and weights are increased.

rewarded STDP rule. The reinforcement learning period was when the network learned to make decisions about which direction to move based on the visual input. For Task 1, horizontal patterns were rewarded and negative diagonal patterns were punished (Fig 1D). During both the rewarded training and the testing periods only 2 types of patterns were present in the environment (e.g. horizontal and negative diagonal for Task 1).

After training Task 1, mean performance across ten trials on Task 1 was 0.70 ± 0.02 while performance on the untrained Task 2 was 0.53 ± 0.02 (chance level). The naive agent moved randomly through the environment (Fig 1D, left), but after task training, moved to seek out horizontal patterns and largely avoid negative diagonal ones (Fig 1D, right). The complementary paradigm for Task 2 (vertical patterns are rewarded, and positive diagonal are punished) is shown in Fig 1C and 1E. These results demonstrate that the network is capable of learning and performing either one of the two complementary complex foraging tasks. The similarity between these tasks is evident in their definition (symmetrical particle orientations; Fig 1D and 1E), through the similar performances attained by the network on each task (Fig 1B and 1C), and through the similar levels of activity induced in the network when training each task (S1A and S1B Fig)

To understand how sensitive a trained network was to pruning, we employed a neuronal dropout procedure which progressively removes neurons from the hidden layer at random (S2 Fig). We found the network was able to keep performance steady on either task following training until around 70% of the hidden layer was pruned. Such high resiliency suggests the network utilizes a highly distributed coding strategy to develop its policy.

Next, to understand synaptic changes during training, we computed receptive fields of each neuron in layer O with respect to the inputs from layer I (see schematic in Fig 2A and 2C). This was done by first computing the receptive fields of all of the neurons in layer H with respect to I, then performing a weighted average where the weights were given by the synaptic strength from each neuron in layer H to the particular neuron in layer O. Fig 2A shows a representative example of the receptive field which developed after training on Task 1 for one specific neuron in layer O which controls movements to the upper-left direction. This neuron responded most robustly to bars of horizontal orientation (rewarded) in the upper-left quadrant of the visual field and, importantly, did not respond to bars of negative diagonal orientation (punished).

Fig 2B shows examples of receptive fields of six neurons in layer H which synapse strongly onto the upper-left neuron in layer O (the neuron shown in Fig 2A). These neurons formed high level representations of the input patterns, similar to the neurons in the primary visual system or later layers of a convolutional neural network [35–37]. The majority of these receptive fields revealed strong selection for the horizontal (i.e. rewarded) food particles in the upper-left quadrant of the visual field. As a particularly notable example, one of these layer H neurons (Fig 2B; middle-right) preferentially responded to negative diagonal (i.e. punished) food particles in the bottom-right quadrant of the visual field. Thus, spiking in this neuron caused the agent to move away from these punished food particles. Similar findings after training on Task 2 are shown in Fig 2C and 2D.

To further quantify the network's sensitivity to various particle types we developed a metric termed the Particle Responsiveness Metric (PRM) to gauge how specific particles influence activity of the output layer neurons (see the section Methods: Particle responsiveness metric for further details). Using PRM on all food particle orientations across ten trials, we found that following Task 1 training the network is drawn to horizontal particles (S3A Fig) while post Task 2 training vertical particles drive output layer activity (S3B Fig), thus quantitatively supporting the qualitative results displayed in Fig 2.

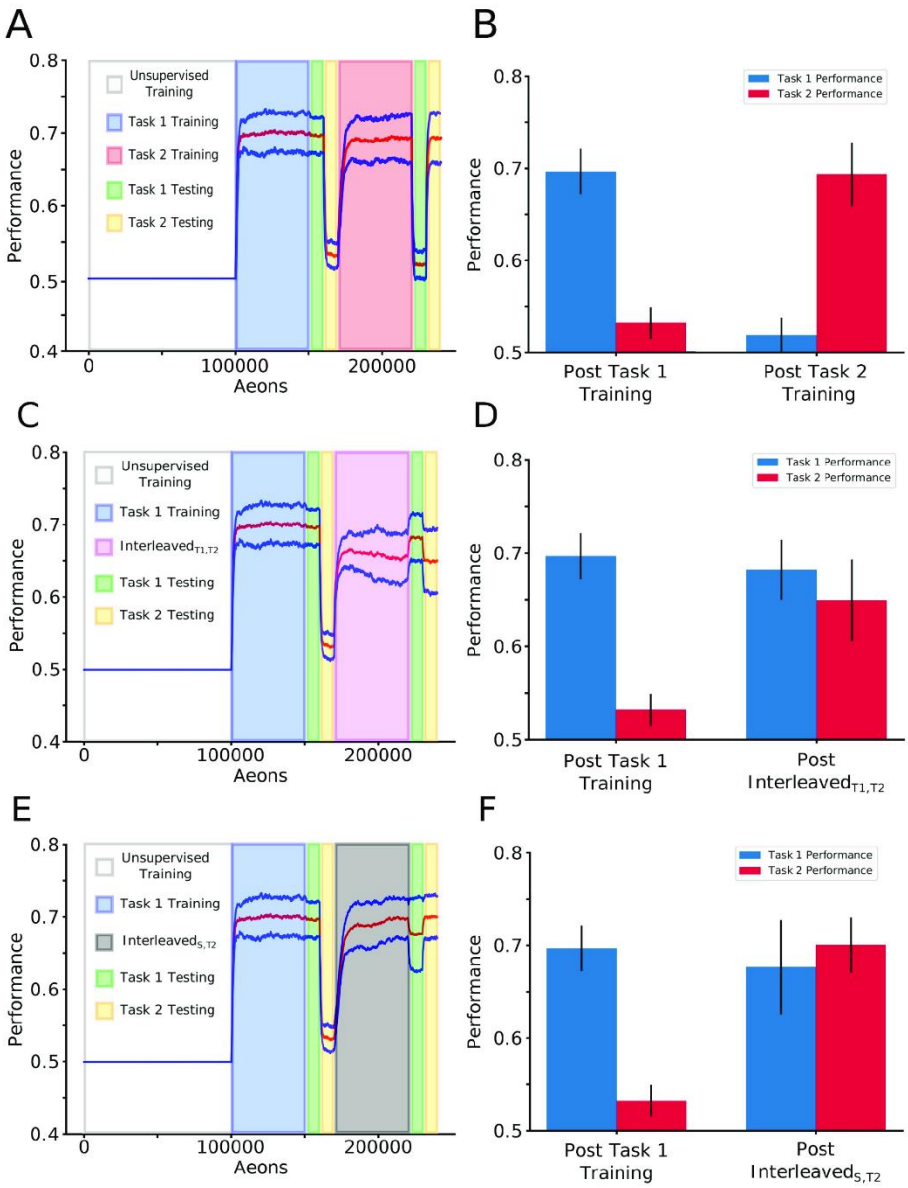


Fig 3. Sleep prevents catastrophic forgetting during new task training. **(A)** Mean performance (red line) and standard deviation (blue lines) over time: unsupervised training (white), Task 1 training (blue), Task 1/2 testing (green/yellow), Task 2 training (red), Task 1/2 testing (green/yellow). **(B)** Mean and standard deviation of performance during testing on Task 1 (blue) and Task 2 (red). Task 2 training after Task 1 training led to Task 1 forgetting. **(C)** Task paradigm similar to that shown in (A) but with Interleaved$_{T1,T2}$ training (pink) instead of Task 2 training. **(D)** Mean and standard deviation of performance during testing on Task 1 (blue) and Task 2 (red).

Interleaved$_{T1,T2}$ training allowed new Task 2 learning without forgetting old Task 1. **(E)** Task paradigm similar to that shown in (A) but with Interleaved$_{S,T2}$ training (gray) instead of Task 2 training. **(F)** Mean and standard deviation of performance during testing on Task 1 (blue) and Task 2 (red). Embedding sleep phases to the new Task 2 training protected old Task 1 memory.

https://doi.org/10.1371/journal.pcbi.1010628.g003

Sleep can protect synaptic configuration from previous training but does not provide training by itself

In simulations presented in Fig 3, during sleep phase, each hidden layer neuron was stimulated by noise, a Poisson distributed spike train, and we ensured that its firing rate during sleep would be close to the mean rate of that neuron firing across all the preceding training sessions. Therefore, intensity of the noise input during Interleaved$_{S,T2}$ was influenced by preceding Task 1 training and could also vary between H neurons. To eliminate the possibility that such input may provide direct Task 1 training during sleep, three additional experiments were conducted. First, we applied Interleaved$_{S,T1}$ phase to a completely naive network. Importantly, even though this network was never trained on Task 2, we used information about hidden layer neuron firing rates after Task 2 training from another experiment. In other words, we artificially took into account Task 2 firing rate data to design random input during sleep to check if this might be sufficient to improve the network performance on Task 2. We found that the network learns Task 1 but Task 2 performance remained at baseline (S4A and S4B Fig). In a second experiment, a similar period of Interleaved$_{S,T1}$ was applied following Task 1 training (S4C and S4D Fig) and we found that it maintained performance on Task 1 but again without any performance gain for Task 2.

In a third experiment, we repeated the sequence shown in Fig 3E, however, during the sleep phase, we provided each hidden layer neuron with a Poisson spike train input which was drawn (independently) from the same distribution, i.e., we used the same input firing rate for all hidden layer neurons determined by the mean firing of the entire hidden layer population as opposed to the private spiking history of individual H neurons in the Fig 3E and 3F experiments (termed Uniform-Noise Sleep (US)). The network's performance under this implementation of noise, Interleaved$_{US,T1}$, (S4E and S4F Fig) was similar to that from our original sleep implementation (see Fig 3E and 3F). Taken together, these results suggest that the properties of the input that drives firing during sleep are not essential to enable replay, any similar to awake random activity in layers H and O is sufficient to prevent forgetting.

Sleep replay protects critical synapses of the old tasks

To reveal synaptic weights dynamics during training and sleep, we next traced "task-relevant" synapses, i.e. synapses identified in the top 10% of the distribution following training on that specific task. We first trained Task 1, followed by Task 2 training (Fig 4A) and we identified "task-relevant" synapses after each task training. Next, we continued by training Task 1 again but we interleaved it with periods of sleep: T1->T2->Interleaved$_{S,T1}$. Sequential training of Task 2 after Task 1 led to forgetting of Task 1, but after Interleaved$_{S,T1}$ Task 1 was relearned while Task 2 was preserved (Fig 4A and 4B), as in the experiments in the previous section (Fig 3C). Importantly, this protocol allowed us to compare synaptic weights after Interleaved$_{S,T1}$ training with those identified as task-relevant after individual Task 1 and Task 2 training (Fig 4C). The structure in the distribution of Task 1-relevant synapses formed following Task 1 training (Fig 4C; top-left) was destroyed following Task 2 training (top-middle) but partially recovered following Interleaved$_{S,T1}$ training (top-right). The distribution structure of Task 2-relevant synapses following Task 2 training (bottom-middle) was not present following Task 1 training (bottom-left) and was partially retained following Interleaved$_{S,T1}$ training (bottom-

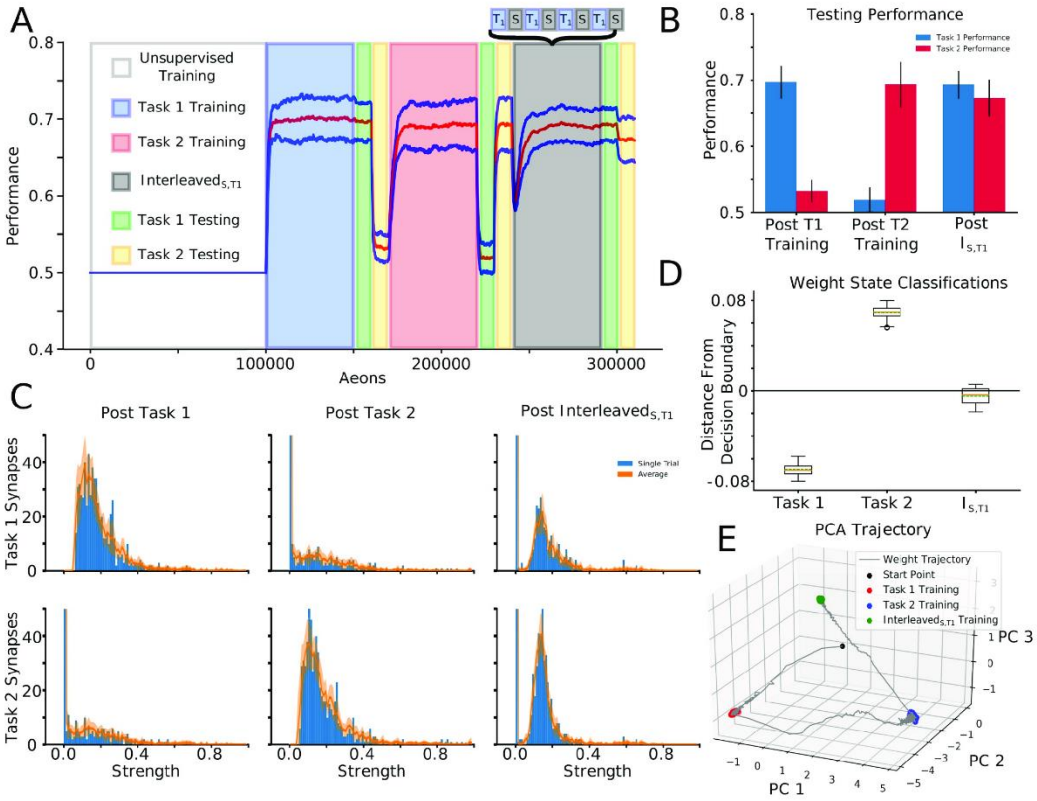


Fig 4. Interleaving periods of new task training with sleep allows integrating synaptic information relevant to new task while preserving old task information. **(A)** Mean performance (red line) and standard deviation (blue lines) over time: unsupervised training(white), Task 1 training (blue), Task 1/2 testing (green/yellow), Task 2 training (red), Task 1/2 testing (green/yellow), Interleaved$_{S,T1}$ training (grey), Task 1/2 testing (green/yellow). Note that performance for Task 2 remains high at the end despite no Task 2 training during Interleaved$_{S,T1}$. **(B)** Mean and standard deviation of performance during testing on Task 1 (blue) and Task 2 (red). **(C)** Distributions of task-relevant synaptic weights (blue bars–single trial, orange line / shaded region–mean / std across 10 trails). The distributional structure of Task 1-relevant synapses following Task 1 training (top-left) is destroyed following Task 2 training (top-middle), but partially recovered following Interleaved$_{S,T1}$ training (top-right). Similarly, the distributional structure of Task 2-relevant synapses following Task 2 training (bottom-middle), which was not present following Task 1 training (bottom-left), was partially preserved following Interleaved$_{S,T1}$ training (bottom-right). **(D)** Box plots with mean (dashed green line) and median (dashed orange line) of the distance to the decision boundary found by an SVM trained to classify Task 1 and Task 2 synaptic weight matrices for Task 1, Task 2, and Interleaved$_{S,T1}$ training across trials. Task 1 and Task 2 synaptic weight matrices had mean classification values of -0.069 and 0.069 respectively, while that of Interleaved$_{S,T1}$ training was -0.0047. **(E)** Trajectory of H to O layer synaptic weights through PC space. Synaptic weights which evolved during Interleaved$_{S,T1}$ training (green dots) clustered in a location of PC space intermediary between the clusters of synaptic weights which evolved during training on Task 1 (red dots) and Task 2 (blue dots).

https://doi.org/10.1371/journal.pcbi.1010628.g004

right). It should be noted that this qualitative pattern can be distinctly observed in a single trial (Fig 4C; Blue Bars), but also generalizes across trials (Fig 4C; Orange Line). Thus, sleep can preserve important synapses while incorporating new ones.

To better understand the effect of Interleaved$_{S,T1}$ training on the synaptic weights, we trained a support vector machine (SVM; see *Method*: *Support Vector Machine Training* for details) to classify the synaptic weight configurations between layers H and O according to whether they serve to perform Task 1 or Task 2 on every trial. Fig 4D shows that the SVMs

robustly and consistently classified the synaptic weight states after Task 1 and Task 2 training while those after Interleaved$_{S,T1}$ fell significantly closer to the decision boundary. This indicates that the synaptic weight matrices which result from Interleaved$_{S,T1}$ training are a mixture of Task 1 and Task 2 states. Using principal components analysis (PCA), we found that while synaptic weight matrices associated with Task 1 and Task 2 training cluster in distinct regions of PC space, Interleaved$_{S,T1}$ training pushes the synaptic weights to an intermediate location between Task 1and Task 2 (Fig 4E). Importantly, the smoothness of this trajectory to its steady state suggests that Task 2 information is never completely erased during this evolution. We take this as evidence that Interleaved$_{S,T1}$ training is capable of integrating synaptic information relevant to Task 1 while protecting Task 2 information.

This analysis applied during interleaved training of Task 1 and Task 2 (Interleaved$_{T1,T2}$), revealed similar results (S5 Fig), suggesting that Interleaved$_{S,T1}$ can enable similar synaptic weights dynamics as Interleaved$_{T1,T2}$ training, but without access to the old task data (old training environment).

Receptive fields of decision-making neurons after sleep represent multiple tasks

To confirm that the network had learned both tasks after Interleaved$_{S,T1}$ training, we visualized the receptive fields of decision-making neurons in layer O (Fig 5; see Fig 2 for comparison). Fig 5A shows the receptive field for the neuron in layer O which controls movement in the upper-left direction. This neuron responded to both horizontal (rewarded for Task 1) and vertical (rewarded for Task 2) orientations in the upper-left quadrant of the visual field. Although it initially appears that this layer O neuron may also be responsive to diagonal patterns in this region, analysis of the receptive fields of neurons in layer H (Fig 5B) revealed that these receptive fields are selective to either horizontal food particles (left six panels; rewarded for Task 1) or vertical food particles (right six panels; rewarded for Task 2) in the upper-left quadrant of the visual field. Other receptive fields were responsible for avoidance of punished particles for both tasks (see examples in Fig 5B, bottom-middle-right and bottom-middle-left). Thus, the network utilizes one of two distinct sets of layer H neurons, selective for either Task 1 or Task 2, depending on which food particles are present in the environment. To validate these qualitative results we inspected the PRM metrics for all food particle orientations across ten trials following Interleaved$_{S,T1}$ training. The comparatively high mean values for horizontal and vertical food particle orientations revealed the network's movement was significantly driven by these rewarded food particle orientations (horizontal and vertical), quantifying multitask memory integration into the network's synaptic weight matrix. (S3C Fig).

Periods of sleep allow for integration of a new task memory without interference through renormalization of task-relevant synapses

To visualize synaptic weight dynamics during Interleaved$_{S,T1}$ training, traces of all synapses projecting to a single representative layer O neuron were plotted (Fig 6A). As in Fig 4, we wanted to monitor task specific synapses, so we used the training paradigm of T1->T2->Interleaved$_{S,T1}$, and Task 1 and Task 2 relevant synapses were identified after each individual task training. At the onset of Interleaved$_{S,T1}$ training (i.e. 240,000 aeons), the network was only able to perform on Task 2, meaning the strong synapses in the network were specific to this task. These synapses were represented by a cluster ranging from ~0.08 to ~0.4; the rest of synapses grouped near 0. As Interleaved$_{S,T1}$ training progressed, Task 1 specific synapses moved to the strong cluster and some, presumably less important, Task 2 synapses moved to the weak

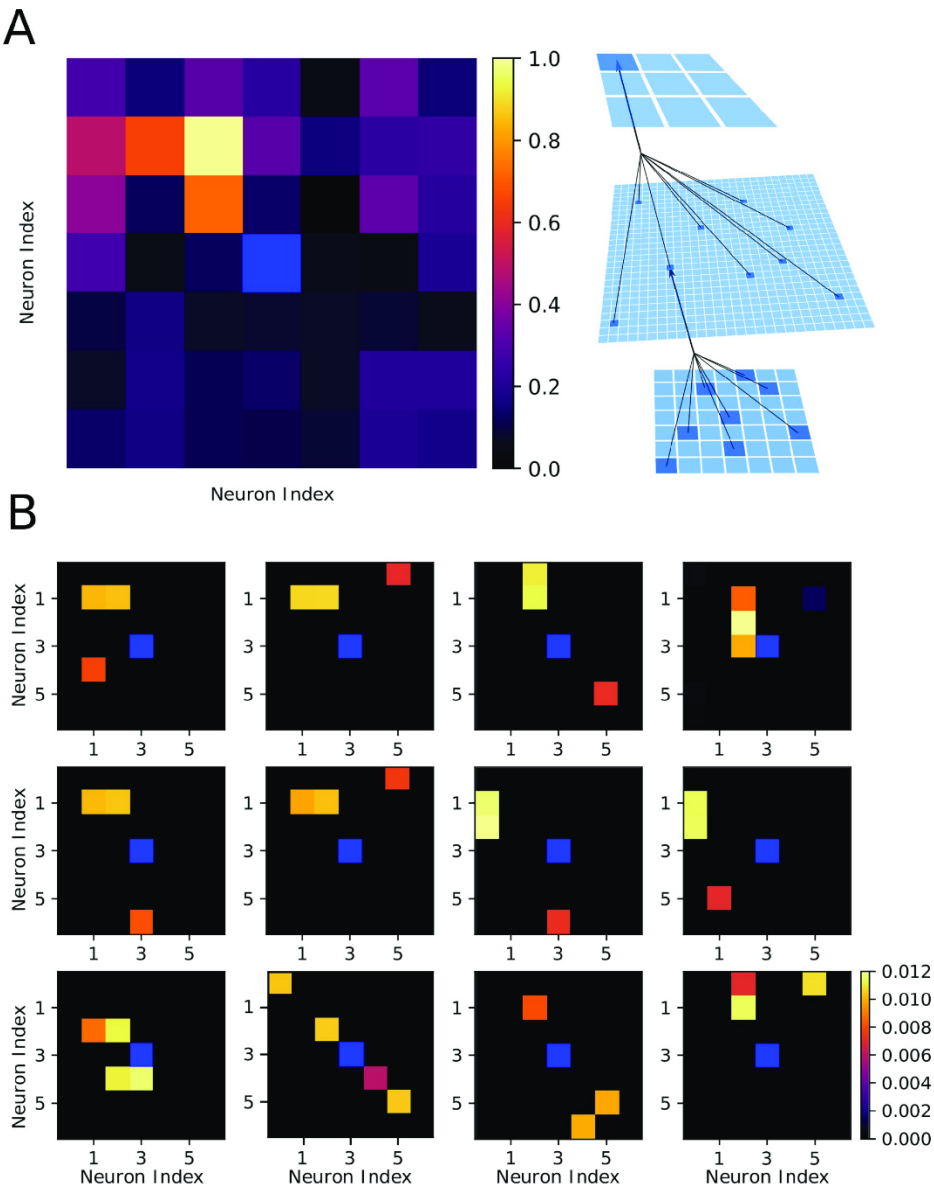


Fig 5. Receptive fields following interleaved Sleep and Task 1 training reveal how the network can multiplex the complementary tasks. (A) Left, Receptive field of the output layer neuron controlling movement to the upper-left direction following interleaved sleep and Task 1training. This neuron has a complex receptive field capable of responding to horizontal and vertical orientations in the upper-left quadrant of the visual field. Right, Schematic of the connectivity between layers. **(B)** Examples of receptive fields of hidden layer neurons which synapse strongly onto the output neuron from (A) after interleaved Sleep and Task 1 training. The majority of these neurons selectively respond to horizontal food

particles (left half) or vertical food particles (right half) in the upper-left quadrant of the visual field, promoting movement in that direction and acquisition of the rewarded patters.

https://doi.org/10.1371/journal.pcbi.1010628.g005

cluster. After a period of time the rate of transfer decreased and the total number of synapses in each group stabilized, showing that the network approached equilibrium (Fig 6B).

To visualize how sleep renormalizes task relevant synapses, we plotted two-dimensional weight distributions for T1->T2 (Fig 6C) and T2->Interleaved$_{S,T1}$ (Fig 6D) experiments (see *Methods*: *2-D Synaptic Weight Distributions* for details). To establish a baseline, in Fig 6C (left) the weight state at the end of Task 1 training (X-axis) (see overall timeline of this experiment in Fig 4A) was compared to itself (Y-axis). This formed a perfectly diagonal plot. The next comparison (Fig 6C, middle) was between the weight state after Task 1 training (X-axis) and a time early on Task 2 training (Y-axis). At that time, synapses were only able to modify their strength slightly, causing most points to lie close to the diagonal. As training on Task 2 continued, synapses moved far away from the diagonal (Fig 6C, right). Two trends were observed: (a) set of synapses that had a strength near zero following Task 1 training increased strength following Task 2 training (Fig 6D, right, red dots along Y-axis); (b) many strongly trained by Task 1 synapses were depressed down to zero (Fig 6C, right, red dots along X-axis). The latter illustrates the effect of catastrophic forgetting—complete overwriting of the synaptic weight matrix caused performance of Task 1 to return to baseline after training on Task 2.

Does sleep prevent overwriting of the synaptic weight matrix? Fig 6D plots used the weight state at the end of training Task 2 as a reference which is then compared to different times during Interleaved$_{S,T1}$ training. The first two plots (Fig 6D, left/middle) are similar to those in Fig 6C. However, after continuing Interleaved$_{S,T1}$ training (Fig 6D, right) many synapses that were strong following Task 2 training were not depressed to zero but rather were pushed to an intermediate strength (note cluster of points parallel to X-axis). Thus, Interleaved$_{S,T1}$ training preserved strong synapses from a previously learned task while also introducing new strong synapses to perform the new task.

Can we prevent old task forgetting simply by freezing a fraction of old task-relevant synapses to prevent their damage by new training? We found that freezing 1% of Task 1-relevant weights allowed Task 2 to be learned, but was not capable of preserving Task 1 (S6A Fig). Freezing 5% of Task 1-relevant weights (S6B Fig) resulted in modest performance on both tasks, but significantly below that seen after Interleaved$_{S,T2}$ (see Fig 3F). Finally, freezing 10% of Task 1-relevant weights (S6C Fig) was capable of fully preserving Task 1 performance, but prevented Task 2 from being learned.

Thus, in all cases, some degree of retroactive or prospective interference was observed highlighting the fact that the sleep-like phase performs a significantly more sophisticated modification to the weight matrix than simply freezing (or amplifying) task relevant synapses. Sleep is capable of intelligently selecting which certain strong synapses to maintain in addition to which weak synapses should be strengthened. Indeed, the sleep phase results in a large cluster of weights being renormalized around an intermediate value of synaptic strength in the network. This may also explain why we observed somewhat better overall performance (combined performance on both tasks) after sleep compare with interleaved training (see Fig 3). Indeed, interleaved training requires repetitive activation of the entire memory pattern, so if different memory patterns compete for synaptic resources then each phase of interleaved training will enhance one memory trace but damage the others. This is in contrast to spontaneous replay during sleep when only task specific subsets of neurons and synapses may be involved in each replay episode. It is worth mentioning that freezing a fraction of synaptic weights that are most relevant to old tasks (however, implemented in more complex form) is

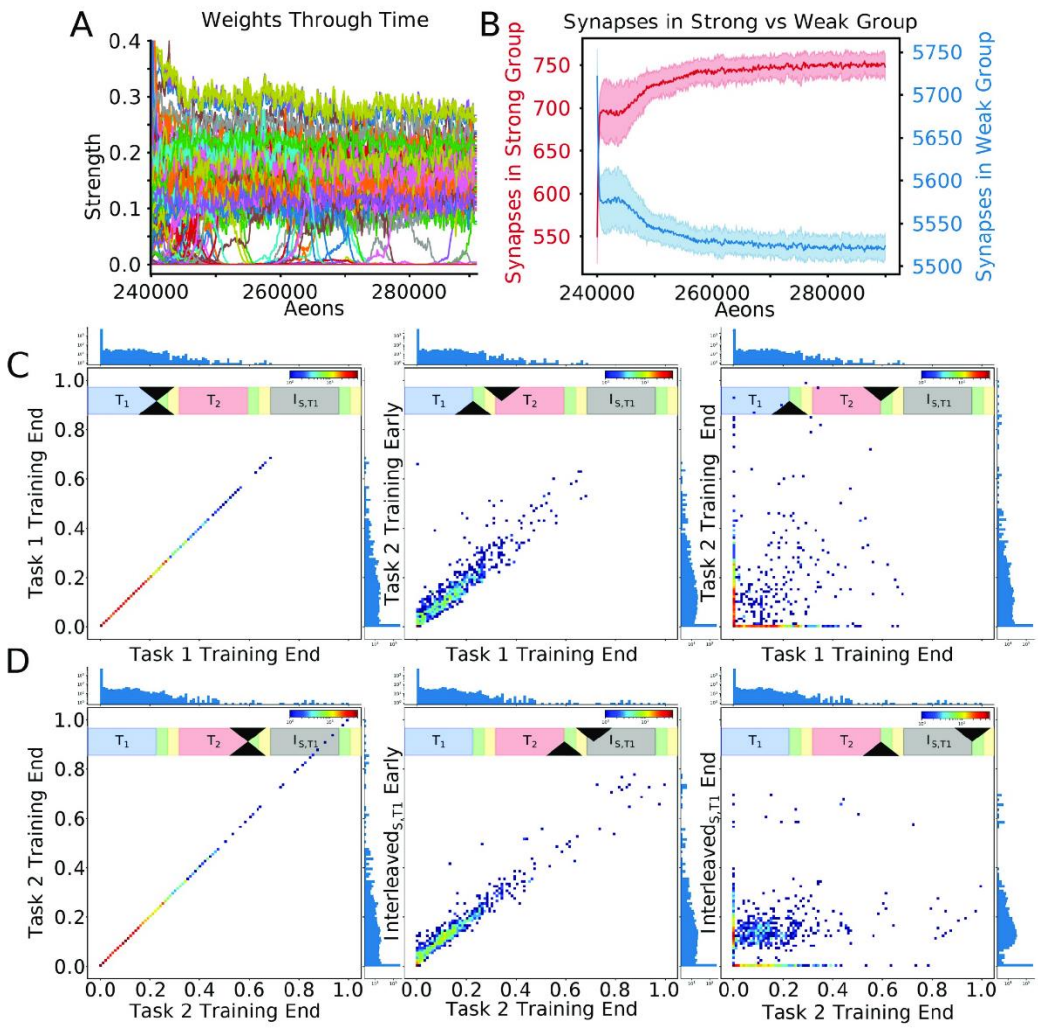


Fig 6. Periods of sleep allow learning Task 1 without interference with old Task 2 through renormalization of task-relevant synapses. **(A)** Dynamics of all incoming synapses to a single output layer neuron during Interleaved$_{S,T1}$ training shows the synapses separate into two clusters. The network was trained in the following order: T1->T2->Interleaved$_{S,T1}$. **(B)** Number of synapses in the strong (red) and weak (blue) clusters during Interleaved$_{S,T1}$. **(C)** Two-dimensional histograms illustrating synaptic weights dynamics. For each plot, the x-axis represents synaptic weight after Task 1 training and the y-axis represents the synaptic weight at a different point in time (Scale bar: brown—50 synapses/bin, blue—1 synapse/bin). One-dimensional projections along top and right sides show the global distribution of synapses at the time slices for a given plot. **(D)** Same as (C) except the x-axis refers to the end of Task 2 training. Note, that after a full period of Interleaved$_{S,T1}$ training (right), weak synapses were recruited to support Task 1 (red cluster along the y-axis) and many Task 2 specific synapses remained moderately strong (blue cluster along x-axis).

https://doi.org/10.1371/journal.pcbi.1010628.g006

one of the approaches in machine learning to avoid catastrophic forgetting–Elastic Weight Consolidation [7].

Periods of interleaved sleep and new task training push the network weight state towards the intersection of Task 1 and Task 2 synaptic weights configuration manifolds

Can many distinct synaptic weight configurations support a given task, or is each task supported by a unique synaptic connectivity matrix? Our previous analysis suggests that each task can be served by at least two different configurations–one unique for that task (Task 1 or Task 2) and another one that supports both Task 1 and Task 2. To further explore this question and to identify possible task-specific solution manifolds (M_{T1} and M_{T2}) and their intersection ($M_{T1\cap T2}$) in synaptic weights space, we used multiple trials of Task 1 and Task 2 training to sample the manifolds (Fig 7A). Here, red/blue dots indicate an exclusive high degree of performance on Task 1/2 respectively, while cyan and greed dots indicate states where the network is able to perform on both tasks simultaneously. Since this analysis was generated utilizing a wide variety of simulation paradigms with many corresponding trials differing in randomness, we believe it allows us to draw generalized conclusions. We therefore interpret these results as evidence that synaptic weight space includes a manifold, M_{T1}, where different configurations of weights (red, green, cyan dots) all allow for Task 1 to perform well. This manifold intersects with another one, M_{T2}, where different weights configurations (blue, green, cyan dots) are all suitable for Task 2. Fig 7B and 7C show 2D dimensionality reductions to PCA space, and include trajectories in addition to end states. One can see that PC 1 seems to capture the extent to which a synaptic weight configuration is associated with Task 1 (positive values) or Task 2 (negative values), while PC 2 and PC 3 capture the variance in synaptic weight configurations associated with Task 1 and Task 2, respectively. Note, the trajectories through this space (red/blue lines) during $\text{Interleaved}_{\text{T1,T2}}$ and $\text{Interleaved}_{\text{S,T1/T2}}$ training would also belong to the respective task manifolds as performance on the old tasks was never lost in these training scenarios.

We next calculated the distance from the current synaptic weight configurations to M_{T1} (Fig 7D), M_{T2} (Fig 7E), and $M_{T1\cap T2}$ (Fig 7F; see *Methods*: *Distance from Solution Manifolds* for details) during different training protocols. Fig 7D and 7E show that while Sequential (T1->T2 or T2->T1) training causes synaptic weight configurations to diverge quickly from its initial solution manifold (i.e. M_{T1} or M_{T2}) and to remain far (suggesting quick forgetting of the original task), both $\text{Interleaved}_{\text{T1,T2}}$ and $\text{Interleaved}_{\text{S,T1/T2}}$ training cause synaptic weight configurations to stay relatively close to the initial solution manifold as a new task was learned. (Note, that we certainly under sampled M_{T1} and M_{T2}, which may explains initial distance increase.) Importantly, Fig 7F shows that both $\text{Interleaved}_{\text{T1,T2}}$ and $\text{Interleaved}_{\text{S,T1/T2}}$ training cause synaptic weight configurations to smoothly converge towards $M_{T1\cap T2}$, while Sequential training avoids this intersection entirely.

In Fig 7G we present a schematic depiction of these results. The task-specific manifolds, M_{T1} and M_{T2}, are depicted in 3D as two volumes whose boundaries are defined by two orthogonal elliptic paraboloids with opposite orientation. The ellipsoidal intersection approximates the volume comprising $M_{T1\cap T2}$. Fig 7H and 7I depict a cartoon of trajectories taken by the network in this space following Task 2 and Task 1 training, respectively. Sequential training causes the network to jump directly from one task-specific solution manifold to the other, resulting in catastrophic forgetting. In contrast, interleaving new task training with sleep ($\text{Interleaved}_{\text{S,T1/T2}}$) prevents catastrophic forgetting by keeping the network close to the old task solution manifold as it converges towards $M_{T1\cap T2}$ –a region capable of supporting both tasks simultaneously.

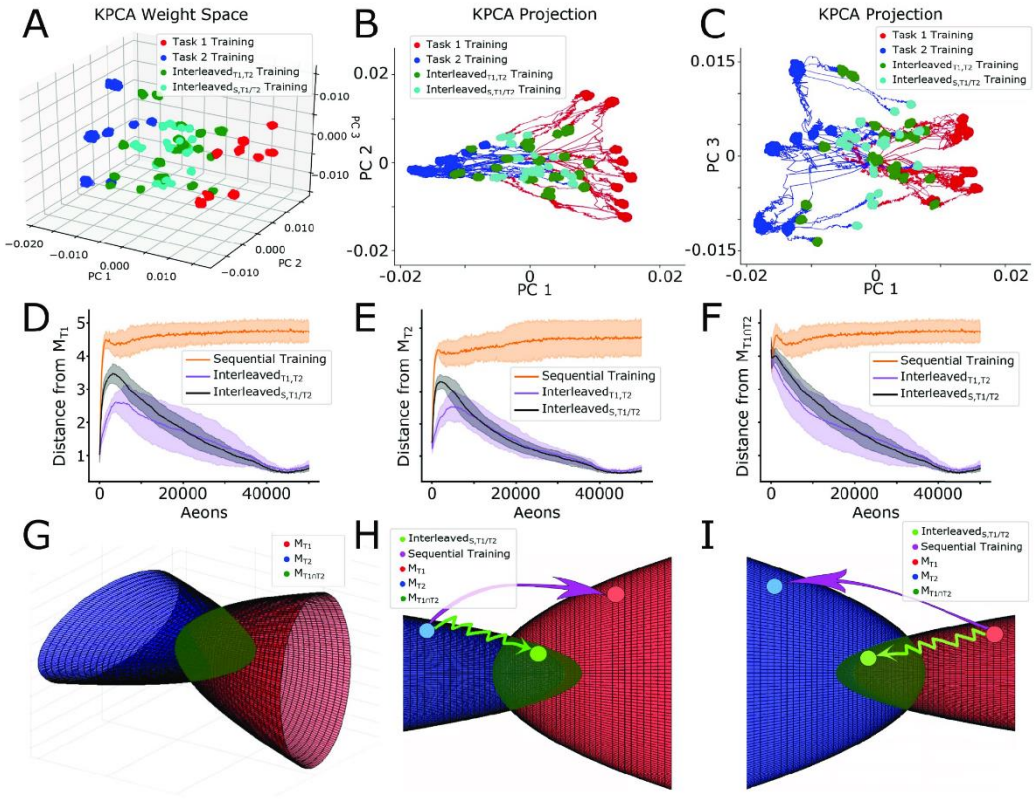


Fig 7. Periods of sleep push the network towards the intersection of Task 1 and Task 2 synaptic weight manifolds. (A-C) Low-dimensional visualizations of the synaptic weight configurations of 10 networks obtained through kPCA for 3-dimensions (A) and 2-dimensions (B-C). Synaptic weight configurations taken from the last fifth of Task 1 (red dots), Task 2 (blue dots), Interleaved$_{T1,T2}$ (green dots), and Interleaved$_{S,T1/T2}$ (cyan dots) training are shown. Trajectories resulting from Interleaved$_{T1,T2}$ and Interleaved$_{S,T1/T2}$ training following Task 1 (Task 2) training are shown in red (blue). **(D-F)** Average (solid lines) and standard deviation (shaded regions) of the Euclidean distances between the current synaptic weight configuration and M_{T1} (D), M_{T2} (E), and $M_{T1\cap T2}$ (F) during Sequential (orange), Interleaved$_{T1,T2}$ (purple), and Interleaved$_{S,T1/T2}$ (black) training. **(G)** Cartoon illustration of the task-specific point-sets shown in (A-C) as solution manifolds M_{T1} (red) and M_{T2} (blue). M_{T1} and M_{T2} can be thought of as two volumes with boundaries defined by the interiors of oppositely oriented elliptic paraboloids which intersect orthogonally defining an approximately ellipsoidal volume near the origin ($M_{T1\cap T2}$; dark green). **(H, I)** Sequential training (pink arrow) causes the network to jump from one solution manifold to the other while avoiding $M_{T1\cap T2}$, while Interleaved$_{S,T1/T2}$ training (light green arrow) keep the network close to the initial solution manifold as it converges towards $M_{T1\cap T2}$.

https://doi.org/10.1371/journal.pcbi.1010628.g007

Discussion

We report that a multi-layer spiking neural network utilizing reinforcement learning exhibits catastrophic forgetting upon sequential training of two complementary complex foraging tasks, however the problem is mitigated if the network is allowed, during new task training, to undergo intervening periods of spontaneous reactivation which are equivalent to the periods of sleep in a biological brain. Old task was spontaneously replayed during sleep, therefore interleaving new task training with sleep was effectively equivalent to explicit interleaved training of the old and new tasks without the need to store and train on previous task data or environments. At the synaptic level, training a new task alone led to complete overwriting of

synaptic weights responsible for the old task. In contrast, interleaving periods of reinforcement learning on a new task with periods of unsupervised plasticity during sleep preserved critical old task synapses to avoid forgetting and enhanced synapses relevant for a new task to allow new task learning. Thus, in synaptic weight space, the network weight configuration was pushed towards the intersection of the manifolds representing synaptic weight configurations associated with individual tasks—an optimal compromise for performing both tasks.

The critical role that sleep plays in learning and memory is supported by a vast, interdisciplinary literature spanning both psychology and neuroscience [16,22,43–45]. Specifically, it has been suggested that REM sleep supports the consolidation of non-declarative or procedural memories while non-REM sleep supports the consolidation of declarative memories [16,21,22]. In particular, REM sleep has been shown to be important for the consolidation of memories of hippocampus-independent tasks involving perceptual pattern separation, such as the texture discrimination task [16,46]. Despite the difference in the cellular and network dynamics during these two stages of sleep [16,22], both are thought to contribute to memory consolidation through repeated reactivation, or replay, of specific memory traces acquired during learning [16,21,39,44,47–49]. These studies suggest that through replay, sleep can support the process of off-line memory consolidation to circumvent the problem of catastrophic forgetting.

From mechanistic perspective, the sleep phase in our model protects old memories by enabling spontaneous reactivation of neurons and changing synapses responsible for previously learned tasks. We previously reported that in the thalamocortical model a sleep phase may enable replay of spike sequences learned in awake to improve post-sleep performance [38–40] and to protect old memories from catastrophic forgetting [41]. Here we found, however, that a single episode of new task training using reinforcement learning could quickly erase old memories to the point that they cannot be recovered by subsequent sleep. The solution was similar to how the brain slowly learns procedural (hippocampal-independent) memories [16,21,22,46,50]. Each episode of new task training improves new task performance only slightly but also damages slightly synaptic connectivity responsible for the older task. Subsequent sleep phases enable replay that preferentially benefits the strongest synapses, such as those from old memory traces, to allow them to recover.

We found that multiple distinct configurations of synaptic weights can support each task, suggesting the existence of task specific solution manifolds in synaptic weight space. Sequential training of new tasks makes the network to jump from one solution manifold to another, enabling memory for the most recent task but erasing memories of the previous tasks. Interleaving new task training with sleep phases enables the system to evolve towards intersection of these manifolds where synaptic weight configurations can support multiple tasks (a similar idea was recently proposed in the machine learning literature to minimize catastrophic interference by learning representations that accelerate future learning [51]). From this point of view having multiple episodes of new task training interleaved with multiple sleep episodes allows gradual convergence to the intersection of the manifolds representing old and new tasks, while staying close to the old task manifold. In contrast, a single long episode of new task learning would push the network far away from the old task manifold making it impossible to recover by subsequent sleep.

Although classical interleaved training of the old and new tasks showed similar performance results in our model as interleaving new task training with sleep, we believe the latter to be superior on the following theoretical grounds. Classical interleaved training will necessarily cause the system to oscillate about the optimal location in synaptic weight space which can support both tasks because each training cycle uses a cost function specific to only a single task. While this can be ameliorated with a learning rate decay schedule, the system is never actually optimizing for the desired dual-task state. Sleep, on the other hand, can support not

only replays of the old task, but also support replays which are a mixture of both tasks [41,52,53]. Thus, through unsupervised plasticity during sleep replay, the system is able to perform approximate optimization for the desired dual-task (or multi-task) state.

Our results are in line with a large body of literature suggesting that interleaved training is capable of mitigating catastrophic forgetting in ANNs [4,10,11] and SNNs [12,13], which led to a number of replay-like algorithms involving storing a subset of previous veridical inputs and mixing them with more recent inputs to update the networks (reviewed in [9]). The novel contribution from our study is that the data intensive process of storing old data and using them for retraining can be avoided in SNN by implementing periods of noise-induced spontaneous reactivation during new task training; similar to how brains undergo offline consolidation periods during sleep resulting in reduced retroactive interference to previously learned tasks [16,50]. Indeed, we recently successfully implemented a similar approach in feedforward ANNs, where sleep-like phase prevented catastrophic forgetting and improved generalization and adversarial robustness [54–56]. And our results are in line with previous work done in humans showing that perceptual learning tasks are subject to retroactive interference by competing memories without an intervening period of REM sleep [21,46]. Moreover, performance on visual discrimination tasks in particular have been shown to steadily improve over successive nights of sleep [46], consistent with our findings that interleaving multiple periods of sleep with novel task learning leads to optimal performance on each task.

In comparing our modeling results to those found in the literature on biological learning, it is important to note an important difference in the "baseline" state of an animal undergoing an experimental training condition versus a neural network model. In our model, and indeed in all neural network models, the system begins as a "blank slate" without knowledge of any previous learning or competing demands. In contrast, animals under experimental training paradigms have a wealth of experiences which would serve as priors to bias the subsequent learning during training, leading potentially to proactive interference. Moreover, training is typically conducted across multiple days, with intervening periods during which the animal will be subject to an array of various task-irrelevant stimuli and organismal demands possibly leading to retroactive interference. Both of these ensure that the baseline state of the animal entering a given training session is far from that of the "blank slate" a neural network model enters with, as well as that recently learned memories may start degrading quickly in the brain while the network weights remain unchanged post training (unless new task is explicitly trained). Due to this stark differences, we focus our attention on the interference phenomena which follow training on an initial task as opposed to initial learning. Viewed from this perspective, initial task training in our network can serve a similar role to the prior personal history of an animal subject.

While our model represents a dramatic simplification of a living system, we believe that it captures some important elements of how animal and human brains interact with the external world. The primary visual system is believed to employ a sequence of processing steps when visual information is increasingly represented by neurons encoding higher level features [35–37]. In insects, complex patterns of olfactory receptors activation by odors are encoded by sparse patterns of the mushroom body Kenyon cells firing [57–59]. This processing step is also similar to the function performed by convolutional layers of an ANN [34] and it was reduced to very simple convolution from the input to hidden layer in our model. Subsequently, in the vertebrate brain, associative areas and motor cortex are trained to make decisions based on reward signals released by neuro modulatory centers [10,60–62]. In insects, Kenyon cells make plastic (subject to rewarded STDP) projections to the lobes [27,63]. This was reduced in our model to synaptic projections from the hidden to output (decision making) layer implementing rewarded STDP to learn a task [30–32]. While NREM sleep in vertebrates is characterized

by complex patterns of synchronized neuronal activity [16], REM sleep is characterized by low-synchronized firing [42], similar to activity during sleep-like phase in our model and paradoxical sleep with similar properties has been reported in honeybee and fruit fly [64–66].

Our study predicts synaptic level mechanisms of how sleep-based memory reactivation can protect old memory traces during training of a new interfering memory task. It suggests that, at least for procedural memories that are directly encoded to the cortical network connectivity during new training, multiple episodes of training interleaved with periods of sleep provide necessary mechanisms to prevent forgetting old memories. Interleaving new task training with sleep enables the connectivity matrix to evolve towards the joint synaptic weight configuration, representing the intersection of manifolds supporting individual tasks. Sleep makes this possible by replaying old memory traces without explicit usage of the old training data.

Methods

Environment

Foraging behavior took place in a virtual environment consisting of a 50x50 grid with randomly distributed "food" particles. Each particle was two pixels in length and could be classified into one of four types depending on its orientation: vertical, horizontal, positively sloped diagonal, or negatively sloped diagonal. During the initial unsupervised training period, the particles are distributed at random with the constraints that each of the four types are equally represented and no two particles can be directly adjacent. During training and testing periods only the task-relevant particles were present. When a particle was acquired as a result of the virtual agent moving, it was removed from its current location (simulating consumption) and randomly assigned to a new location on the grid, again with the constraint that it not be directly adjacent to another particle. This ensures a continuously changing environment with a constant particle density. The density of particles in the environment was set to 10%. The virtual agent can see a 7x7 grid of squares (the "visual field") centered on its current location and it could move to any adjacent square, including diagonally, for a total of eight directions.

Network structure

The network was composed of 842 spiking reduced (map-based) model neurons (see *Methods*: *Map-based neuron model* below) [67,68], arranged into three feed-forward layers to mimic a basic biological circuit: a 7x7 input layer (I), a 28x28 hidden layer (H), and a 3x3 output layer (O) with a nonfunctional center neuron (Fig 1). Input to the network was simulated as a set of suprathreshold inputs to the neurons in layer I, equivalent to the lower levels of the visual system, which represent the position of particles in an egocentric reference frame relative to the virtual agent (positioned in the center of the 7x7 visual field). The most active neuron in layer O, playing the role of biological motor cortex, determined the direction of the subsequent movement. Each neuron in layer H, which can be loosely defined as higher levels of the visual system or associative cortex, received excitatory synapses from 9 randomly selected neurons inlayer I. These connections initially had random strengths drawn from a normal distribution. Each neuron in layer H connected to every neuron in layer O with both an excitatory (*Wij*) and an inhibitory (*WIij*) synapse. This provided an all-to-all connectivity pattern between these two layers and accomplished a balanced feed-forward inhibition [69] found in many biological structures [69–74]. Initially, all these connections had uniform strengths and the responses in layer O were due to the random synaptic variability. Random variability was a property of all synaptic interactions between neurons and was implemented as variability in the magnitude of the individual synaptic events.

Policy

Simulation time was divided up into epochs of 600 timesteps, each roughly equivalent to 300 ms. At the start of each epoch the virtual agent received input corresponding to locations of nearby particles within the 7x7 "visual field". Thus 48 of the 49 neurons in layer I received input from a unique location relative to the virtual agent. At the end of the epoch the virtual agent made a single move based on the activity in layer O. If the virtual agent moved to a grid location with a "food" particle present, the particle was removed and assigned to a randomly selected new location.

Each epoch was of sufficient duration for the network to receive inputs, propagate activity forward, produce outputs, and return to a resting state. Neurons in layer I which represent locations in the visual field containing particles received a brief pulse of excitatory stimulation sufficient to trigger a spike; this stimulation was applied at the start of each movement cycle (epoch). At the end of each epoch the virtual agent moved according to the activity which has occurred in layer O. Each simulation consisted of millions of these movement cycles / epochs, therefore a unit of time was introduced termed aeon (1 aeon = 100 epochs) for concise reporting.

The activity in layer O controlled the direction of the virtual agent's movement. Each of the neurons in layer O mapped onto a specific direction (i.e. one of the eight adjacent locations or the current location). The neuron in layer O which spiked the greatest number of times during the first half of the epoch defined the direction of movement for that epoch. If there was a tie, the direction was chosen at random from the set of tied directions. If no neurons in layer O spiked, the virtual agent continued in the direction it had moved during the previous epoch.

There was a 1% chance on every move that the virtual agent would ignore the activity inlayer O and instead move in a random direction. Moreover, for every movement cycle that passed without the virtual agent acquiring a particle, this probability was increased by 1%. The random variability promoted exploration vs exploitation dynamics and essentially prevented the virtual agent from getting stuck in movement patterns corresponding to infinite loops. While biological systems could utilize various different mechanisms to achieve the same goal, the method we implemented was efficient and effective for the scope of our study.

Neuron models

For all neurons we used spiking model identical to the model used in in [14,15] that can be described by the following set of difference equations [68,75,76]:

$$V_{n+1} = f_\alpha(V_n, I_n + \beta_n),$$

$$I_{n+1} = I_n - \mu(V_n + 1) + \mu\sigma + \mu\sigma_n,$$

where Vn is the membrane potential, In is a slow dynamical variable describing the effects of slow conductances, and n is a discrete time-step (0.5 ms). Slow temporal evolution of In was achieved by using small values of the parameter $\mu << 1$. Input variables β_n and σ_n were used to incorporate external current $I^{ext}n$ (e.g. background synaptic input): $\beta_n = \beta^e I^{ext}{}_n$, $\sigma_n = \sigma^e I^{ext}{}_n$. Parameter values were set to $\sigma = 0.06$, $\beta^e = 0.133$, $\sigma^e = 1$, and $\mu = 0.0005$. The nonlinearity $f\alpha$ (V_n, I_n) was defined in the form of the piece-wise continuous function:

$$f_\alpha(V_n, I_n) = \begin{cases} \alpha(1 - V_n)^{-1} + I_n, & V_n \le 0 \\ \alpha + I_n, & 0 < V_n < \alpha + I_n \& V_{n-1} \le 0 \\ -1 & \alpha + I_n \le V_n \text{ or } V_{n-1} > 0, \end{cases}$$

where $\alpha = 3.65$. This model is very computationally efficient, and, despite its intrinsic low dimensionality, produces a rich repertoire of dynamics capable of mimicking the dynamics of Hodgkin-Huxley type neurons both at the single neuron level and in the context of network dynamics [68,75,77].

To model the synaptic interactions, we used the following piece-wise difference equation:

$$g_{n+1}^{syn} = \gamma g_n^{syn} + \begin{cases} (1 - R + 2XR) g_{syn}/W_j, & spike_{pre} \\ 0, & \text{otherwise,} \end{cases}$$

$$I_n^{syn} = -g_n^{syn}(V_n^{post} - V_{rp}).$$

Here *gsyn* is the strength of the synaptic coupling, modulated by the target rate *Wj* of receiving neuron *j*. Indices *pre* and *post* stand for the pre- and post-synaptic variables, respectively. The first condition, *spikepre*, is satisfied when the pre-synaptic spikes are generated. Parameter γ controls the relaxation rate of synaptic current after a presynaptic spike is received ($0 \leq \gamma < 1$). The parameter *R* is the coefficient of variability in synaptic release. The standard value of *R* is 0.12. *X* is a random variable sampled from a uniform distribution with range [0, 1]. Parameter *Vrp* defines the reversal potential and, therefore, the type of synapse (i.e. excitatory or inhibitory). The term (1-*R*+2*XR*) introduces a variability in synaptic release such that the effect of any synaptic interaction has an amplitude that is pulled from a uniform distribution with range [1-*R*,1+*R*] multiplied by the average value of the synapse.

Synaptic plasticity

Synaptic plasticity closely followed the rules introduced in [14,15]. A rewarded STDP rule [30–33] was operated on synapses between layers H and O while a standard STDP rule operated on synapses between layers I and H. A spike in a post-synaptic neuron that directly followed a spike in a pre-synaptic neuron created a *pre before post* event while the converse created a *post before pre* event. Each new post-synaptic (pre-synaptic) spike was compared to all pre-synaptic (post-synaptic) spikes with a time window of 120 iterations.

The value of an STDP event (trace) was calculated using the following equation [28,29]:

$$p = \frac{-|t_r - t_p|}{T_c},$$

$$tr_k = Ke^p$$

where t_r and t_p are the times at which the pre- and post-synaptic spike events occurred respectively, *Tc* is the time constant and is set to 40 ms, and *K* is maximum value of the trace tr_k and is set to -0.04 for a *post before pre* event and 0.04 for a *pre before post* event.

A trace was immediately applied to synapse between neurons in layers I and H. However, for synapses between neurons in layers H and O the traces were stored for 6 epochs after its creation before being erased. During storage, a trace had an effect whenever there was a rewarding or punishing event. In such a case, the synaptic weights are updated as follows:

$$W_{ij} \leftarrow W_{ij} \prod_k^{traces} \left(1 + \frac{W_{i0}}{W_i} * \Delta_k\right),$$

$$\Delta_k = S_{rp}\left(\frac{tr_k}{t - t_k + c}\right) \frac{Sum_{tr}}{Avg_{tr}},$$

$$Sum_{tr} = \sum_{k}^{traces} \frac{tr_k}{t - t_k + c},$$

$$Avg_{tr} \leftarrow (1 - \delta) Avg_{tr} + \delta Sum_{tr},$$

where t is the current timestep, S_{rp} is a scaling factor for reward/punishment, trk is the magnitude of the trace, tk is the time of the trace event, c is a constant (= 1 epoch) used for decreasing sensitivity to very recent spikes, $W_i = \Sigma_j W_{ij}$ is the total synaptic strength of all connections from the neuron i in layer H to all neurons in layer O, W_{i0} is a constant that is set to the initial value(*target value*) of Wi at the beginning of the simulation. The term W_{i0}/W_i helped to keep the output weight sum close to the initial target value. The effect of these rules was that neurons with lower total output strength could increase their output strength more easily.

The network was rewarded when the virtual agent moved to a location which contained a particle from a "food" pattern (horizontal in Task 1, vertical in Task 2) and S_{rp} = 1, and received a punishment of S_{rp} = -0.001 when it moved to a location with a particle from a neutral pattern(negative/positive diagonal in Task 1/2). A small punishment of S_{rp} = -0.0001 was applied if the agent moved to a location without a particle present to help the virtual agent learn to acquire "food" as rapidly as possible. During periods of sleep the network received a constant reward of S_{rp} = 0.5 on each movement cycle.

To ensure that neurons in layer O maintained a relatively constant long-term firing rate, the model incorporated homeostatic synaptic scaling which was applied every epoch. Each timestep, the total strength of synaptic inputs $W_j = \Sigma_i W_{ij}$ to a given neuron in layer O was set equal to the target synaptic input W_{j0} –a slow variable which varied over many epochs depending on the activity of the given neuron in layer O–which was updated according to:

$$W_{j0} \leftarrow \begin{cases} W_{j0}(1 + D_{tar}) \text{ spike rate} < \text{target rate} \\ W_{j0}(1 - D_{tar}) \text{ spike rate} > \text{target rate} \end{cases}$$

To ensure that the net synaptic input W_j to any neuron was unaffected by plasticity events at the individual synapses at distinct timesteps and equal to W_{j0}, we implemented a scaling process akin to heterosynaptic plasticity which occurs after each STDP event. When any excitatory synapse of neuron in layer O changed in strength, all other excitatory synapses received by that neuron were updated according to:

$$W_{ij} \leftarrow W_{ij} \frac{W_{j0}}{\sum_i W_{ij}}$$

Additionally, all inhibitory synapses were modified via a similar heterosynaptic update rule following each STDP event where the strength of every outgoing inhibitory weight from a given neuron was set to the negative mean of all outgoing excitatory synapses of that same neuron. More rigorously:

$$WI_{ij} \leftarrow -\frac{1}{|j|} \sum_j W_{ij}$$

Simulated sleep

To simulate the sleep phase, we inactive the sensory receptors (i.e. the input layer of network), cut off all sensory signals (i.e. remove all particles from the environment), and decouple output

$\text{Interleaved}_{T1,T2}$ and $\text{Interleaved}_{S,T1}$ training. As the network evolved along its trajectory in synaptic weight space, the distance from the current point in synaptic weight space, *pt*, to the two solution manifolds and their intersection were computed as follows:

$$d^n(p_t, M_\tau) = \min_{x \in M\tau}(d^n(p_t, x)).$$

Here, d^n is the n-dimensional Euclidean-distance function, where n is the dimensionality of synaptic weight space (i.e. $n = 6272$ here), M_τ is the point-set specific to the manifold or intersection in question (i.e. either M_{T1}, M_{T2}, or $M_{T1 \cap T2}$), and x is a particular element of the point-set M_τ.

Particle responsiveness metric (PRM)

The particle responsiveness metric (PRM) developed to quantify how responsive the network's weight matrix is to specific food particle orientations thereby allowing the quality of the receptive field for a given task to be determined was defined as follows:

$$\text{PRM}(\text{Particle Type}) = \sum_{\forall O \in Output} \text{grand}(\text{DirectionMask}(\text{O}) \odot \sum_{\forall H \in Hidden} W_{H \to O} * \sum_{\forall \text{P} \in \text{Particle}Masks} (W_H \odot P) * \text{grand}(W_H \odot P)^2)$$

Here *Output* is the set of all output layer neurons, *O; Hidden* is the set of all hidden layer neurons, *H; ParticleMasks* is the set of masks, *P*, representing all possible locations of a single instance of a *ParticleType* in the input field (e.g., horizontal bars would be a set of masks with a single horizontal bar placed in all possible locations in the visual field; each particle mask *P* consists of a *7 x 7* matrix of zeros with ones being placed in locations that correspond to current food pixels). W_H is a *7x7* synaptic weights matrix of a given hidden layer neuron *H;* $\odot$ gives Hadamard (or element-wise) product of two matrixes, *grand*(*A*) is a grand sum of all the elements of a matrix A (*grand*(*A*) = $e^T Ae$, where *e* is all-ones vector*). DirectionMask*(*O*) takes in an output layer neuron, *O*, and returns a matrix that represents the direction of motion with respect to the input field. For example, when the neuron that directs the critter to move up and to the left is supplied as input, the function returns a *7 x 7* matrix of zeros with the top left *3 x 3* submatrix being ones. $W_{H \to O}$ simply returns the synapse strength from the source *(H)* to destination *(O)* neuron.

Although this is seemingly an intricate metric, it captures many desired features of the network's connectivity and responses to food particles present in the visual field. Conceptually, this metric is similar to the method used for developing the receptive fields of output layer neurons with respect to the input field (Figs 2 and 5). PRM builds upon this qualitative visualization, allowing us to numerically assess how specific particles influence output layer neurons to spike when present in the portion of the visual field that corresponds to the direction of motion for that neuron. The intuitions of the metric are as follows: $W_H \odot P$ develops a notion of how well the current hidden neuron's (*H*) connections to the input layer overlaps with the current food particle (*P*) placed at specific location. The resulting matrix is then multiplied by $\text{grand}(W_H \odot P)^2$, which emphasizes contribution of the *H* neurons receiving input from adjacent pixels in correct orientation (i.e., sensitive to the food particles) vs those receiving input from random pixels. Indeed, when a hidden layer neuron *H* overlaps strongly with a food particle *P*, the chances of spiking are significantly increased, thus this nonlinear term captures the high impact overlapping receptive fields and food particles has on output layer activity. $W_{H \to O}$ captures how strongly the current output layer neuron *O* is listening to the current hidden layer neuron *H*.

layer activity from motor control (i.e. the output layer can spike but no longer causes the agent to move). We also change the learning rule between the hidden and output layer from rewarded to unsupervised STDP (see *Methods*: *Synaptic Plasticity* for details) as there is no way to evaluate decision-making without sensory input or motor output.

To simulate the spontaneous activity observed during REM sleep, we provided noise to each neuron in the hidden layer in a way which ensured that the spiking statistics of each neuron was conserved across awake and sleep phases. To determine these spiking rates, we recorded average spiking rates of neurons in the hidden layer H during preceding training of both Task 1 and Task 2; these task specific spiking rates were then averaged to generate target spiking rates for hidden layer neurons. $\text{Interleaved}_{S,T1}$ training consisted of alternating intervals of this sleep phase and training on Task 1, with each interval lasting 100 movement cycles (although no movement occurred).

Support vector machine training

A support vector machine with a radial basis function kernel was trained to classify synaptic weight configurations as being related to Task 1 or Task2. Labeled training data were obtained by taking the excitatory synaptic weight matrices between the hidden and output layers from the last fifth of the Task 1 and Task 2 training phases (i.e. after performance had appeared to asymptote). These synaptic weight matrices were then flattened into column vectors, and the column vectors were concatenated to form a training data matrix of size *number of features* x *number of samples*. The number of features was equal to the total number of excitatory synapses between the hidden and output layer– 6272 dimensions. We then used this support vector machine to classify held out synaptic weight configurations from Task 1 and Task 2 training, as well as ones which resulted from $\text{Interleaved}_{T1,T2}$ and $\text{Interleaved}_{S,T1}$ training.

2-D synaptic weight distributions (Fig 6)

First for each synapse we found how its synaptic strength changes between two slices in time, where the given synapse's strength at time slice 1 is the point's X-value and strength at time slice 2 is its Y-value. Then we binned this space and counted synapses in each bin to make two dimensional histograms where blue color corresponds to a single synapse found in a bin and brown corresponds to the max of 50 synapses. These two-dimensional histograms assist in visualizing the movement of all synapses between the two slices in time that are specified by the timelines at the top of each plot. Conceptually, it is important to note that if a synapse does not change in strength between time slice 1 and time slice 2, then point the synapse corresponds to in this space will lie on the diagonal of the plot since the X-value will match the Y-value. If a great change in the synapse's strength has occurred between time slice 1 and time slice 2, then the synapse's corresponding point will lie far from the diagonal since the X-value will be distant from the Y-value. The points on the X-(Y-) axis represent synapses that lost (gained) all synaptic strength between time slice 1 and time slice 2.

Distance from solution manifolds (Fig 7)

Each of the two solution manifolds (i.e. Task 1 and Task 2 specific manifolds) were defined by the point-sets in synaptic weight space which were capable of supporting robust performance on that particular task, namely the sets M_{T1} and M_{T2}. This included the synaptic weight states from the last fifth of training on a particular task(i.e. after performance on that task appeared to asymptote) and all of the synaptic weight states from the last fifth of both $\text{Interleaved}_{T1,T2}$ and $\text{Interleaved}_{S,T1/T2}$ training. The intersection of the two solution manifolds (i.e. the point-set $M_{T1 \cap T2}$) was defined solely by the synaptic weight states from the last fifth of both

These described pieces are multiplied together to form a weighted input receptive field of the output layer neuron with respect to a specific hidden layer neuron and food particle type / location. The sum of these terms for all hidden layer neurons and food particle locations is taken for a single output layer neuron, achieving a global view of all hidden layer neurons and food particle types / locations influencing the current output layer neuron. The *grand*(*A*) operation between the *DirectionMask*(*O*) and the previously described summed term is then taken to see how much the summed weighted receptive fields overlap with the corresponding direction of movement for output neuron *O*. This process is repeated for all output layer neurons to get a global quantification of how the current food particle influences activity in the direction of motion for all output layer neurons. When this metric is calculated for a given network state across food particle types we can observe what food particles impact output layer activity and drive the critter to move, highlighting what particle orientations the network is attracted to.

Supporting information

S1 Fig. Spike rasters showing network activity across various training regimes. (A-D) Representative spike rasters from various training regimes. The vertical axis specifies a unique neuron in the network while time in epochs is shown horizontally. Here a single dot represents a specific neuron spiking at a given time while the color of the dot dictates what layer that neuron belongs to (green, blue, red corresponding to input, hidden, and output layers respectively). Panels A, B, C, D correspond to sample activity from Task 1 training, Task 2 training, $I_{T1,T2}$ training and $I_{S,T1}$ training respectively. Note, in panel D activity is taken during a period of sleep when the hidden layer is spontaneously activated. Thus, there are hidden (blue) and output (red) layer spikes while the input (green) layer is completely silent.
(EPS)

S2 Fig. Model displays graceful degradation in performance as a result of hidden layer dropout. (A) Mean performance (red line) and standard deviation (blue lines) over time: unsupervised training (white), Task 1 training (blue), Task 1 testing (green). Hidden layer neurons are randomly removed during testing period. Gradient bar above Task 1 testing (green) displays the number of hidden layer neurons over time starting at 784 and decreasing down to 0. The testing performance remains high until ~25% of neurons are left, after which it starts to drop. This highlights the formation of a distributed synaptic structure between hidden and output layer neurons developed during training, ensuring output layer activity is not dictated by a select few hidden layer neurons. **(B)** Same as in (A) but for Task 2.
(EPS)

S3 Fig. Particle responsiveness metric (PRM) shows correspondence between type of training and particles preferred by the network. (A-D) Mean and standard deviation (blue bars and black lines respectively) of the PRM for various types of training and particle orientations across ten trials. The title of each plot reflects the most recently trained stage, the vertical axis corresponds to the value of the PRM while the horizontal axis identifies the particle type (bold labels indicate ideal particles the network would be attracted to following the corresponding training). It can be seen that the metric indicates the network is most responsive to the corresponding ideal particle types following a specific training regime e.g. Post Task 1 the network is most responsive to horizontal particles (A), Post Task 2 the network is most responsive to vertical particles (B), Post $I_{S,T1}$ the network is most responsive to horizontal and vertical particles (C), Post $I_{T1,T2}$ the network is most responsive to horizontal and vertical particles (D).
(EPS)

S4 Fig. Effect of sleep to protect old memory does not depend on specific properties of noise applied during sleep phase. **(A)** Mean performance (red line) and standard deviation (blue lines) over time: unsupervised training (white), Interleaved$_{S,T1}$ (grey), Task 1/2 testing (green/yellow). **(B)** Mean and standard deviation of performance during testing on Task 1 (blue) and Task 2 (red). Following Interleaved$_{S,T1}$, mean performance on Task 1 was 0.60 ± 0.03 while Task 2 was 0.49 ± 0.05. (In all experiments, 0.5 represents chance performance.) Note that periods of Task 1 training interleaved with sleep do not lead to increase in performance on untrained Task 2, even when Task 2 data from another experiment were used to set up mean firing rates of the random input during sleep. **(C)** Same as in (A) but the sequence of training was: unsupervised training (white), Task 1 training (blue), Task 1/2 testing (green/yellow), Interleaved$_{S,T1}$ (grey), Task 1/2 testing (green/yellow). **(D)** Mean and standard deviation of performance during testing on Task 1 (blue) and Task 2 (red) after Task 1 training and after Interleaved$_{S,T1}$. Following Task 1 training, mean performance on Task 1 was 0.70 ± 0.02 while Task 2 was 0.53 ± 0.02. Post Interleaved$_{S,T1}$ training, mean performance on Task 1 was 0.71 ± 0.02 and Task 2 was 0.51 ± 0.02. Task 1 performance remained high after Interleaved$_{S,T1}$ but no improvement on Task 2 was observed. **(E)** Mean performance (red line) and standard deviation (blue lines) over time: unsupervised training (white), Task 1 training (blue), Task 1/2 testing (green/yellow), Interleaved$_{US,T2}$ (burnt orange), Task 1/2 testing (green/yellow). **(F)** Mean and standard deviation of performance during testing on Task 1 (blue) and Task 2 (red). Following Task 1 training, mean performance on Task 1 was 0.70 ± 0.02 while Task 2 was 0.53 ± 0.02. Post Interleaved$_{US,T2}$ training, mean performance on Task 1 was 0.67 ± 0.05 and Task 2 was 0.69 ± 0.03.
(EPS)

S5 Fig. Interleaving old and new task training allows integrating synaptic information relevant to new task while preserving old task information. **(A)** Mean performance (red line) and standard deviation (blue lines) over time: unsupervised training (white), Task 1 training (blue), Task 1/2 testing (green/yellow), Task 2 training (red), Task 1/2 testing (green/yellow), Interleaved$_{T1,T2}$ training (purple), Task 1/2 testing (green/yellow). **(B)** Mean and standard deviation of performance during testing on Task 1 (blue) and Task 2 (red). Following Task 1training, mean performance on Task 1 was 0.69 ± 0.02 while Task 2 was 0.53 ± 0.02. Conversely, following Task 2 training, mean performance on Task 1 was 0.52 ± 0.02 while Task2 was 0.69 ± 0.04. Following Interleaved$_{T1,T2}$ training, mean performance on Task 1 was 0.65 ±0.03 while Task 2 was 0.67 ± 0.04. **(C)** Distributions of task-relevant synaptic weights (blue bars–single trial, orange line / shaded region–mean / std across 10 trails. The distributional structure of Task 1-relevant synapses following Task 1 training (top-left) is destroyed following Task 2 training (top-middle), but partially recovered following. Interleaved$_{T1,T2}$ training (top-right). Similarly, the distributional structure of Task 2-relevantsynapses following Task 2 training (bottom-middle), which was not present following Task 1training (bottom-left), was partially preserved following Interleaved$_{T1,T2}$ training (bottom-right).**(D)** Box plots with mean (dashed green line) and median (dashed orange line) of the distance to the decision boundary found by an SVM trained to classify Task 1 and Task 2 synaptic weight matrices for Task 1, Task 2, and Interleaved$_{T1,T2}$ training across trials. Task 1 and Task 2synaptic weight matrices had mean classification values of -0.069 and 0.069 respectively, while that of Interleaved$_{T1,T2}$ training was 0.016. **(E)** Trajectory of H to O layer synaptic weights through PC space. Synaptic weights which evolved during Interleaved$_{T1,T2}$ training (green dots)clustered in a location of PC space intermediary between the clusters of synaptic weights which evolved during training on Task 1 (red dots) and Task 2 (blue dots).
(EPS)

S6 Fig. Freezing a fraction of task specific strong synapses preserves differing degrees of performance in a sequential learning paradigm. (A-C) Mean and standard deviation of performance during testing on Task 1 (blue) and Task 2 (red). Left, Performance after Task 1 training. Right, Performance after Task 2 training when a fraction of the strongest (after Task 1 training) synapses remained frozen– 1% (A), 5% (B), 10% (C). In all cases, after Task 1 training, Task 1 performance was 0.70 ± 0.02 and Task 2 performance was 0.53 ± 0.02. (A) Freezing the top 1% of Task 1 synapses resulted in a Task 1 performance of 0.54 ± 0.02 and Task 2 performance of 0.68 ± 0.03. (B) Freezing the top 5% of Task 1 synapses resulted in a Task 1 performance of 0.65 ± 0.02 and Task 2 performance of 0.61 ± 0.01. (C) Freezing the top 10% of Task 1 synapses resulted in a Task 1 performance of 0.70 ± 0.03 and Task 2 performance of 0.53 ± 0.03. Freezing the top 1% of Task 1 synapses was not sufficient to maintain Task 1 performance, thus enabling Task 2 relevant synapses to dominate the network; however, freezing the top 10% of Task 1 synapses fully retains Task 1 performance preventing Task 2 to be learned.
(EPS)

Author Contributions

Conceptualization: Ryan Golden, Pavel Sanda, Maxim Bazhenov.

Data curation: Jean Erik Delanois.

Formal analysis: Ryan Golden, Jean Erik Delanois, Maxim Bazhenov.

Funding acquisition: Maxim Bazhenov.

Investigation: Jean Erik Delanois.

Methodology: Ryan Golden, Jean Erik Delanois, Pavel Sanda, Maxim Bazhenov.

Project administration: Maxim Bazhenov.

Resources: Maxim Bazhenov.

Software: Jean Erik Delanois.

Supervision: Pavel Sanda, Maxim Bazhenov.

Visualization: Jean Erik Delanois.

Writing – original draft: Ryan Golden, Jean Erik Delanois, Pavel Sanda, Maxim Bazhenov.

Writing – review & editing: Ryan Golden, Jean Erik Delanois, Pavel Sanda, Maxim Bazhenov.

References

1. French RM. Catastrophic forgetting in connectionist networks. Trends Cogn Sci. 1999; 3(4):128–35. https://doi.org/10.1016/s1364-6613(99)01294-2 PMID: 10322466
2. Mccloskey M, Cohen NJ. CATASTROPHIC INTERFERENCE IN CONNECTIONIST NETWORKS: THE SEQUENTIAL LEARNING PROBLEM. The Psychology of Learning and Motivation. 1989; 24:109–65.
3. Ratcliff R. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. Psychol Rev. 1990; 97(2):285–308. https://doi.org/10.1037/0033-295x.97.2.285 PMID: 2186426
4. Hasselmo ME. Avoiding Catastrophic Forgetting. Trends Cogn Sci. 2017; 21(6):407–8. https://doi.org/10.1016/j.tics.2017.04.001 PMID: 28442279
5. Hassabis D, Kumaran D, Summerfield C, Botvinick M. Neuroscience-Inspired Artificial Intelligence. Neuron. 2017; 95(2):245–58. https://doi.org/10.1016/j.neuron.2017.06.011 PMID: 28728020

27. Cassenaer S, Laurent G. Hebbian STDP in mushroom bodies facilitates the synchronous flow of olfactory information in locusts. Nature. 2007; 448(7154):709–13. https://doi.org/10.1038/nature05973 PMID: 17581587

28. Bi GQ, Poo MM. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. J Neurosci. 1998; 18(24):10464–72. https://doi.org/10.1523/JNEUROSCI.18-24-10464.1998 PMID: 9852584

29. Markram H, Lubke J, Frotscher M, Sakmann B. Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. Science. 1997; 275(5297):213–5. https://doi.org/10.1126/science.275.5297.213 PMID: 8985014

30. Farries MA, Fairhall AL. Reinforcement learning with modulated spike timing dependent synaptic plasticity. J Neurophysiol. 2007; 98(6):3648–65. https://doi.org/10.1152/jn.00364.2007 PMID: 17928565

31. Florian RV. Reinforcement learning through modulation of spike-timing-dependent synaptic plasticity. Neural Comput. 2007; 19(6):1468–502. https://doi.org/10.1162/neco.2007.19.6.1468 PMID: 17444757

32. Izhikevich EM. Solving the distal reward problem through linkage of STDP and dopamine signaling. Cereb Cortex. 2007; 17(10):2443–52. https://doi.org/10.1093/cercor/bhl152 PMID: 17220510

33. Legenstein R, Pecevski D, Maass W. A learning theory for reward-modulated spike-timing-dependent plasticity with application to biofeedback. PLoS Comput Biol. 2008; 4(10):e1000180. https://doi.org/10.1371/journal.pcbi.1000180 PMID: 18846203

34. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015; 521(7553):436–44. https://doi.org/10.1038/nature14539 PMID: 26017442

35. Cadieu CF, Hong H, Yamins DL, Pinto N, Ardila D, Solomon EA, et al. Deep neural networks rival the representation of primate IT cortex for core visual object recognition. PLoS Comput Biol. 2014; 10(12):e1003963. https://doi.org/10.1371/journal.pcbi.1003963 PMID: 25521294

36. Yamins DL, DiCarlo JJ. Using goal-driven deep learning models to understand sensory cortex. Nat Neurosci. 2016; 19(3):356–65. https://doi.org/10.1038/nn.4244 PMID: 26906502

37. Yamins DL, Hong H, Cadieu CF, Solomon EA, Seibert D, DiCarlo JJ. Performance-optimized hierarchical models predict neural responses in higher visual cortex. Proc Natl Acad Sci U S A. 2014; 111(23):8619–24. https://doi.org/10.1073/pnas.1403112111 PMID: 24812127

38. Wei Y, Krishnan G, Bazhenov M. Synaptic Mechanisms of Memory Consolidation during Sleep Slow Oscillations. Journal of Neuroscience. 2016; 36(15):4231–47. https://doi.org/10.1523/JNEUROSCI.3648-15.2016 PMID: 27076422

39. Wei Y, Krishnan GP, Komarov M, Bazhenov M. Differential roles of sleep spindles and sleep slow oscillations in memory consolidation. PLoS Comput Biol. 2018; 14(7):e1006322. https://doi.org/10.1371/journal.pcbi.1006322 PMID: 29985966

40. Wei Y, Krishnan GP, Marshall L, Martinetz T, Bazhenov M. Stimulation Augments Spike Sequence Replay and Memory Consolidation during Slow-Wave Sleep. The Journal of neuroscience: the official journal of the Society for Neuroscience. 2020; 40(4):811–24. https://doi.org/10.1523/JNEUROSCI.1427-19.2019 PMID: 31792151

41. Gonzalez OC, Sokolov Y, Krishnan GP, Delanois JE, Bazhenov M. Can sleep protect memories from catastrophic forgetting? Elife. 2020;9. https://doi.org/10.7554/eLife.51005 PMID: 32748786

42. Peever J, Fuller PM. The Biology of REM Sleep. Curr Biol. 2017; 27(22):R1237–R48. https://doi.org/10.1016/j.cub.2017.10.026 PMID: 29161567

43. Oudiette D, Antony JW, Creery JD, Paller KA. The role of memory reactivation during wakefulness and sleep in determining which memories endure. J Neurosci. 2013; 33(15):6672–8. https://doi.org/10.1523/JNEUROSCI.5497-12.2013 PMID: 23575863

44. Paller KA, Voss JL. Memory reactivation and consolidation during sleep. Learn Mem. 2004; 11(6):664–70. https://doi.org/10.1101/lm.75704 PMID: 15576883

45. Walker MP, Stickgold R. Sleep-dependent learning and memory consolidation. Neuron. 2004; 44(1):121–33. https://doi.org/10.1016/j.neuron.2004.08.031 PMID: 15450165

46. Stickgold R, James L, Hobson JA. Visual discrimination learning requires sleep after training. Nat Neurosci. 2000; 3(12):1237–8. https://doi.org/10.1038/81756 PMID: 11100141

47. Hennevin E, Hars B, Maho C, Bloch V. Processing of learned information in paradoxical sleep: relevance for memory. Behav Brain Res. 1995; 69(1–2):125–35. https://doi.org/10.1016/0166-4328(95)00013-j PMID: 7546303

48. Lewis PA, Knoblich G, Poe G. How Memory Replay in Sleep Boosts Creative Problem-Solving. Trends Cogn Sci. 2018; 22(6):491–503. https://doi.org/10.1016/j.tics.2018.03.009 PMID: 29776467

49. Oudiette D, Paller KA. Upgrading the sleeping brain with targeted memory reactivation. Trends Cogn Sci. 2013; 17(3):142–9. https://doi.org/10.1016/j.tics.2013.01.006 PMID: 23433937

6. Kemker R, Abitino A, McClure M, Kanan C. Measuring Catastrophic Forgetting in Neural Networks. arXiv:170802072 [Internet]. 2017.
7. Kirkpatrick J, Pascanu R, Rabinowitz N, Veness J, Desjardins G, Rusu AA, et al. Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences. 2017; 114 (13):3521–6. https://doi.org/10.1073/pnas.1611835114 PMID: 28292907
8. Kemker R, Kanan C. Fearnet: Brain-inspired model for incremental learning. arXiv:171110563. 2017.
9. Hayes TL, Krishnan GP, Bazhenov M, Siegelmann HT, Sejnowski TJ, Kanan C. Replay in Deep Learning: Current Approaches and Missing Biological Elements. Neural computation. 2021; 33(11):2908–50. https://doi.org/10.1162/neco_a_01433 PMID: 34474476
10. Flesch T, Balaguer J, Dekker R, Nili H, Summerfield C. Comparing continual task learning in minds and machines. Proc Natl Acad Sci U S A. 2018; 115(44):E10313–E22. https://doi.org/10.1073/pnas.1800755115 PMID: 30322916
11. McClelland JL, McNaughton BL, O'Reilly RC. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. Psychol Rev. 1995; 102(3):419–57. https://doi.org/10.1037/0033-295X.102.3.419 PMID: 7624455
12. Evans BD, Stringer SM. Transformation-invariant visual representations in self-organizing spiking neural networks. Front Comput Neurosci. 2012; 6:46. https://doi.org/10.3389/fncom.2012.00046 PMID: 22848199
13. Higgins I, Stringer S, Schnupp J. Unsupervised learning of temporal features for word categorization in a spiking neural network model of the auditory brain. PLoS One. 2017; 12(8):e0180174. https://doi.org/10.1371/journal.pone.0180174 PMID: 28797034
14. Sanda P, Skorheim S, Bazhenov M. Multi-layer network utilizing rewarded spike time dependent plasticity to learn a foraging task. PLoS Comput Biol. 2017; 13(9):e1005705. https://doi.org/10.1371/journal.pcbi.1005705 PMID: 28961245
15. Skorheim S, Lonjers P, Bazhenov M. A spiking network model of decision making employing rewarded STDP. PLoS One. 2014; 9(3):e90821. https://doi.org/10.1371/journal.pone.0090821 PMID: 24632858
16. Rasch B, Born J. About sleep's role in memory. Physiological reviews. 2013; 93(2):681–766. https://doi.org/10.1152/physrev.00032.2012 PMID: 23589831
17. Ji D, Wilson MA. Coordinated memory replay in the visual cortex and hippocampus during sleep. Nat Neurosci. 2007; 10(1):100–7. https://doi.org/10.1038/nn1825 PMID: 17173043
18. Euston DR, Tatsuno M, McNaughton BL. Fast-forward playback of recent memory sequences in prefrontal cortex during sleep. Science. 2007; 318(5853):1147–50. https://doi.org/10.1126/science.1148979 PMID: 18006749
19. Peyrache A, Khamassi M, Benchenane K, Wiener SI, Battaglia FP. Replay of rule-learning related neural patterns in the prefrontal cortex during sleep. Nat Neurosci. 2009; 12(7):919–26. https://doi.org/10.1038/nn.2337 PMID: 19483687
20. Barnes DC, Wilson DA. Slow-wave sleep-imposed replay modulates both strength and precision of memory. J Neurosci. 2014; 34(15):5134–42. https://doi.org/10.1523/JNEUROSCI.5274-13.2014 PMID: 24719093
21. Mednick SC, Cai DJ, Shuman T, Anagnostaras S, Wixted JT. An opportunistic theory of cellular and systems consolidation. Trends Neurosci. 2011; 34(10):504–14. https://doi.org/10.1016/j.tins.2011.06.003 PMID: 21742389
22. Stickgold R. Parsing the role of sleep in memory processing. Curr Opin Neurobiol. 2013; 23(5):847–53. https://doi.org/10.1016/j.conb.2013.04.002 PMID: 23618558
23. Ramanathan DS, Gulati T, Ganguly K. Sleep-Dependent Reactivation of Ensembles in Motor Cortex Promotes Skill Consolidation. PLOS Biology. 2015; 13(9):e1002263. https://doi.org/10.1371/journal.pbio.1002263 PMID: 26382320
24. Zwaka H, Bartels R, Gora J, Franck V, Culo A, Gotsch M, et al. Context odor presentation during sleep enhances memory in honeybees. Curr Biol. 2015; 25(21):2869–74. https://doi.org/10.1016/j.cub.2015.09.069 PMID: 26592345
25. Melnattur K, Kirszenblat L, Morgan E, Militchin V, Sakran B, English D, et al. A conserved role for sleep in supporting Spatial Learning in Drosophila. Sleep. 2021; 44(3). https://doi.org/10.1093/sleep/zsaa197 PMID: 32959053
26. Donlea JM, Thimgan MS, Suzuki Y, Gottschalk L, Shaw PJ. Inducing sleep by remote control facilitates memory consolidation in Drosophila. Science. 2011; 332(6037):1571–6. https://doi.org/10.1126/science.1202249 PMID: 21700877

50. McDevitt EA, Duggan KA, Mednick SC. REM sleep rescues learning from interference. Neurobiol Learn Mem. 2015; 122:51–62. https://doi.org/10.1016/j.nlm.2014.11.015 PMID: 25498222

51. Javed K, White M. Meta-Learning Representations for Continual Learning. arXiv e-prints [Internet]. 2019 May 01, 2019:[arXiv:1905.12588 p.]. Available from: https://ui.adsabs.harvard.edu/abs/2019arXiv190512588J.

52. Roumis DK, Frank LM. Hippocampal sharp-wave ripples in waking and sleeping states. Curr Opin Neurobiol. 2015; 35:6–12. https://doi.org/10.1016/j.conb.2015.05.001 PMID: 26011627

53. Swanson RA, Levenstein D, McClain K, Tingley D, Buzsáki G. Variable specificity of memory trace reactivation during hippocampal sharp wave ripples. Current Opinion in Behavioral Sciences. 2020; 32:126–35. https://doi.org/10.1016/j.cobeha.2020.02.008 PMID: 36034494

54. Krishnan GP, Tadros T, Ramyaa R, Bazhenov M. Biologically inspired sleep algorithm for artificial neural networks. arXiv. 2019:1908.02240v1.

55. Tadros T, Krishnan G, Ramyaa R, Bazhenov M. Biologically inspired sleep algorithm for reducing catastrophic forgetting in neural networks. AAAI Conference on Artificial Intelligence 2020. p. 13933–4.

56. Tadros T, Krishnan GP, Ramyaa R, Bazhenov M. Biologically inspired sleep algorithm for increased generalization and adversarial robustness in deep neural networks. International Conference on Learning Representations [Internet]. 2019.

57. Laurent G. Olfactory network dynamics and the coding of multidimensional signals. Nat Rev Neurosci. 2002; 3(11):884–95. https://doi.org/10.1038/nrn964 PMID: 12415296

58. Assisi C, Stopfer M, Laurent G, Bazhenov M. Adaptive regulation of sparseness by feedforward inhibition. Nature neuroscience. 2007; 10(9):1176–84. https://doi.org/10.1038/nn1947 PMID: 17660812

59. Perez-Orive J, Bazhenov M, Laurent G. Intrinsic and circuit properties favor coincidence detection for decoding oscillatory input. The Journal of neuroscience: the official journal of the Society for Neuroscience. 2004; 24(26):6037–47.

60. Schultz W. Dopamine reward prediction-error signalling: a two-component response. Nat Rev Neurosci. 2016; 17(3):183–95. https://doi.org/10.1038/nrn.2015.26 PMID: 26865020

61. Schultz W. Dopamine reward prediction error coding. Dialogues Clin Neurosci. 2016; 18(1):23–32. https://doi.org/10.31887/DCNS.2016.18.1/wschultz PMID: 27069377

62. Schultz W, Dayan P, Montague PR. A neural substrate of prediction and reward. Science. 1997; 275 (5306):1593–9. https://doi.org/10.1126/science.275.5306.1593 PMID: 9054347

63. Cassenaer S, Laurent G. Conditional modulation of spike-timing-dependent plasticity for olfactory learning. Nature. 2012; 482(7383):47–52. https://doi.org/10.1038/nature10776 PMID: 22278062

64. Tainton-Heap LAL, Kirszenblat LC, Notaras ET, Grabowska MJ, Jeans R, Feng K, et al. A Paradoxical Kind of Sleep in Drosophila melanogaster. Curr Biol. 2021; 31(3):578–90 e6. https://doi.org/10.1016/j.cub.2020.10.081 PMID: 33238155

65. Kaiser W, Steiner-Kaiser J. Neuronal correlates of sleep, wakefulness and arousal in a diurnal insect. Nature. 1983; 301(5902):707–9. https://doi.org/10.1038/301707a0 PMID: 6828153

66. Sauer S, Kinkelin M, Herrmann E, Kaiser W. The dynamics of sleep-like behaviour in honey bees. Journal of comparative physiology A, Neuroethology, sensory, neural, and behavioral physiology. 2003; 189 (8):599–607. https://doi.org/10.1007/s00359-003-0436-9 PMID: 12861424

67. Rulkov NF, Bazhenov M. Oscillations and synchrony in large-scale cortical network models. J Biol Phys. 2008; 34(3–4):279–99. https://doi.org/10.1007/s10867-008-9079-y PMID: 19669478

68. Rulkov NF, Timofeev I, Bazhenov M. Oscillations in large-scale cortical networks: map-based model. J Comput Neurosci. 2004; 17(2):203–23. https://doi.org/10.1023/B:JCNS.0000037683.55688.7e PMID: 15306740

69. Bazhenov M, Stopfer M. Forward and back: motifs of inhibition in olfactory processing. Neuron. 2010; 67(3):357–8. https://doi.org/10.1016/j.neuron.2010.07.023 PMID: 20696373

70. Bruno RM. Synchrony in sensation. Curr Opin Neurobiol. 2011; 21(5):701–8. https://doi.org/10.1016/j.conb.2011.06.003 PMID: 21723114

71. Dong H, Shao Z, Nerbonne JM, Burkhalter A. Differential depression of inhibitory synaptic responses in feedforward and feedback circuits between different areas of mouse visual cortex. J Comp Neurol. 2004; 475(3):361–73. https://doi.org/10.1002/cne.20164 PMID: 15221951

72. Pouille F, Scanziani M. Enforcement of temporal fidelity in pyramidal cells by somatic feed-forward inhibition. Science. 2001; 293(5532):1159–63. https://doi.org/10.1126/science.1060342 PMID: 11498596

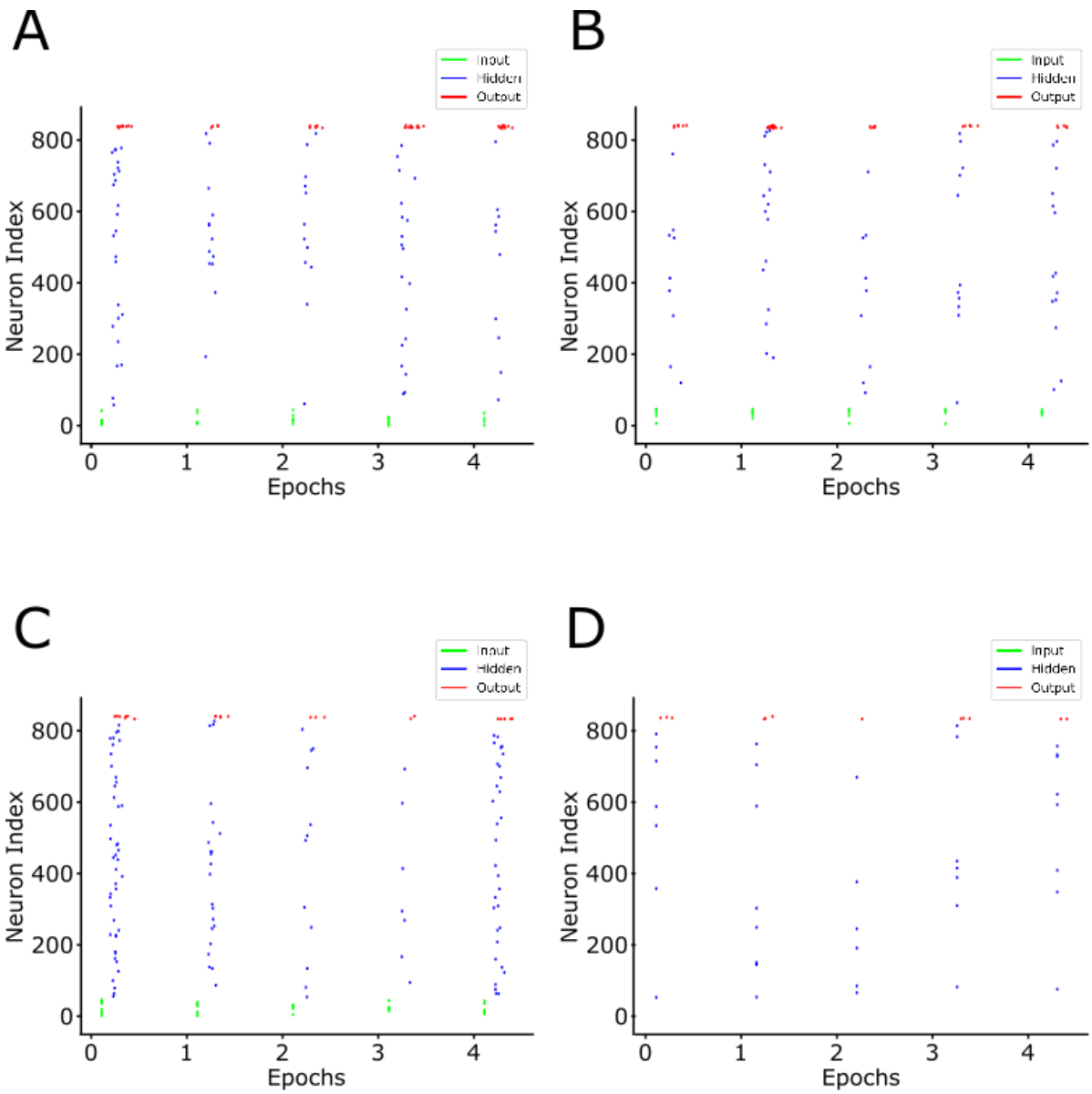
73. Shao Z, Burkhalter A. Different balance of excitation and inhibition in forward and feedback circuits of rat visual cortex. Journal of Neuroscience. 1996; 16(22):7353–65. https://doi.org/10.1523/JNEUROSCI.16-22-07353.1996 PMID: 8929442

74. Silberberg G. Polysynaptic subcircuits in the neocortex: spatial and temporal diversity. Curr Opin Neurobiol. 2008; 18(3):332–7. https://doi.org/10.1016/j.conb.2008.08.009 PMID: 18801433

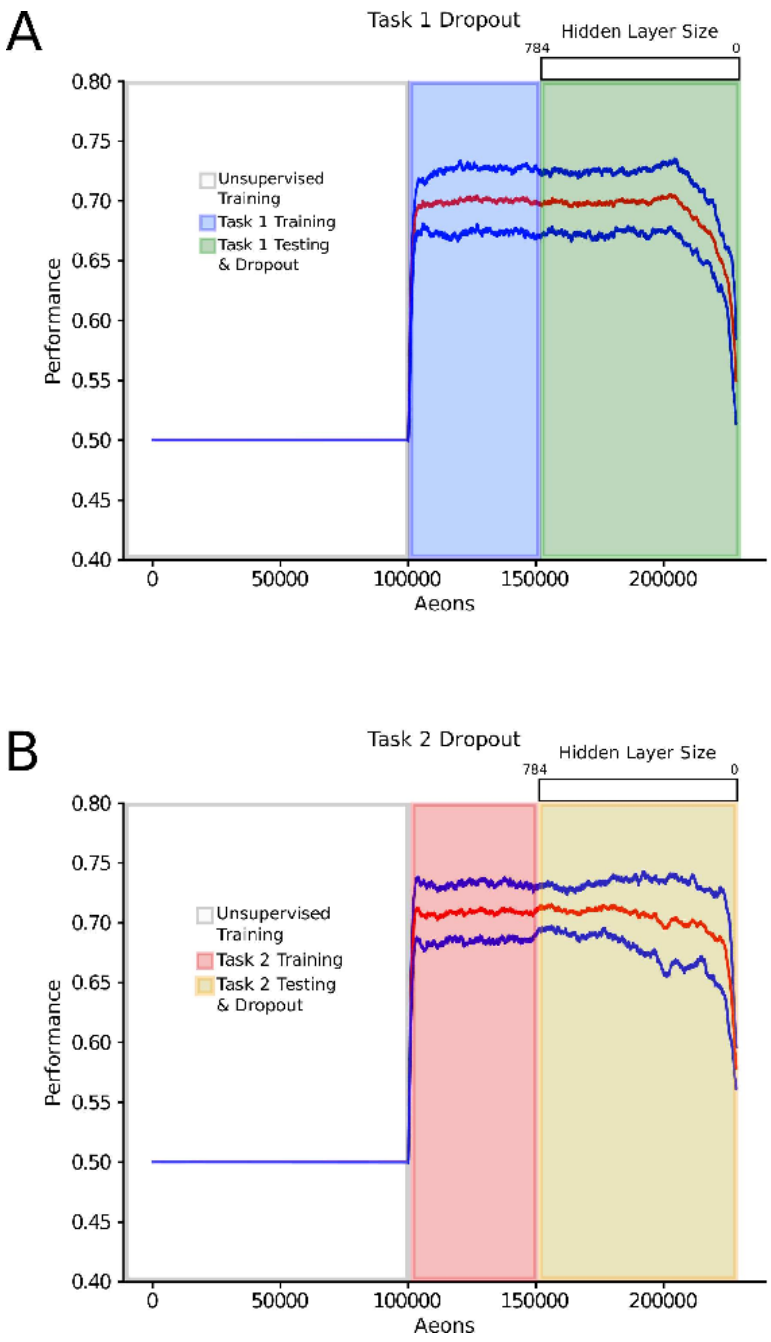
75. Bazhenov M, Rulkov NF, Fellous JM, Timofeev I. Role of network dynamics in shaping spike timing reliability. Phys Rev E Stat Nonlin Soft Matter Phys. 2005; 72(4 Pt 1):041903. https://doi.org/10.1103/PhysRevE.72.041903 PMID: 16383416

76. Rulkov NF. Modeling of spiking-bursting neural behavior using two-dimensional map. Phys Rev E Stat Nonlin Soft Matter Phys. 2002; 65(4 Pt 1):041922. https://doi.org/10.1103/PhysRevE.65.041922 PMID: 12005888

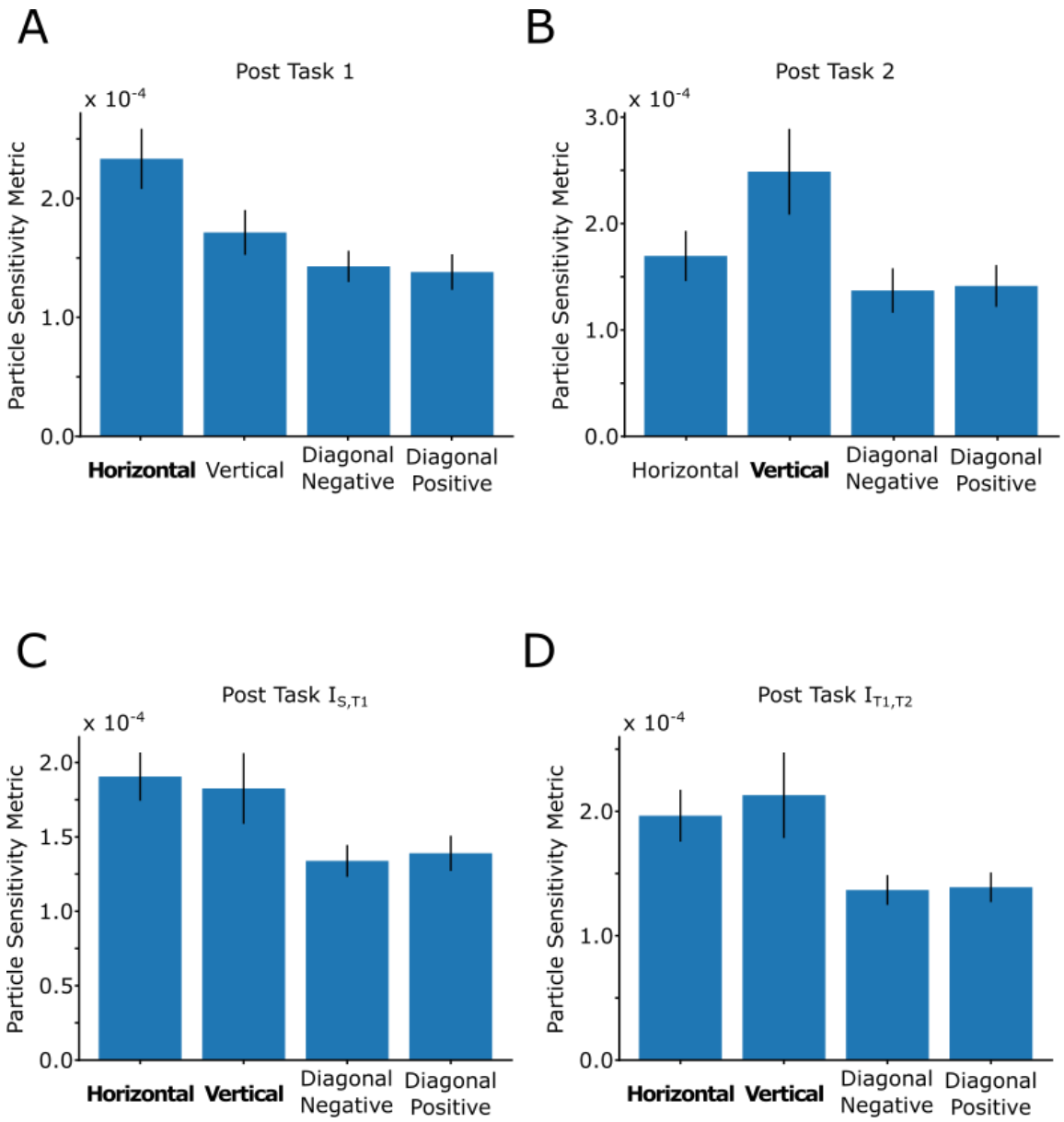
77. Komarov M, Krishnan G, Chauvette S, Rulkov N, Timofeev I, Bazhenov M. New class of reduced computationally efficient neuronal models for large-scale simulations of brain dynamics. J Comput Neurosci. 2018; 44(1):1–24. https://doi.org/10.1007/s10827-017-0663-7 PMID: 29230640



S1 Figure. Spike rasters showing network activity across various training regimes. (A-D) Representative spike rasters from various training regimes. The vertical axis specifies a unique neuron in the network while time in epochs is shown horizontally. Here a single dot represents a specific neuron spiking at a given time while the color of the dot dictates what layer that neuron belongs to (green, blue, red corresponding to input, hidden, and output layers respectively). Panels A, B, C, D correspond to sample activity from Task 1 training, Task 2 training, IT1,T2 training and IS,T1 training respectively. Note, in panel D activity is taken during a period of sleep when the hidden layer is spontaneously activated. Thus, there are hidden (blue) and output (red) layer spikes while the input (green) layer is completely silent.

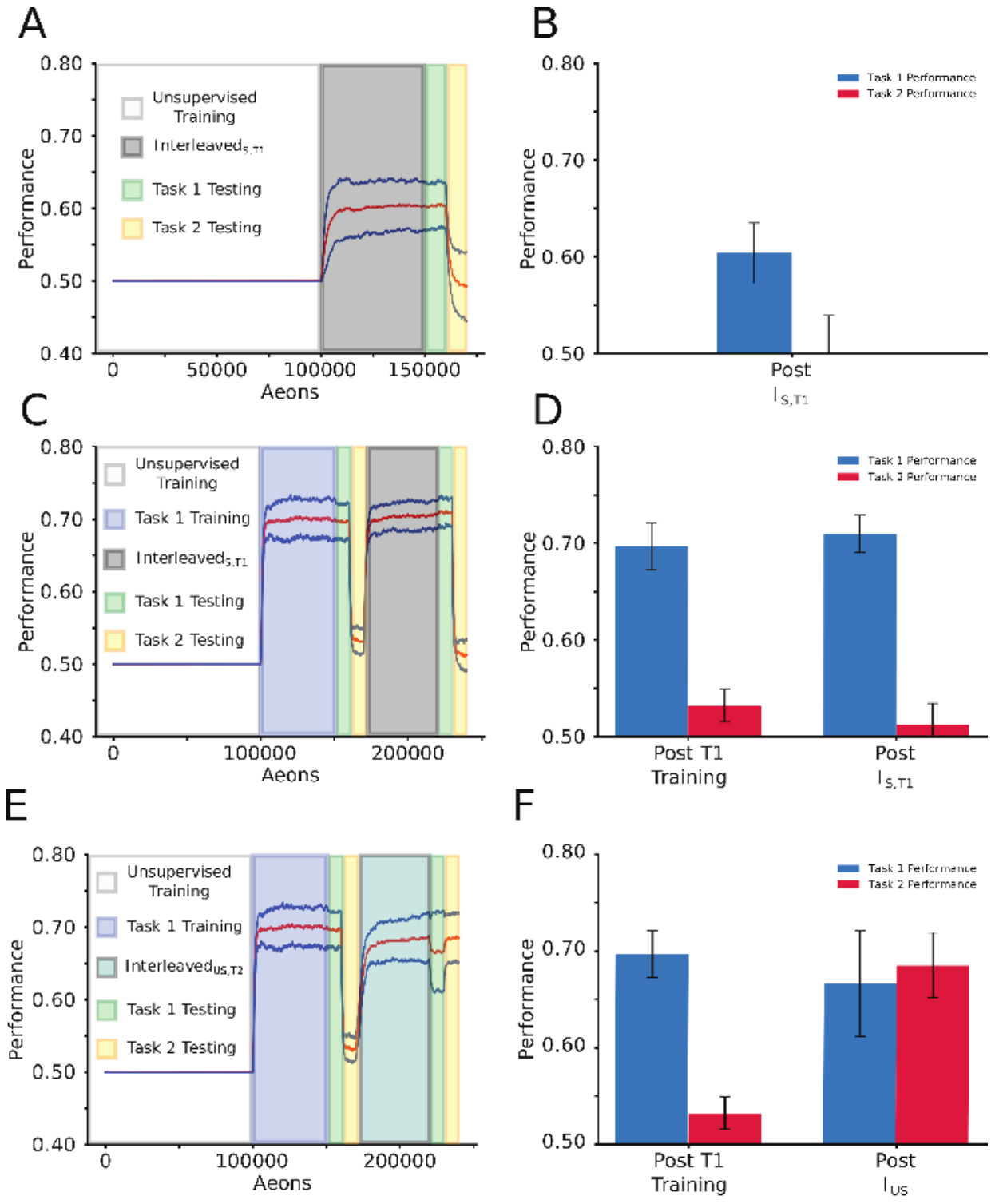


S2 Figure. Model displays graceful degradation in performance as a result of hidden layer dropout. (A) Mean performance (red line) and standard deviation (blue lines) over time: unsupervised training (white), Task 1 training (blue), Task 1 testing (green). Hidden layer neurons are randomly removed during testing period. Gradient bar above Task 1 testing (green) displays the number of hidden layer neurons over time starting at 784 and decreasing down to 0. The testing performance remains high until ~25% of neurons are left, after which it starts to drop. This highlights the formation of a distributed synaptic structure between hidden and output layer neurons developed during training, ensuring output layer activity is not dictated by a select few hidden layer neurons. (B) Same as in (A) but for Task 2.

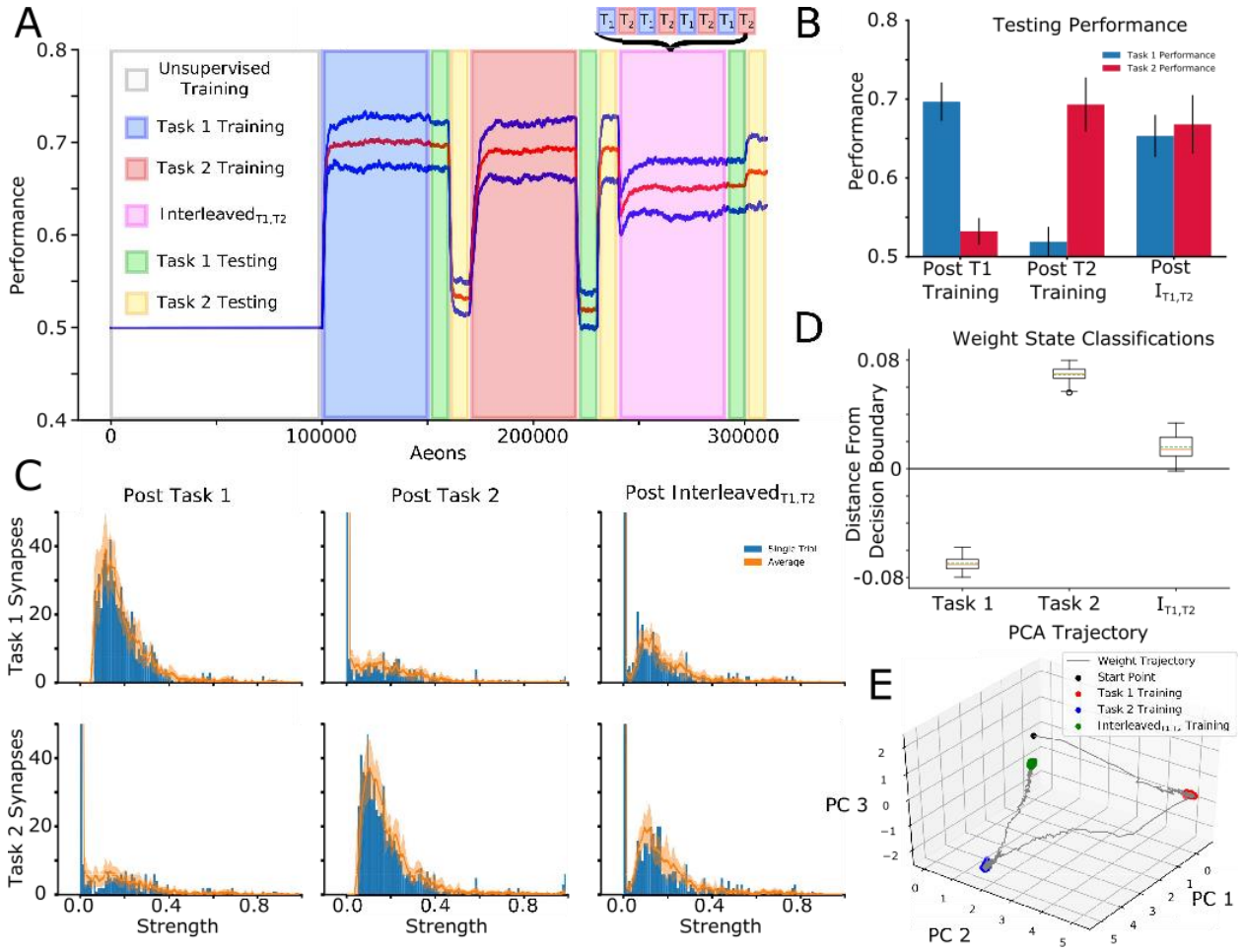


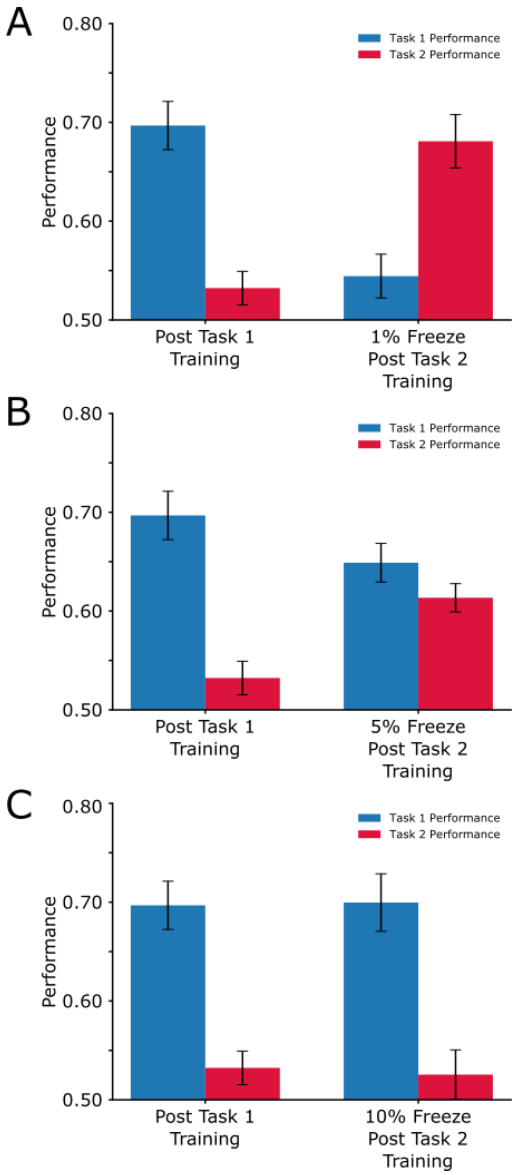
S3 Figure. Particle responsiveness metric (PRM) shows correspondence between type of training and particles preferred by the network. (A-D) Mean and standard deviation (blue bars and black lines respectively) of the PRM for various types of training and particle orientations across ten trials. The title of each plot reflects the most recently trained stage, the vertical axis corresponds to the value of the PRM while the horizontal axis identifies the particle type (bold labels indicate ideal particles the network would be attracted to following the corresponding training). It can be seen that the metric indicates the network is most responsive to the corresponding ideal particle types following a specific training regime e.g. Post Task 1 the network is most responsive to horizontal particles (A), Post Task 2 the network is most responsive to vertical particles (B), Post IS,T1 the network is most responsive to horizontal and vertical particles (C), Post IT1,T2 the network is most responsive to horizontal and vertical particles (D).

S4 Figure. Effect of sleep to protect old memory does not depend on specific properties of noise applied during sleep phase. (A) Mean performance (red line) and standard deviation (blue lines) over time: unsupervised training (white), InterleavedS,T1 (grey), Task 1/2 testing (green/yellow). (B) Mean and standard deviation of performance during testing on Task 1 (blue) and Task 2 (red). Following InterleavedS,T1, mean performance on Task 1 was 0.60 ± 0.03 while Task 2 was 0.49 ± 0.05. (In all experiments, 0.5 represents chance performance.) Note that periods of Task 1 training interleaved with sleep do not lead to increase in performance on untrained Task 2, even when Task 2 data from another experiment were used to set up mean firing rates of the random input during sleep. (C) Same as in (A) but the sequence of training was: unsupervised training (white), Task 1 training (blue), Task 1/2 testing (green/yellow), InterleavedS,T1 (grey), Task 1/2 testing (green/yellow). (D) Mean and standard deviation of performance during testing on Task 1 (blue) and Task 2 (red) after Task 1 training and after InterleavedS,T1. Following Task 1 training, mean performance on Task 1 was 0.70 ± 0.02 while Task 2 was 0.53 ± 0.02. Post InterleavedS,T1 training, mean performance on Task 1 was 0.71 ± 0.02 and Task 2 was 0.51 ± 0.02. Task 1 performance remained high after InterleavedS,T1 but no improvement on Task 2 was observed. (E) Mean performance (red line) and standard deviation (blue lines) over time: unsupervised training (white), Task 1 training (blue), Task 1/2 testing (green/yellow), InterleavedUS,T2 (burnt orange), Task 1/2 testing (green/yellow). (F) Mean and standard deviation of performance during testing on Task 1 (blue) and Task 2 (red). Following Task 1 training, mean performance on Task 1 was 0.70 ± 0.02 while Task 2 was 0.53 ± 0.02. Post InterleavedUS,T2 training, mean performance on Task 1 was 0.67 ± 0.05 and Task 2 was 0.69 ± 0.03.


A
Performance
Aeons
Unsupervised Training
Interleaved$_{S,T1}$
Task 1 Testing
Task 2 Testing
B
Task 1 Performance
Task 2 Performance
Post
$I_{S,T1}$
C
Unsupervised Training
Task 1 Training
Interleaved$_{S,T1}$
Task 1 Testing
Task 2 Testing
D
Post T1 Training
Post
$I_{S,T1}$
E
Unsupervised Training
Task 1 Training
Interleaved$_{US,T2}$
Task 1 Testing
Task 2 Testing
F
Post T1 Training
Post
I_{US}

S5 Figure. Interleaving old and new task training allows integrating synaptic information relevant to new task while preserving old task information. (A) Mean performance (red line) and standard deviation (blue lines) over time: unsupervised training (white), Task 1 training(blue), Task 1/2 testing (green/yellow), Task 2 training (red), Task 1/2 testing (green/yellow), InterleavedT1,T2 training (purple), Task 1/2 testing (green/yellow). (B) Mean and standard deviation of performance during testing on Task 1 (blue) and Task 2 (red). Following Task 1training, mean performance on Task 1 was 0.69 ± 0.02 while Task 2 was 0.53 ± 0.02. Conversely, following Task 2 training, mean performance on Task 1 was 0.52 ± 0.02 while Task2 was 0.69 ± 0.04. Following InterleavedT1,T2 training, mean performance on Task 1 was 0.65 ±0.03 while Task 2 was 0.67 ± 0.04. (C) Distributions of task-relevant synaptic weights (blue bars–single trial, orange line / shaded region–mean / std across 10 trails. The distributional structure of Task 1-relevant synapses following Task 1 training (top-left) is destroyed following Task 2 training (top-middle), but partially recovered following. InterleavedT1,T2 training (top-right). Similarly, the distributional structure of Task 2-relevantsynapses following Task 2 training (bottom-middle), which was not present following Task 1training (bottom-left), was partially preserved following InterleavedT1,T2 training (bottom-right).(D) Box plots with mean (dashed green line) and median (dashed orange line) of the distance to the decision boundary found by an SVM trained to classify Task 1 and Task 2 synaptic weight matrices for Task 1, Task 2, and InterleavedT1,T2 training across trials. Task 1 and Task 2synaptic weight matrices had mean classification values of -0.069 and 0.069 respectively, while that of InterleavedT1,T2 training was 0.016. (E) Trajectory of H to O layer synaptic weights through PC space. Synaptic weights which evolved during InterleavedT1,T2 training (green dots)clustered in a location of PC space intermediary between the clusters of synaptic weights which evolved during training on Task 1 (red dots) and Task 2 (blue dots).


A
Unsupervised Training
Task 1 Training
Task 2 Training
Interleaved$_{T1,T2}$
Task 1 Testing
Task 2 Testing
Performance
Aeons
B
Testing Performance
Task 1 Performance
Task 2 Performance
Performance
Post T1 Training
Post T2 Training
Post $I_{T1,T2}$
C
Post Task 1
Post Task 2
Post Interleaved$_{T1,T2}$
Single Trial
Average
Task 1 Synapses
Task 2 Synapses
Strength
D
Weight State Classifications
Distance From Decision Boundary
Task 1
Task 2
$I_{T1,T2}$
PCA Trajectory
E
Weight Trajectory
Start Point
Task 1 Training
Task 2 Training
Interleaved$_{T1,T2}$ Training
PC 1
PC 2
PC 3



S6 Figure. Freezing a fraction of task specific strong synapses preserves differing degrees of performance in a sequential learning paradigm. (A-C) Mean and standard deviation of performance during testing on Task 1 (blue) and Task 2 (red). Left, Performance after Task 1 training. Right, Performance after Task 2 training when a fraction of the strongest (after Task 1 training) synapses remained frozen– 1% (A), 5% (B), 10% (C). In all cases, after Task 1 training, Task 1 performance was 0.70 ± 0.02 and Task 2 performance was 0.53 ± 0.02. (A) Freezing the top 1% of Task 1 synapses resulted in a Task 1 performance of 0.54 ± 0.02 and Task 2 performance of 0.68 ± 0.03. (B) Freezing the top 5% of Task 1 synapses resulted in a Task 1 performance of 0.65 ± 0.02 and Task 2 performance of 0.61 ± 0.01. (C) Freezing the top 10% of Task 1 synapses resulted in a Task 1 performance of 0.70 ± 0.03 and Task 2 performance of 0.53 ± 0.03. Freezing the top 1% of Task 1 synapses was not sufficient to maintain Task 1 performance, thus enabling Task 2 relevant synapses to dominate the network; however, freezing the top 10% of Task 1 synapses fully retains Task 1 performance preventing Task 2 to be learned.

Chapter 2, in full, is a reprint of the material as it appears in PLOS Computational Biology. Delanois, J. E., Golden, R., Sanda, P., & Bazhenov, M. (2022). Sleep prevents catastrophic forgetting in spiking neural networks by forming a joint synaptic weight representation. PLOS Computational Biology, 18(11), e1010628.

Improving Robustness of Convolutional Networks Through Sleep-Like Replay

Jean Erik Delanois
Dept. of Computer Science & Engineering
University of California San Diego
La Jolla, USA
jdelanois@ucsd.edu

Aditya Ahuja
Dept. of Computer Science & Engineering
University of California San Diego
La Jolla, USA
adahuja@ucsd.edu

Giri P Krishnan
Dept. of Medicine
University of California San Diego
La Jolla, USA
gkrishnan@ucsd.edu

Timothy Tadros
Dept. of Medicine
University of California San Diego
La Jolla, USA
tttadros@ucsd.edu

Julian McAuley
Dept. of Computer Science & Engineering
University of California San Diego
La Jolla, USA
jmcauley@ucsd.edu

Maxim Bazhenov
Dept. of Medicine
University of California San Diego
La Jolla, USA
mbazhenov@ucsd.edu

***Abstract*—Convolutional neural networks (CNNs) are a foundational model architecture utilized to perform a wide variety of visual tasks. On image classification tasks CNNs achieve high performance, however model accuracy degrades quickly when inputs are perturbed by distortions such as additive noise or blurring. This drop in performance partly arises from incorrect detection of local features by convolutional layers. In this work, we develop a neuroscience-inspired unsupervised Sleep Replay Consolidation (SRC) algorithm for improving convolutional filter's robustness to perturbations. We demonstrate that sleep-based optimization improves the quality of convolutional layers by the selective modification of spatial gradients across filters. We further show that, compared to other approaches such as fine-tuning, a single sleep phase improves robustness across different types of distortions in a data efficient manner.**

***Index Terms*—cnn, convolution, sleep, generalization, robustness**

I. Introduction

Over the past few decades, computer science has made remarkable advancements in the development of models capable of performing intricate visual tasks. Deep learning, in particular, has played a pivotal role in driving this progress, with convolutional neural networks (CNNs) emerging as a significant breakthrough. Inspired by the structural characteristics of the human visual system [8], CNNs owe their success to the introduction of convolutional layers by Lecun et al. [14], [15]. By combining convolutional and feedforward layers, deep networks have achieved state-of-the-art performance for classification and generative tasks [23].

However, despite their proven usefulness, convolutional filters still suffer from significant limitations. While the human visual system excels at accurately performing image-based tasks, even in the presence of substantial perturbations, CNNs trained using backpropagation-based methods are highly sensitive to distortions [4]. The impressive performance of these networks quickly degrade when models operate in real-life applications and dynamic uncontrolled environments modify inputs with perturbations such as additive noise, blur, or other distortions (e.g., lighting, image quality, background, contrast, and perspective) [3]. This decrease in performance could be attributed to the perturbations degrading the quality of features that the convolutional layers are able to extract. Since the convolutional layers are trained on unperturbed (clean) images, they are unable to extract useful features from distorted ones. Most existing methods for improving the robustness of convolutional filters often involve explicit fine-tuning on predefined sets of perturbations or data augmentations [27], [30]. However, such supervised approaches require prior knowledge of the specific deformations or extensive training. These techniques face challenges when limited data is available for fine-tuning or when unforeseen and untrained

Supported by NSF (grant 2223839) and NIH (grant 1R01MH125557)

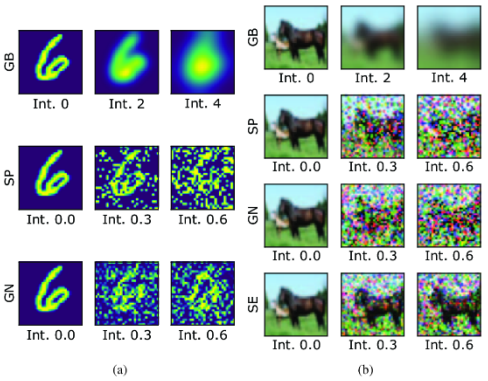


Fig. 1: Example images from MNIST **(a)** and CIFAR-10 **(b)** shown over distortion types (Gaussian noise (GN), Gaussian blur (GB), Salt & pepper (SP), and Speckle (SE)) with varying magnitude. Rows determine distortion type while columns display increasing intensity (Int.) magnitude from left to right.

distortions are encountered in real-world scenarios, this leads to a lack of generalization to out-of-distribution examples.

In contrast, biological systems have leveraged other mechanisms to improve memory representation and increase generalizability. Sleep has long been known to enhance learning in situations with limited experience, facilitate continuous learning, generalize knowledge acquired during wakefulness, and enable backward and forward transfer of knowledge [2], [11], [12], [16], [18], [28]. This functionality is prevalent and highly stereotyped in a variety of species ranging from insects [5], [17], [31] to mammals [2], [18]. Two crucial components are believed to underlie the role of sleep in memory consolidation: the spontaneous replay of memory traces in the absence of external input and local unsupervised synaptic plasticity that modifies synaptic weights [22], [29]. Previous studies have demonstrated that applying sleep-like processing, Sleep Replay Consolidation (SRC), to fully connected feedforward networks can enhance continual learning during sequential task training [25] and improve model robustness and generalizability [24].

While several other biologically inspired approaches to enhance network generalizability to visual distortions exist, they often suffer from increased computational cost [26], lack dynamism [6], or require gathering expensive neural recordings or other hard to acquire data [7], [19]. To address these limitations, we present a novel approach that implements SRC in exclusively convolutional layers, thereby extending the previous work by making SRC applicable to all segments of the CNN architecture. Importantly, our method provides a dynamic solution that does not increase inference computation costs.

SRC is implemented by converting the CNN to a spiking neural network (SNN) and simulating unsupervised replay in SNN. This involves (a) replacing the ReLU activation function with a Heaviside function to gain a notion of spikes, (b) introducing input noise reflective of the training data to induce network activity, (c) applying local Hebbian-type plasticity rules to convolutional layers to modify synapses based on spiking patterns. We evaluate our method using two well-known image classification data sets, MNIST and CIFAR-10, and incorporate standard distortions commonly encountered in both machine learning and real-world environments. These distortions include Gaussian blur, Additive Gaussian noise, Salt & pepper, and Speckle, with varying intensities. Figure 1 illustrates the diverse range of distortions used for evaluation. Our findings demonstrate that sleep-based optimization enhances the structure of convolutional blocks, enabling CNNs to improve their performance on distorted data.

A. Main contributions

- We develop an unsupervised sleep-like optimization algorithm, Sleep Replay Consolidation (SRC), for convolutional networks to improve robustness and generalization to noisy inputs.
- Our biologically inspired approach is computationally efficient, does not increase inference cost, and does not require prior knowledge of the type of input perturbation while providing improvements across different types of distortions. In contrast, other biologically motivated methods are costly and fine tuning approaches only improve performance on pre-defined augmentations.
- We identify that SRC modifies CNN filters through selective gradient expansion focusing CNN attention to the critical image features that result in improved generalization.

II. Methods

A. Data and Distortions

We tested SRC on two standard image classification data sets, MNIST [15] and CIFAR10 [13]. MNIST consists of 60,000 28x28 monochromatic hand written digits (0-9) while CIFAR-10 contains 60,000 32x32 color images of 10 classes (cars, birds, ships, etc). We applied a variety of common distortions (as used in [4], [6], [19], [26], [27], [30]) to these data sets and tested model performance across a variety of intensities. Certain distortions, such as brightening / darkening, yielded minuscule degradation in performance causing any potential benefits to be masked; we therefore only selected distortions that caused a significant decline in accuracy for the baseline model. All distorted values were clamped at the minimum and maximum pixel values to keep inputs in a reasonable range. Our final set of distortions is detailed below:

- Gaussian blur (GB): Involves convolving the input image with a Gaussian kernel, varying σ values are used to modify intensity. This type of distortion can be introduced when items present in the image are in motion.
- Additive Gaussian noise (GN): Noise drawn from a Gaussian distribution is added pixel-wise to the input image.
- Salt and pepper (SP): Also known as impulse noise, randomly selects input image pixels and sets it to either the minimum or maximum possible input value, the frequency of pixels selected controls the intensity. This type of input noise can arise in digital images taken by cameras with faulty sensors.
- Speckle (SE): A pixel-wise multiplicative noise where a random value is drawn from a Gaussian distribution and multiplied with the original pixel value to generate the new input values. Speckle noise is commonly a result of wave interference in images that are generated through the emission of specific frequencies of light, such as ultrasound and/or radar.

Visualizations of all distortions are shown in Figure 1.

B. Models

In an effort to generate interpretable results, we used smaller, more simple models with the goal of improving transparency and understandability of the underlying mechanisms. For MNIST we used a four layer CNN consisting of two convolutional and two feedforward layers. Both convolutional layers leveraged 3x3 filters with a stride of one, no padding, and a ReLU activation, each filter bank had 1/10 input

channels and 10/20 output channels respectively. After each convolution there was a maxpool with a window size and stride of two. The feedforward layers received an input that matched the output size of the convolutional layers (500) followed by a hidden layer of size 64 with an output size of 10. The hidden layer leveraged a ReLU activation function and dropout during training with a rate of 0.5. The CIFAR model was of a similar structure with the only differences being the number of channels in the convolutional layers which was increased to 3/50 and 50/50 and the size of the feedforward portion of the network receiving a 1800 dimensional vector as an input with a 1200 dimensional hidden layer, the output was kept to 10 units. All layers present, both feedforward and convolutional, omitted bias terms to allow for a standard conversion to a spiking neural network [1], this did not notably impact the overall performance of these networks. Model parameters are summarized in Table I.

	MNIST	CIFAR-10
Conv Channels	1. 10, 20	3, 50, 50
Filter Size / Stride	3x3 / 1	3x3 / 1
Maxpool Size / Stride	2 / 2	2 / 2
FF Layer Dims	500, 64, 10	1800, 1200, 10
Dropout	0.5	0.3

TABLE I: Network parameters

C. Sleep Replay Consolidation (SRC)

In short, SRC is applied by first converting a CNN to an SNN using a standard transformation [1], followed by simulated replay, during which unsupervised synaptic modifications occur. The altered SNN is then converted back into a CNN where the updated weights can be used in the conventional CNN forward pass.

In the SNN conversion, original network structure is preserved. A membrane potential (voltage) is simulated for each node in the network. Voltage is comprised of a running sum of inputs determined by presynaptic activity combined with the input weights and is subject to decay, effectively simulating dynamics of a leaky integrate and fire neuron. The ReLU activation is swapped for a Heaviside function to develop a notion of spikes. Once a neuron's membrane potential surpasses the given threshold, the neuron emits a spike and the voltage is reset to 0. To ensure that activity propagated across layers, layer wise scale factors to synaptic weights are generated in accordance with the Data-Based Normalization technique specified in [1] and multiplied by a hyperparameter coefficient. These modifications are applied to convolutional layer neurons, successfully converting CNN to SNN, while preserving network architecture and synaptic weight structure.

During the sleep phase, the SNN's activity is driven by randomly distributed Poisson spiking input with firing rates determined by the average values of each input pixel activation from the training data set. Hebbian style learning rules are applied to modify the weights: a weight is increased between two nodes when both pre- and post-synaptic nodes are activated and a weight is decreased when the post-synaptic node is activated but the pre-synaptic node is not. After this

Algorithm 1 : Sleep Replay Consolidation

procedure SLEEP($nn, I, scales, thresholds$) ▷ I is input
  Initialize v (voltage) $= 0$ vectors for all neurons
  for $t \leftarrow 1$ to Ts **do** ▷ Ts - Time step duration of sleep
    $\mathbf{S} \leftarrow 0s$
    $\mathbf{S}(1) \leftarrow$ Convert input I to Poisson-distributed spiking activity
    $\mathbf{S}$ = ForwardPass($\mathbf{S}, v, \mathbf{W}, scales, thresholds$)
    $\mathbf{W}$ = BackwardPass($\mathbf{S}, \mathbf{W}$)
  end for
end procedure
procedure FORWARDPASS($S, v, W, scales, threshold$)
  for $l \leftarrow 2$ to n **do** ▷ n - number of layers
    $\alpha \leftarrow scales(l-1)$
    $\beta \leftarrow threshold(l)$
    $v(l) \leftarrow \lambda v(l) + (\alpha * \mathbf{W}(l, l-1) * \mathbf{S}(l-1))$ ▷ W(l,l-1) - weights
    ▷ λ - decay rate
    $\mathbf{S}(l)_i \leftarrow 1 \ \forall \ i$ where $v(l)_i > \beta$ ▷ Propagate spikes
    $v(l)_i \leftarrow 0 \ \forall \ i$ where $v(l)_i > \beta$ ▷ Reset spiking voltages
  end for
return S
end procedure
procedure BACKWARD PASS(S, W)
  for $l \leftarrow 2$ to n **do** ▷ n - number of layers
    if isConvolutionalLayer(l) **then**
      $F \leftarrow$ getConvolutionalFilters(l) ▷ All filters in layer l
      for f in F **do** ▷ Loop over all filters
        $L_f \leftarrow$ getFilterActivations(f) ▷ Pre/post activations for f
        for (l_{f-}, l_{f+}) in L_f **do** ▷ For all input/output filters
          $S(l_{f-}) \leftarrow$getSpikes($f-$). ▷ Presynaptic activity
          $S(l_{f+}) \leftarrow$getSpikes($f+$). ▷ Postsynaptic activity
          $\mathbf{W}(f)_{i,j} \leftarrow \begin{cases} \mathbf{W}(f)_{i,j} + inc & \forall i,j \text{ where } \mathbf{S}(l_{f+})_j = 1 \ \& \ \mathbf{S}(l_{f-})_i = 1 \\ \mathbf{W}(f)_{i,j} - dec & \forall i,j \text{ where } \mathbf{S}(l_{f+})_j = 1 \ \& \ \mathbf{S}(l_{f-})_i = 0 \\ \mathbf{W}(f)_{i,j} & \text{Otherwise} \end{cases}$
          ▷ Conv STDP
        end for
      end for
    else
      $\mathbf{W}(l, l-1)_{i,j} \leftarrow \begin{cases} \mathbf{W}(l, l-1)_{i,j} + inc & \forall i,j \text{ where } \mathbf{S}(l)_j = 1 \ \& \ \mathbf{S}(l-1)_i = 1 \\ \mathbf{W}(l, l-1)_{i,j} + dec & \forall i,j \text{ where } \mathbf{S}(l)_j = 1 \ \& \ \mathbf{S}(l-1)_i = 0 \\ \mathbf{W}(l, l-1)_{i,j} & \text{Otherwise} \end{cases}$
      ▷ Linear STDP
    end if
  end for
return W
end procedure

	MNIST	CIFAR
No. of Time Steps (Ts)	222	10
Weight Multiplier ($scales$ coefficient)	2.78	46.81
Voltage Thresholds ($thresholds$)	[4.15, 9.47]	[7.00, 23.96]
Decay Rate (λ)	0.99	0.94
Synaptic Increase (inc)	$3.87 * 10^{-4}$	$6.52 * 10^{-4}$
Synaptic Decrease (dec)	$-3.13 * 10^{-4}$	$-1.98 * 10^{-4}$
Dt	0.001	0.001
Max Firing Rate	328.89	64.62

TABLE II: Hyperparameters used for SRC. Corresponding variable names as used in Algorithm 1 are within parentheses. Dt and the Max Firing Rate are used to generate input for the sleep stage.

unsupervised sleep period has been executed, the CNN model is restored by eliminating the simulated voltage, removing scale factors, and restoring the original activation functions. A pseudo code description of SRC is shown in Algorithm 1.

This approach can be directly applied to a fully connected network (as in [25]) since it produces one-to-one mapping from any pair of pre and post activations to the corresponding weights. However, implementing this to convolutional layers

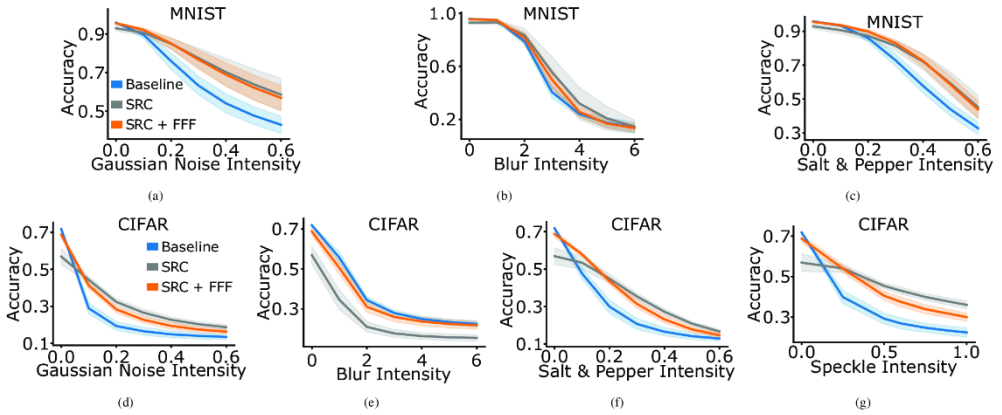


Fig. 2: MNIST (a-c) and CIFAR-10 (d-g) accuracy vs distortion intensity for Gaussian Noise, Blur, Salt & Pepper, and Speckle. Lines / shaded regions correspond to mean / standard deviation across trials. Note that the application of SRC notably improves performance on distorted inputs over baseline model.

is more complicated. Because of parameter sharing, a single weight may take part in multiple synaptic events. Thus, based on the network activity, we have an option of updating the same set of weights multiple times during a single iteration of SRC. Our implementation therefore accumulates synaptic updates over all activations that are associated to a given convolutional weight for every iteration.

The SRC hyperparameters were selected through the use of a standard python Genetic Algorithm implementation tasked to optimize mean validation performance over the Blur and Salt & Pepper distortions for a single trial. The optimal hyper parameters were used across trials to ensure no overfitting occurred, all the parameters are presented in Table II.

D. Experimental Design

All models underwent a standard training protocol. The naive MNIST / CIFAR model was trained for 50 epochs with a learning rate of 0.01 / 0.3 on the undistorted data set until a steady performance was reached. A binary cross entropy loss function along with a standard stochastic gradient decent optimizer was used to alter model parameters. Following baseline training the model underwent periods of SRC and subsequent Feedforward Fitting (Described in Section III-B). Each experiment below was repeated for 10 trials, each of 10 trials received a unique random seed causing differences in model weight initialization, training sample order, and SRC input noise generation.

III. Results

A. SRC improves model performance on distorted data

Our initial set of experiments sought to explore whether SRC was capable of improving CNN generalizability over a variety of distortions for the MNIST and CIFAR-10 data sets. Ten trials (see Methods) were run using the baseline CNN model comprised of two convolutional and two feedforward layers. This model was trained on clean unperturbed images until a plateaued mean performance of roughly 95% (MNIST) and 70% (CIFAR-10) accuracy on the undistorted data set.

The baseline model was tested across a variety of distortions, specifically additive Gaussian noise, Gaussian blur, Salt & Pepper, and Speckle (Speckle noise was excluded from MNIST as maximum intensity minimally degraded baseline performance) with results displayed in Figure 2. There was a direct and clear correlation between distortion intensity and baseline model performance (Figure 2a-g blue line). Increasing distortion intensity led to a significant drop in accuracy, sometimes to chance (see Figure 2b,d,f for intensities (6, 0.6, 0.6) respectively), as the substantial image distortions destroy convolutional feature representations which in turn causes the decision making layers to predict incorrectly.

After establishing the baseline, SRC was applied exclusively to the convolutional layers, as described above, and performance was tested again. We found clear improvement in overall model performance across a wide array of perturbation intensities (see Figure 2; note that the gray line is above the blue line in all cases except for (e)). Particularly for larger distortion values, SRC was capable of improving performance up to roughly 15% for MNIST (Figure 2a, difference between gray and blue) and 10% for CIFAR 10 (Figure 2g, difference between gray and blue). Since SRC weight modifications were only present in convolutional layers, the performance improvements suggest that filter robustness was increased as a result of SRC.

Overall SRC was able to improve performance across most distortion types. However, we found reduced generalizability to the blur distortion, especially for CIFAR 10 (Figure 2e). Although undesirable, it is in line with a variety of biologically inspired works where the given method is not always applicable to all perturbations [19], [6]. While other distortions are

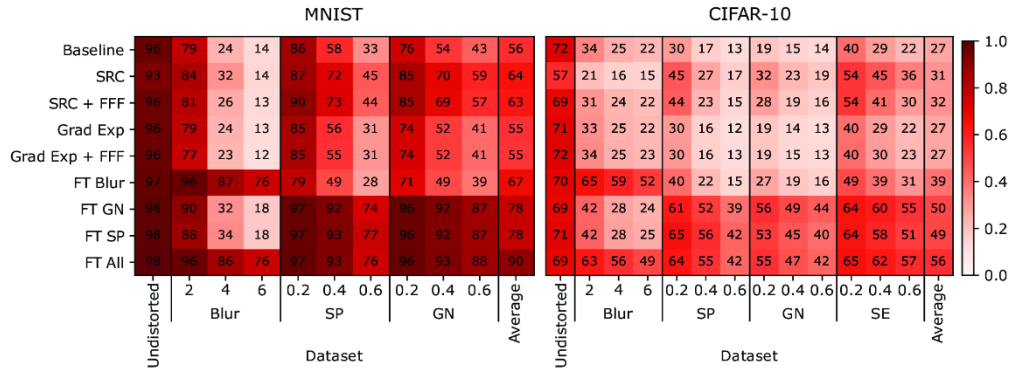


Fig. 3: Model performance on MNIST and CIFAR-10 with varying types and degrees of distortion. The unsupervised SRC phase significantly improved model performance on distorted inputs compared to the baseline while other naive unsupervised approaches (Gradient Expansion) fell short. Although fine-tuning on distortions can enhance performance, it requires extra data and can lack broad generalization.

comprised of pixel-wise perturbations, blurring by definition works on a greater spatial distance which makes it unique. SRC was able to slightly improve MNIST accuracy across greater blur intensities, suggesting that parameter modification may improve performance for blur distortion. It is important to note, the increase in robustness was achieved through a completely unsupervised learning technique which had no information about what specific types of distortions may be used for future testing.

B. Feedforward Fitting (FFF) recovers undistorted performance

While we found a clear improvement in performance on heavily distorted inputs following SRC, we also observed a drop in accuracy for minimally distorted and clean inputs on the order of 1.5% and 10% for MNIST and CIFAR-10 respectively (Figure 2a-g gray below blue for small distortion intensities). Although there may be circumstances where a general model that performs across a wide array of distortions is preferable to a model that performs well narrowly on clean inputs, clearly conserving undistorted performance is desirable. We hypothesised the drop in clean performance may result from a "miss-match" between convolutional and feedforward layers since only convolutional layers were modified by SRC. To test this, an additional training stage was implemented referred to below as Feedforward Fitting (FFF). Here the feedforward head of the network undergoes minimal training on the undistorted training data set; labels along with features extracted by the frozen convolutional weights are used to perform backpropagation on the feedforward layers only. This process thereby adjusts the decision making head of the network to the newly developed feature extractors formed after SRC.

FFF was applied until training set performance was saturated which took 1 / 5 epochs with a learning rate of 0.01 / 0.1 for MNIST / CIFAR. This regained lost performance on the minimally distorted data sets (Figure 2a-g, note orange line near blue line for low distortion values) while significantly maintaining the performance gained for higher distortions (Figure 2a-g orange line near gray line for higher distortion values).

C. Fine-tuning Comparisons

The classic machine learning approach to gain model performance on new data distributions is fine-tuning (FT). Although this is an effective paradigm, it requires foresight of specific potential data perturbations and additional time to train the model. Nevertheless, it represents an ideal accuracy and is used as a benchmark. To compare our unsupervised SRC to this standard supervised method, we developed fine-tuned models each specializing in a specific distortion with one model specializing on all distortions. These fine-tuned models were first initialized using weights from the model trained on undistorted data. They then underwent 10 additional epochs of training (with learning rates of 0.05 / 0.15 for MNIST / CIFAR-10) using the specialized data set comprised of the undistorted data combined with varying levels of distortion from their expertise. The average accuracy across 10 trials for the fine-tuned models along with baseline, SRC, and SRC + FFF models is presented in Figure 3.

As anticipated, each fine-tuned network demonstrated outstanding performance on their respective perturbation, establishing a theoretical performance ceiling for these models on the corresponding distortions (Figure 3). We intuitively predicted fine-tuning on a specific distortion would lead to improved performance on that corresponding perturbation while showing no significant increase, or even a decline, in performance on other distortions. This pattern was evident for the MNIST model fine-tuned on blur which achieved optimal blur performance ranging from 96% - 76% across corresponding blur intensities 2 to 6, while performance on different distortions was below the baseline (Figure 3 left). Interestingly, when the MNIST model was fine-tuned on GN or SP, we observed a remarkable degree of transfer learning

to other distortions; all fine-tuned models for CIFAR-10 also demonstrated this high degree of transfer (Figure 3 right). The reason behind substantial transfer learning in these experiments was not immediately clear as other studies have suggested that this should not typically be the case [10]. While a certain degree of transfer learning between similar distortions might be expected, such as GN and SP (refer to Figure 1 for visualizations), the transfer between dissimilar distortions could be attributed to the simplicity of our data sets or the small size of our models which may act as a form of regularization.

Overall we found the fine-tuned models to be top performers in their respective domains, with the model fine-tuned on all distortions achieving the highest overall average accuracy. We also saw transfer learning proportional to degree of similarity between the trained and tested distortion types. SRC was able to outperform fine-tuned models on untrained distortions where little transfer learning was observed. When a high degree of transfer learning was present, the fine-tuned models outperformed SRC (e.g., fine-tuning on SP, GN and SE led to higher performance compare to SRC or SRC + FFF across distortions). However, it is important to note that the fine tuned models required a significantly higher degree of training. Specialized models were trained for an additional 10 epochs on a fine-tuning data set that contained seven times the number of training examples as in the original training set (one partition undistorted and 6 partitions of varying degrees of distortions). In contrast, SRC was able to increase generalizability with no additional data, highlighting the fact that SRC may also be a more efficient approach to increase model robustness when specifics of anticipated distortions are unknown.

D. Gradient Expansion

To gain insight as to why SRC is capable of improving model performance, we performed a weight analysis on the convolutional filters. Examining the spatial gradient of convolutional filters is often used as a metric for filter quality [9], [20], by inspecting the quality of filters across all convolutional blocks in the network we can determine the quality of the CNN. We developed a measure that is computed by simply taking the pixel-wise spatial gradient (for all filters in a given layer) and fitting a Gaussian probability distribution to their values, thereby obtaining a probabilistic representation for the filter gradients in each convolutional layer. We can examine the properties of this distribution, for instance the variance, to understand the estimated quality of convolutional blocks. A narrow distribution would imply many repeated filters while a wider distribution would suggest a large variety of filters - this variability could enable rich feature extraction and therefore be beneficial for classification.

	Baseline	Baseline + SRC	Baseline + GradExp
MNIST (C1)	$7.21 * 10^{-2}$	$1.47 * 10^{-1}$	$2.72 * 10^{-1}$
MNIST (C2)	$1.06 * 10^{-2}$	$4.36 * 10^{-2}$	$4.18 * 10^{-2}$
CIRAR (C1)	$1.48 * 10^{-1}$	$1.71 * 10^{-1}$	$1.83 * 10^{-1}$
CIFAR (C2)	$9.75 * 10^{-3}$	$1.04 * 10^{-2}$	$1.03 * 10^{-2}$

TABLE III: The mean standard deviation of spatial gradient variance across models. C1 and C2 refer to the results for the first and second convolution layer, respectively. We observe that both the SRC and GradExp models increase variance of the spatial gradient, however these changes are accompanied by a performance increase only in the SRC model.

We noted that sleep increases the variance of the convolutional filter's spatial gradient distribution across layers (compare first two columns in Table III). This can be interpreted as SRC producing more diverse and robust feature extractors through local activation patterns within the network and offers a possible explanation as to why sleep-like replay is capable of improving model performance across distortions.

	Baseline / SRC	GradExp / SRC
MNIST (C1)	$1.4984 * 10^{-1}$	$3.1373 * 10^{-2}$
MNIST (C2)	$3.0821 * 10^{-1}$	$7.6575 * 10^{-3}$
CIRAR (C1)	$6.4440 * 10^{-3}$	$2.8431 * 10^{-4}$
CIFAR (C2)	$8.7511 * 10^{-4}$	$2.5147 * 10^{-5}$

TABLE IV: KL divergence values between the baseline & SRC models (left column), and the Gradient Expansion (GradExp) & SRC models (right column). C1 and C2 refer to results for the first and second convolution layer respectively. Note that distributions on the right are much more similar than distributions on the left, displaying that the spatial gradient distributions of SRC and GradExp are similar - while both being different from the baseline.

To test if simply increasing the variance of filter spatial gradient magnitudes would increase performance, we artificially expanded the spatial gradients of the convolutional filters from the baseline model to approximate distribution of those in the SRC model (compare columns 1 and 3 in Table III). Thus, we choose a set of hyperparameters $\{\alpha_1, ..., \alpha_L\}$ (see Table V for selected values), and increase the absolute value of all filter elements by that amount (Eq. 1). To account for layer specific weight statistics, we choose different α_l values for each layer to approximate changes observed following SRC:

$$W(l) = \begin{cases} W(l) + \alpha_l, & \text{if } W(l) \geq 0 \\ W(l) - \alpha_l, & \text{otherwise} \end{cases} \quad (1)$$

To ensure that these generated Gradient Expansion (GradExp) models have different spatial gradient distributions from our baseline model yet are similar to SRC models, we measured the KL divergence of the convolutional filter's spatial gradient distributions for baseline vs. SRC and SRC vs. GradExp models (Table IV). We found a relatively high KL divergence between baseline and SRC (left column), signifying SRC is meaningfully modifying filters, and a relatively low divergence between SRC and GradExp models (right column) thereby verifying that our artificially generated spatial gradients are statistically similar to those achieved through SRC.

Two versions of the gradient expanded model were tested across distortion intensities for both MNIST and CIFAR-10. The first expanded convolutional filter gradients exclusively, the second applied Feedforward Fitting (FFF) to the network head (utilizing the same hyperparameters described in Section III-B) following filter gradient expansion to allow the decision layers to acclimate to the new feature extractors. Average MNIST and CIFAR-10 accuracy of these models across 10 trials is shown in Figure 3. Both variants of this model show no improvement over baseline (less than 1%) on any

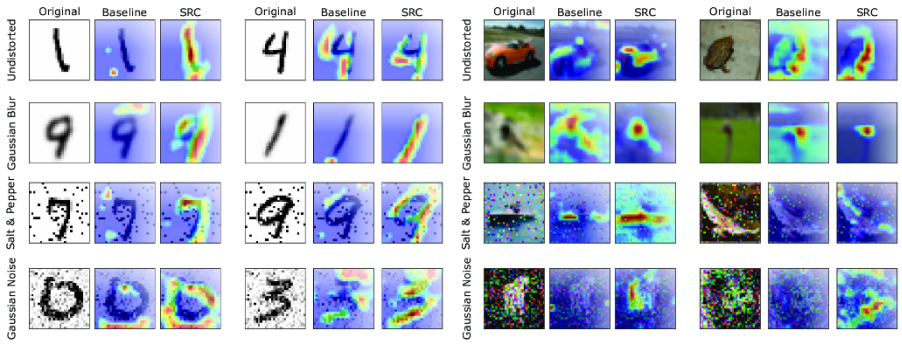


Fig. 4: Grad-CAM visualizations for MNIST **(a)** and CIFAR-10 **(b)** that display SRC improves attention quality over baseline model.

MNIST, Trial 1 - 10	
(C1)	[0.70 0.21 0.32 0.11 0.43 0.31 0.34 0.34 0.19 0.11]
(C2)	[0.11 0.09 0.09 0.10 0.08 0.14 0.08 0.07 0.12 0.11]

CIFAR, Trial 1 - 10	
(C1)	[0.035 0.050 0.050 0.080 0.070 0.060 0.075 0.060 0.065 0.040]
(C2)	[0.005 0.005 0.004 0.003 0.003 0.005 0.005 0.004 0.004 0.004]

TABLE V: Hyperparameters (α_l) for Gradient Expansion as described in Section III-D. We list values used for each of the 10 random trials. C1 and C2 refer to results for the first and second convolution layer respectively.

distortion intensity for either data set. This demonstrates that a general increase of the filter gradients is not sufficient to create robust filters resistant to input perturbations. This further suggests that SRC enables selective increases in the magnitude of convolutional spatial gradients. Additionally, the fact that applying FFF following gradient expansion does not increase performance shows that further feedforward training on equivalent quality convolutional filters is futile. Only if the feedforward head is allowed to train on higher quality convolutional blocks, like the ones developed in SRC, is there an improvement in distorted and undistorted performance.

E. Model attention and Grad-CAM analysis

To gain a deeper qualitative and quantitative understanding of how SRC impacts the network, analysis was developed using Gradient-weighted Class Activation Mapping (Grad-CAM) [21]. Grad-CAM is a visualization technique that creates an attention map for a given input to identify what the network focuses on. It operates by supplying an image as input and performing a forward pass followed by the calculation of gradients with respect to a given output label. Gradient values are then used to weight final convolutional activations (which maintain their spatial relevance), the intuition being more important features will have higher gradient values. This approach develops a notion of what input regions the network is attending to.

Generally speaking, we were able to observe improvements in attention as a result of SRC, some of the best examples from both MNIST and CIFAR-10 are displayed in Figure 4 panels a and b, respectively. The results were particularly dramatic for MNIST. Given the original input image (Figure 4a, 1^{st} and 4^{th} column), the baseline model often attends to seemingly random pixels even on clear images (Figure 4a, 1^{st} row, 2^{nd} column). However, after SRC, model attention overlapped with the original input image significantly better (Figure 4a 3^{rd} and 6^{th} column). Importantly, SRC significantly enhanced attention on perturbed images. In the presence of noise the baseline model would often attend to noisy pixels or attention would be disrupted away from the original digit. Following SRC, the model was able to cut through the noise and the attention heat map took the shape of the original digit, implying the network is focusing on relevant features as opposed to irrelevant noise. A similar result was obtained for CIFAR-10 (Figure 4b) although the improvement was less consistent, some images displayed no improvement while others displayed clear benefit.

In an attempt to quantify attention improvements, we constructed a rudimentary metric that was compatible with the MNIST data set. The metric consisted of developing a pixel wise mask of the original digit (1's were assigned to input locations with nonzero pixel values and 0's everywhere else) followed by a cosine similarity between the mask and the attention vector output by Grad-CAM. Values close to 1 indicate a large overlap between the clean input image and the network's attention while values near 0 signify a misplaced network focus. This metric was averaged across all trials for every distortion / intensity combination for each model with the results displayed in Table VI. The overlap of attention and the original undistorted input digit is significantly higher for the model that underwent SRC when compared to the baseline or GradExp models. This implies the nontrivial selective filter gradient enhancement provided by SRC was able to improve convolutional filter quality and focus, even in the presence of meaningful perturbation; thereby increasing model performance.

Model	Baseline	SRC	SRC + FFF	Grad Exp	Grad Exp + FFF
Attention Overlap	0.145	0.229	0.193	0.146	0.150

TABLE VI: Grad-CAM Attention Overlap Metric. It can be seen that the SRC increases attention overlap with the ground truth image over baseline. Gradient Expansion models also increase accurate attention but without the performance benefit seen with SRC.

IV. Conclusion

In this work we developed a biologically inspired sleep-like optimization stage, termed the Sleep Replay Consolidation (SRC) algorithm, and showed it is compatible with CNNs and capable of improving convolutional filter quality thereby increasing model performance on distorted data sets. We examined SRC on standard image classification data sets, MNIST and CIFAR-10, and found that it substantially improves performance for moderate to high levels of distortion intensity. We further identified mechanisms of improvement as related to non-linear selective expansion of the convolutional filter's spatial gradient distribution across layers. Our study, combined with previous work [24], [25], suggests that sleep-like unsupervised replay may provide multiple benefits to different classes of ANNs, including improving continual learning, generalization and adversarial robustness.

References

[1] Diehl, P. U., Neil, D., Binas, J., Cook, M., Liu, S.-C., and Pfeiffer, M. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *2015 International Joint Conference on Neural Networks (IJCNN)* (2015), ieee, pp. 1–8.

[2] Diekelmann, S., and Born, J. The memory function of sleep. *Nature Reviews Neuroscience 11*, 2 (Jan. 2010), 114–126.

[3] Dodge, S., and Karam, L. Understanding how image quality affects deep neural networks. In *2016 eighth international conference on quality of multimedia experience (QoMEX)* (2016), IEEE, pp. 1–6.

[4] Dodge, S., and Karam, L. A study and comparison of human and deep learning recognition performance under visual distortions. In *2017 26th international conference on computer communication and networks (ICCCN)* (2017), IEEE, pp. 1–7.

[5] Donlea, J. M., Thimgan, M. S., Suzuki, Y., Gottschalk, L., and Shaw, P. J. Inducing sleep by remote control facilitates memory consolidation in idrosophila/i. *Science 332*, 6037 (June 2011), 1571–1576.

[6] Evans, B. D., Malhotra, G., and Bowers, J. S. Biological convolutions improve dnn robustness to noise and generalisation. *Neural Networks 148* (2022), 96–110.

[7] Fel, T., Rodriguez Rodriguez, I. F., Linsley, D., and Serre, T. Harmonizing the object recognition strategies of deep neural networks with humans. *Advances in Neural Information Processing Systems 35* (2022), 9432–9446.

[8] Fukushima, K., and Miyake, S. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*. Springer, 1982, pp. 267–285.

[9] Gavrikov, P., and Keuper, J. Cnn filter db: An empirical investigation of trained convolutional filters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 19066–19076.

[10] Geirhos, R., Temme, C. R. M., Rauber, J., Schütt, H. H., Bethge, M., and Wichmann, F. A. Generalisation in humans and deep neural networks. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 7538–7550.

[11] Ji, D., and Wilson, M. A. Coordinated memory replay in the visual cortex and hippocampus during sleep. *Nature neuroscience 10*, 1 (2007), 100–107.

[12] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences 114*, 13 (2017), 3521–3526.

[13] Krizhevsky, A., and Hinton, G. Learning multiple layers of features from tiny images, 2009.

[14] LeCun, Y., Bengio, Y., et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks 3361*, 10 (1995), 1995.

[15] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE 86*, 11 (1998), 2278–2324.

[16] Lewis, P. A., and Durrant, S. J. Overlapping memory replay during sleep builds cognitive schemata. *Trends in cognitive sciences 15*, 8 (2011), 343–351.

[17] Melnattur, K., Kirszenblat, L., Morgan, E., Militchin, V., Sakran, B., English, D., Patel, R., Chan, D., van Swinderen, B., and Shaw, P. J. A conserved role for sleep in supporting spatial learning in idrosophila/i. *Sleep 44*, 3 (Sept. 2020).

[18] Rasch, B., and Born, J. About sleep's role in memory. *Physiological Reviews 93*, 2 (Apr. 2013), 681–766.

[19] Safarani, S., Nix, A., Willeke, K., Cadena, S., Restivo, K., Denfield, G., Tolias, A., and Sinz, F. Towards robust vision by multi-task learning on monkey visual cortex. *Advances in Neural Information Processing Systems 34* (2021), 739–751.

[20] Schubert, L., Voss, C., Cammarata, N., Goh, G., and Olah, C. High-low frequency detectors. *Distill 6*, 1 (2021), e00024–005.

[21] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 618–626.

[22] Stickgold, R. Sleep-dependent memory consolidation. *Nature 437*, 7063 (2005), 1272–1278.

[23] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 1–9.

[24] Tadros, T., Krishnan, G., Ramyaa, R., and Bazhenov, M. Biologically inspired sleep algorithm for increased generalization and adversarial robustness in deep neural networks. In *International Conference on Learning Representations* (2019).

[25] Tadros, T., Krishnan, G. P., Ramyaa, R., and Bazhenov, M. Sleep-like unsupervised replay reduces catastrophic forgetting in artificial neural networks. *Nature Communications 13*, 1 (2022), 7742.

[26] Teti, M., Kenyon, G., Migliori, B., and Moore, J. Lcanets: Lateral competition improves robustness against corruption and attack. In *International Conference on Machine Learning* (2022), PMLR, pp. 21232–21252.

[27] Vasiljevic, I., Chakrabarti, A., and Shakhnarovich, G. Examining the impact of blur on recognition by convolutional networks. *arXiv preprint arXiv:1611.05760* (2016).

[28] Walker, M. P., and Stickgold, R. Sleep-dependent learning and memory consolidation. *Neuron 44*, 1 (2004), 121–133.

[29] Wei, Y., Krishnan, G. P., and Bazhenov, M. Synaptic mechanisms of memory consolidation during sleep slow oscillations. *Journal of Neuroscience 36*, 15 (2016), 4231–4247.

[30] Zhou, Y., Song, S., and Cheung, N.-M. On classification of distorted images with deep convolutional neural networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2017), IEEE, pp. 1213–1217.

[31] Zwaka, H., Bartels, R., Gora, J., Franck, V., Culo, A., Götsch, M., and Menzel, R. Context odor presentation during sleep enhances memory in honeybees. *Current biology : CB 25*, 21 (November 2015), 2869—2874.

Chapter 3, in full, is a reprint of the material as it appears in ICMLA.

Delanois, J. E., Ahuja, A., Krishnan, G. P., Tadros, T., McAuley, J., & Bazhenov, M. (2023, December). Improving Robustness of Convolutional Networks Through Sleep-Like Replay. In 2023 International Conference on Machine Learning and Applications (ICMLA) (pp. 257-264). IEEE.

Conclusion

In conclusion, this dissertation offers advancements in the interdisciplinary understanding of neuroscience and artificial intelligence. It demonstrates how the incorporation of sleep and sleep-like stages can enhance memory consolidation, representation, and robustness across neural networks with varying degrees of biological realism. Through biophysical modeling, this research proposes potential sleep-induced synaptic dynamics that could be crucial for memory consolidation in living organisms. When analogous mechanisms were applied to artificial spiking neural networks, the observed memory enhancement indicated the benefits of sleep extend to artificial contexts as well. Furthermore, incorporating sleep-like stages in artificial neural networks was shown to enhance synaptic memories and feature representations, leading to improved model performance. This work not only elucidates potential intricate mechanisms underlying sleep and memory consolidation in biological brains but also reconceptualizes the relationship between artificial neural networks and sleep-like processes. These advancements pave the way for further understanding living brains and developing more robust and brain-like artificial intelligence systems, thereby helping to bridge the gap between biological and artificial intelligence.