



OPEN ACCESS

EDITED BY

Fernando Montani,
National Scientific and Technical Research
Council (CONICET), Argentina

REVIEWED BY

Eugenio Urdapilleta,
Bariloche Atomic Centre (CNEA), Argentina
Germán Mato,
Bariloche Atomic Centre (CNEA), Argentina

*CORRESPONDENCE

Paul Saighi
✉ paul.saighi@universite-paris-saclay.fr

RECEIVED 28 June 2025

ACCEPTED 05 August 2025

PUBLISHED 26 August 2025

CITATION

Saighi P and Rozenberg M (2025)
Autonomous retrieval for continuous learning
in associative memory networks.
Front. Comput. Neurosci. 19:1655701.
doi: 10.3389/fncom.2025.1655701

COPYRIGHT

© 2025 Saighi and Rozenberg. This is an
open-access article distributed under the
terms of the [Creative Commons Attribution
License \(CC BY\)](#). The use, distribution or
reproduction in other forums is permitted,
provided the original author(s) and the
copyright owner(s) are credited and that the
original publication in this journal is cited, in
accordance with accepted academic practice.
No use, distribution or reproduction is
permitted which does not comply with these
terms.

Autonomous retrieval for continuous learning in associative memory networks

Paul Saighi^{1*} and Marcelo Rozenberg^{1,2}

¹Laboratoire de Physique des Solides, CNRS, Université Paris-Saclay, Orsay, France, ²CNRS, Integrative Neuroscience and Cognition Center, Université Paris-Cité, Paris, France

The brain's faculty to assimilate and retain information, continually updating its memory while limiting the loss of valuable past knowledge, remains largely a mystery. We address this challenge related to continuous learning in the context of associative memory networks, where the sequential storage of correlated patterns typically requires non-local learning rules or external memory systems. Our work demonstrates how incorporating biologically inspired inhibitory plasticity enables networks to autonomously explore their attractor landscape. The algorithm presented here allows for the autonomous retrieval of stored patterns, enabling the progressive incorporation of correlated memories. This mechanism is reminiscent of memory consolidation during sleep-like states in the mammalian central nervous system. The resulting framework provides insights into how neural circuits might maintain memories through purely local interactions and takes a step forward toward a more biologically plausible mechanism for memory rehearsal and continuous learning.

KEYWORDS

neural networks, memory consolidation, continuous learning, catastrophic forgetting, unsupervised learning, neuromorphic computing, associative memory networks

1 Introduction

Continuous learning (CL) refers to a system's ability to maintain performance across multiple tasks when operating in environments that evolve over time, requiring adaptation to changing data distributions. To do so, the learning mechanism should avoid uncontrolled forgetting of previously acquired knowledge when adapting to new information or contexts. In associative memory networks, this challenge arises when storing new activity patterns sequentially deteriorates existing memory representations, a phenomenon called catastrophic forgetting.

It is important to note that this work specifically addresses catastrophic forgetting in the context of sequential learning. This approach addresses a different challenge than the well-studied spin glass phase transitions that occur in Hopfield networks at high memory loads.

Memory rehearsal is a method that addresses the challenge of catastrophic forgetting by periodically retraining the model on previously stored patterns. This process reinforces older memory representations, preventing their degradation when new information is incorporated (Robins, 1995; McCallum, 1998). In the mammalian nervous system, spontaneous memory replays occur during sleep (Robins, 1995; Louie and Wilson, 2001; Peyrache et al., 2009; Fauth and Van Rossum, 2019; Tononi and Cirelli, 2014), suggesting a biological mechanism analogous to rehearsal techniques in artificial networks (Robins, 1995). This parallel raises a fundamental question: what mechanisms enable these autonomous memory replays in biological systems?

Previous research on bioinspired neural networks demonstrates that short-term synaptic depression can facilitate spontaneous rehearsal of neural assemblies (Fauth and Van Rossum, 2019). However, a significant constraint of this approach is its dependence on minimal overlap between neural assemblies. The neuronal populations in these studies share few neurons, resulting in effectively decorrelated memory representations. By contrast, classical associative memory networks like Hopfield Networks can effectively store uncorrelated memories that share many units. Some learning algorithms even allow the storage of highly correlated patterns that share a majority of their units (Diederich and Oppel, 1987). From a biological perspective, understanding how networks implement the rehearsal of correlated populations is crucial as neural representations found in the cortex generally recruit extensively overlapping assemblies (Haxby et al., 2001; Kriegeskorte, 2008). The maintenance of overlapping assemblies is widely considered essential for cortical computation, as it supports stimulus generalization and the emergence of invariant, high-level concepts in which individual neurons participate in multiple but related representations (Haxby et al., 2001; Quiroga et al., 2005; Kriegeskorte, 2008). This problematic is of similar importance in neuromorphic engineering contexts as highly correlated representations are an emerging feature of artificial neural networks (Kothapalli, 2023).

How the rehearsal of correlated memories takes place autonomously on neural substrates remains a largely unaddressed question. In this work, we focus on this issue and explore its potential application for continuous learning (CL). For the sake of bio-plausibility and potential implementation in neuromorphic substrates, we shall demand that our system exhibits the following features: 1) It stores memory states that can be highly correlated; 2) During the pattern recovery, the network does not converge toward strange attractors which would constitute false memories; 3) Plasticity rules are local, meaning that the modification of synaptic efficacy can be computed in terms of its pre- and post-synaptic neuron states; 4) It is autonomous, namely, it should retrieve all previously stored patterns from its own dynamics. The network does not have access to an external list of previously recorded memory states. For the sake of requirements (1) and (3), we shall adopt a perceptron-like algorithm, inspired by the work of Diederich and Oppel (1987). On the other hand, for requirement (2), we shall use continuous Hopfield Networks (CHNs) (Hopfield, 1984). Our work demonstrates that, kept under a certain memory load, CHNs converge exclusively to stored patterns during the retrieval. This approach avoids both the spurious state proliferation common in Discrete Hopfield Networks (DHNs) and the shortcomings of temperature parameter fine-tuning inherent to Stochastic Hopfield Networks (SHNs) (Amit et al., 1985a).

The main contribution of the present work is to introduce an algorithm to address the requirement (4). A crucial feature of our approach is the use of self-inhibition to shrink the basin of attraction of previously visited attractor states, thus allowing for a sequential and thorough search and recovery of all previously stored correlated memory states. This recovery effectively allows the rehearsal of the stored pattern for CL purposes. The dynamic of plastic recurrent inhibition is inspired by computational

neuroscience work based on actual neurophysiological data (Vogels et al., 2011).

2 Methods

2.1 Continuous Hopfield Network (CHN)

In contrast with the conventional DHN model (Hopfield, 1982), where neural states are defined as binary variables, in a CHN they are continuous (Hopfield, 1984). The dynamics of the network is defined by a set of differential equations with each neuron unit described as a leaky integration:

$$c \frac{du_i}{dt} = \sum_j W_{ij} v_j - \frac{u_i}{r} \quad (1)$$

where u_i is the membrane potential of neuron i , c is the membrane capacitance, r is the leak resistance of each neuron, W_{ij} is the synaptic efficacy between neurons j and i , and v_i is the activity (or firing rate) of neuron i that depends solely on the potential as

$$v_i = \sigma(u_i)$$

where σ is a monotonically increasing function of u with saturation to prevent runaway dynamics. We adopt $\sigma(x) = \frac{1}{1+\exp(-x)}$. Therefore, as $\sigma(0) = 0.5$, each unit has a positive output at the resting state, allowing the network to have a baseline activity without external input. We shall refer to either the vector $\mathbf{v}(t)$ or $\mathbf{u}(t)$ as “states”, which should be clear from the context.

The convergence of the flow to stable states for the case of symmetric synaptic weights W_{ij} has been demonstrated (Hopfield, 1984). Throughout this work, whenever we integrate these equations using Euler method, we do so until the network reaches convergence, defined as $|\frac{du}{dt}|_{\infty} < \epsilon$, where $\epsilon = 10^{-6}$.

2.2 Pattern storage and reading

The states $\mathbf{v}(t)$ of the CHN evolve on $[0, 1]^N$ through continuous dynamics, with N the number of neurons. Any state in this space may represent a stored pattern. For simplicity, we restrict ourselves to store binary patterns for which active neurons have a high firing rate, $v_i \approx 1$, and inactive neurons have a low firing rate, $v_i \approx 0$.

Hence, a pattern \mathbf{x}^μ is defined as a binary vector such that $\mathbf{x}^\mu = (x_1^\mu, x_2^\mu, \dots, x_N^\mu)$ where $x_i^\mu \in \{0, 1\}$ for each unit i . A pattern is read from the state of the network at time t using a threshold:

$$x_i^\mu = \begin{cases} 1 & \text{if } v_i(t) > 0.5 \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Given a binary pattern \mathbf{x}^μ , it is convenient to define target potentials $\tilde{\mathbf{u}}^\mu$ as

$$\tilde{u}_i^\mu = \begin{cases} +u_{\text{target}}^\mu & \text{if } x_i^\mu = 1 \\ -u_{\text{target}}^\mu & \text{if } x_i^\mu = 0 \end{cases} \quad (3)$$

We adopt here $u_{\text{target}} = 6$ to ensure proper pattern reading following the thresholding procedure.

Inspired by previous work on DHN (Diederich and Oppel, 1987), we introduce, in Algorithm 1, a perceptron-inspired learning algorithm for efficient storage of correlated patterns in CHNs. The algorithm minimizes the error between the target states and the network's equilibrium states, ensuring that each memory becomes a stable state. Gradient descent methods typically require small step sizes to prevent the optimization process from becoming unstable and to reduce oscillations around local minima. In our implementation, we select $\alpha = 0.0001$ as the learning rate to ensure stable convergence of weight updates. The derivation of the weight update rule can be found in the Appendix 1. Although a rigorous proof of convergence for the algorithm is beyond the scope of this work, we expect that arguments demonstrating the convergence of the gradient descent algorithm (GDA) in the context of DHN could be adapted for this purpose (Diederich and Oppel, 1987). Here, we rely on numerical evidence showing the network's ability to successfully query and revisit stored patterns.

Following the network training with the GDA, each activity vector $\tilde{\mathbf{u}}^\mu$ becomes an attractor. The network can now be queried as the system reliably converges to the nearest stored state from a partial cue. The corresponding binary patterns $\tilde{\mathbf{x}}^\mu$ can then be accurately retrieved by thresholding at time t_f (Equation 3) when the system reaches convergence. The querying procedure is detailed in Algorithm 1 (Appendix 2) and illustrated in Figure 1.

It is worth emphasizing that despite the continuous nature of our model, which theoretically allows for a richer state space, we deliberately restrict ourselves to binary patterns. The interest of using a CHN is the simplicity it allows when implementing our pattern recovery mechanism (Section 2.4), limiting the appearance of false memories as spurious states, which are often encountered in DHNs (Amit et al., 1985b). A variant of the DHN with stochastic units, the SHN (Amit et al., 1985b), would be a possible candidate to implement our algorithm, as they tend to visit only the stored patterns by properly controlling the annealing temperatures. However, the stochastic properties of these networks would require a more complex setup.

```

1: Initialize  $W_{ij} = 0$  for all  $i, j$ 
2: repeat
3:   for each pattern  $\mu$  do
4:     for each neuron  $i$  do
5:       Compute the target potential  $\tilde{u}_j^\mu$ 
         (Equation 3) for each neuron  $j$  with
          $j \neq i$ 
6:       Compute the expected potential:  $\hat{u}_i^\mu =$ 
          $r \sum_j W_{ij} \sigma(\tilde{u}_j^\mu)$ 
7:       Update weights:  $W_{ij} \leftarrow W_{ij} - \alpha(\tilde{u}_i^\mu - \hat{u}_i^\mu) r \sigma(\hat{u}_j^\mu)$ 
8:     end for
9:   end for
10: until  $\|\Delta W\|_\infty < \epsilon$ 

```

Algorithm 1. Gradient descent for the storage of correlated patterns (GDA)

2.3 Continuous incorporation of correlated memories

While the GDA effectively enables the storage of correlated memories, its implementation in Algorithm 1 reveals a significant limitation: It requires multiple iterations over the entire set of patterns to achieve convergence. Without the ability to reprocess all patterns, the network would suffer from catastrophic forgetting, where learning a new pattern in isolation rapidly erodes previously stored memories (McCloskey and Cohen, 1989; Robins, 1995; Kirkpatrick et al., 2017; Shen et al., 2023). By repeatedly processing all patterns, the algorithm can find a weight matrix \mathbf{W} that properly separates the patterns, despite their correlations (Diederich and Oppel, 1987). Adding a new pattern, therefore, requires access to all previously stored patterns from an external source.

This requirement for external access to the complete memory dataset stands in contrast to biological learning systems, which must incorporate new information while maintaining past memories without relying on an explicit external copy of the already stored data. To overcome this external dependency and move toward more biologically plausible learning, a solution is to develop a mechanism that allows the network to internally recover its stored memories.

The development of such an autonomous retrieval mechanism would allow us to exploit the GDA's ability for the continuous incorporation of correlated memories. The continuous learning algorithm is formally defined in Algorithm 2. The act of retraining the network on the whole set is called a rehearsal. Recovering the whole set from the network to allow rehearsal is called retrieval.

2.4 Autonomous retrieval

In this section, we present the autonomous retrieval (AR) mechanism allowing the recovery of stored correlated patterns in a network.

Given a trained network initialized at the “neutral” state, $u_i = 0$ for each unit i , the network dynamics described by Equation 1 converge deterministically to a given stored attractor, which is thus “retrieved”. This attractor can be seen as the dominant attractor from the neutral state. The goal now is to allow for the exploration of other states to permit a complete recovery of the stored memories. To do so, we introduce the adaptation terms A_i .

$$c \frac{du_i}{dt} = \sum_j W_{ij} v_j - A_i v_i - \frac{u_i}{r} \quad (4)$$

$$A_i \leftarrow A_i + \beta v_i(t_f) \quad (5)$$

with $v_i(t_f)$ the firing rate of neuron i after convergence of the dynamics. β represents the adaptation strength and is chosen to be small, typically below 0.1. Adaptation terms could correspond to various components commonly observed in the mammalian central nervous system. On short time scales, spike frequency adaptation (SFA) of excitatory neurons functions as plastic self-inhibition (Ha and Cheong, 2017; Peron and Gabbiani, 2009). Repeated stimulation progressively reduces firing activity in the neuron, mirroring the dynamics produced by our adaptation

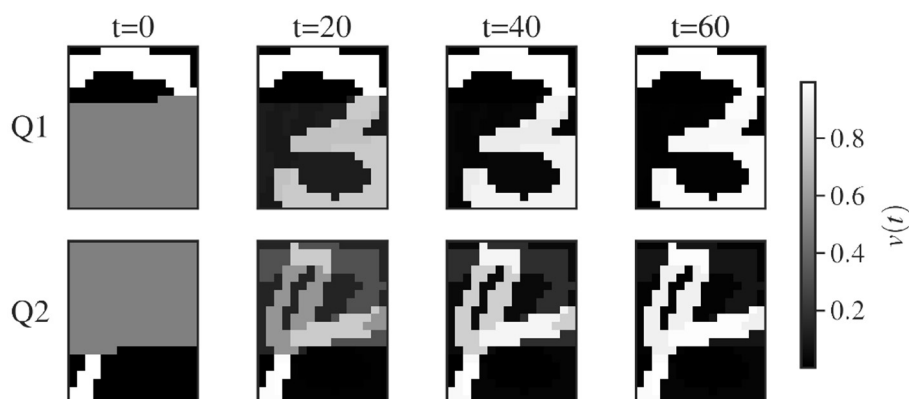


FIGURE 1

Querying of two binary-pattern representation of the handwritten digit “3” and “4” from the MNIST dataset. The network is of size 20×16 as each neuron codes for a pixel. From black to white, the color represents the rate $v_i(t)$ of each unit. At $t = 0$, the network is initialized with the part of the units set to the target value as described in [Appendix 2](#). The figure displays snapshots of the evolution of the rates in time for each unit of the network. The network is queried two times, for the first query Q1 the network gradually settles to the nearest stored state corresponding to the digit “3”. For Q2, the network settles to the stored state corresponding to the digit “4”.

- 1: **Input:** CHN trained on p patterns using the GDA
- 2: **Given:** New pattern \mathbf{x}^{p+1} to be stored
- 3: Retrieve set $\{\mathbf{x}\}$ of stored patterns through autonomous retrieval (AR) introduced in Section 2.4
- 4: Update pattern set: $\{\mathbf{x}\} \leftarrow \{\mathbf{x}\} \cup \{\mathbf{x}^{p+1}\}$
- 5: Apply GDA to store updated pattern set

Algorithm 2. Continuous incorporation of correlated patterns through rehearsal of the whole memory set

A_i . Alternatively, A_i can be interpreted as recurrent inhibition mediated by local inter-neurons, which frequently exhibit Hebbian plasticity ([Kodangattil et al., 2013](#); [D’amour and Froemke, 2015](#)).

After each visited attractor, i.e., memory retrieved, its basin of attraction is made smaller by the update of the adaptation term ([Equation 5](#)). Once a pattern has been inhibited, the probability for the network to converge into it from the neutral state is reduced. By resetting the network to the neutral state after each convergence-inhibition cycle, we allow the sequential recovery of stored patterns.

[Figure 2](#) illustrates the sequential recovery of two stored patterns and the modification of their attractor basin during the procedure. The geometry of these attractor basins explains why a minimal inhibitory influence, resulting from a small β value, is sufficient to alter the trajectory. The neutral state resides near the separatrix that divides the attractor basins. Consequently, even a slight modification of the separatrix position, caused by inhibitory potentiation, can significantly redirect the network’s trajectory.

The increase in adaptation tends to distort the stable states associated with stored patterns. As this distortion is minimal for small β , it is mainly compensated for by the thresholding mechanism used to read the network output (Section 2.2). To further reduce the impact of this distortion, we divide the convergence dynamic into two phases, the “biased” phase and the “free” phase. The biased phase guides the network to converge

toward states that have not been retrieved. It corresponds to the simulation of the network with adaptation, [Equation 4](#), until convergence. The optional free phase then allows the network to complete its convergence to an undistorted stored state. It corresponds to the simulation of the network without inhibitory synapses ([Equation 1](#)) until convergence. The whole procedure is detailed in [Algorithm 3](#).

[Figure 3](#) provides a visualization of AR for binary-pattern representation of handwritten digits from the MNIST dataset ([LeCun et al., 1998](#)). For the first iteration, the inhibitory drive is null as no pattern has been retrieved. The pattern corresponding to the binary picture of a 3 is recovered. The network state is reinitialized with the updated adaptation (**A**). The network now inhibits the recovery of a 3, which induces convergence toward a second pattern, here a 4. Adaptation is updated again and now inhibits both the 3 and the 4 together. Inhibition of the 3 and the 4 combined allows the recovery of the 5 and so on. For the recovery of MNIST binary digits, as a large inhibitory coefficient β has been chosen, the free phase is mandatory to reduce the distortion of the stored pattern.

The order in which patterns will be visited is an emerging feature of the learning algorithm that has not been studied in this work. With each iteration of the AR algorithm, inhibitory synapses undergo potentiation. This growth is constrained only by the number of iterations and the value of β . Theoretically, such an unbounded growth could cause the adaptation A_i to disrupt the attractor dynamics induced by **W**. However, in practice, this potential issue can be managed by adjusting the value of β based on the number of iterations. Specifically, when more iterations are required, using a smaller β is sufficient to maintain robust pattern retrieval without compromising attractor dynamics.

In [Figure 4](#), we illustrate the dynamics of the CHN during the retrieval of stored memory patterns in networks with various loads. The load corresponds to M/N , with M the number of stored memories, and N the size of the network. For low loads (see [Figure 4 top](#)), the system converges sequentially to the attractors

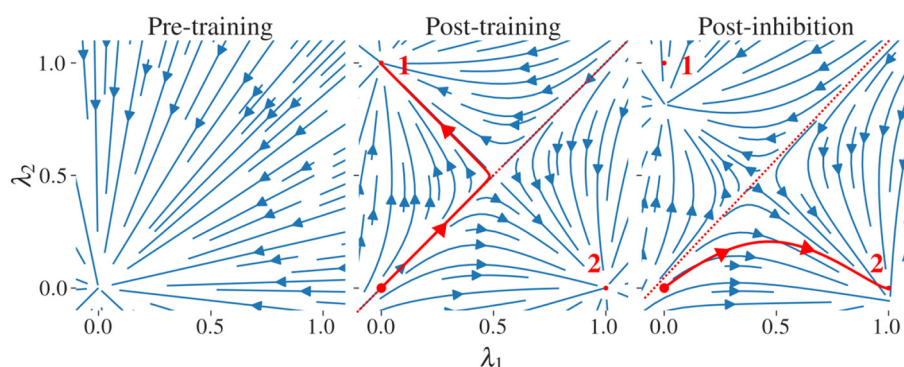


FIGURE 2

Stream plots of the CHN in a 2D subspace spanned by two pattern states. We examine a network with two stored patterns, labeled “1” and “2”, and their stable states \tilde{u}^1 and \tilde{u}^2 ; these define a two-dimensional subspace parametrized by $\tilde{u}(\lambda_1, \lambda_2) = \lambda_1 \tilde{u}^1 + \lambda_2 \tilde{u}^2$. At each point $\tilde{u}(\lambda_1, \lambda_2)$ in this subspace, we compute the full N -dimensional time derivative via Equation 4 and project it back onto the $\{\tilde{u}^1, \tilde{u}^2\}$ -plane to obtain the plotted flow. \tilde{u}^1 and \tilde{u}^2 have minimal correlation as they are generated from random binary patterns \mathbf{x}^1 and $\mathbf{x}^2 \in \{0, 1\}^N$. **(left)** Before training. **(middle)** After storing the two patterns using the GDA, each pattern state (small red dots) becomes a stable attractor; the neutral state $u_N = 0$ (large red dot) is on an unstable manifold. The red line corresponds to the trajectory of the network projected on the subspace. The trajectory follows the unstable manifold before arbitrarily falling into one of the two stored patterns. **(right)** Adaptation **A** is updated to apply enhanced inhibition to pattern 1 which shrinks its basin of attraction. This induces a shift in the position of the separatrix, favoring the flow to pattern 2. An exaggerated large inhibitory coefficient $\beta = 0.5$ is used to illustrate the modification of the vector field. Similar stream plots are obtained from networks storing patterns with various degrees of correlation using the GDA.

```

1: Input: Trained network weights  $W_{ij}$ , number of
   iterations  $k$ , plasticity rate  $\beta$ 
2: Initialize:  $A_i = 0$ ,  $\{\mathbf{x}\} = \emptyset$ 
3: while  $j < k$  do
4:   Set neutral initial conditions:  $\mathbf{u}(t=0) = \mathbf{0}$ 
5:   Biased phase: Integrate Equation 4 until
   convergence to the state  $\mathbf{u}(t_b) = \mathbf{u}_b$ 
6:   Free phase: from the state  $\mathbf{u}(t_b)$ , integrate
   Equation 1 until convergence           ▷ optional
7:   Read pattern  $\mathbf{x}^\mu$  from final state  $\mathbf{v}(t_f)$  via
   thresholding (Equation 3)
8:   Update retrieved pattern set:  $\{\mathbf{x}\} \leftarrow \{\mathbf{x}\} \cup \{\mathbf{x}^\mu\}$ 
9:   Update inhibitory weights:  $A_i \leftarrow A_i + \beta v_i(t_f)$ 
10:   $j \leftarrow j + 1$ 
11: end while
12: Return:  $\{\mathbf{x}\}$ 

```

Algorithm 3. Autonomous retrieval (AR)

corresponding to the stored patterns. Once every memory has been recovered, if the simulation is not ended, the network falls back into the stored states without showing any false memories. The lower the value of β , the longer this dynamic can proceed without encountering spurious states. The free phase has no utility in this scenario, the attractors are well separated, and the disruption of the energy landscape by adaptation is minimal. For critical loads (see Figure 4 middle), the retrieval process becomes more challenging. The attractors exhibit reduced separation, and the network struggles to converge to the stored states once inhibition is applied. In this scenario, the free phase demonstrates its utility. At the end of the biased phase, during the second iteration of

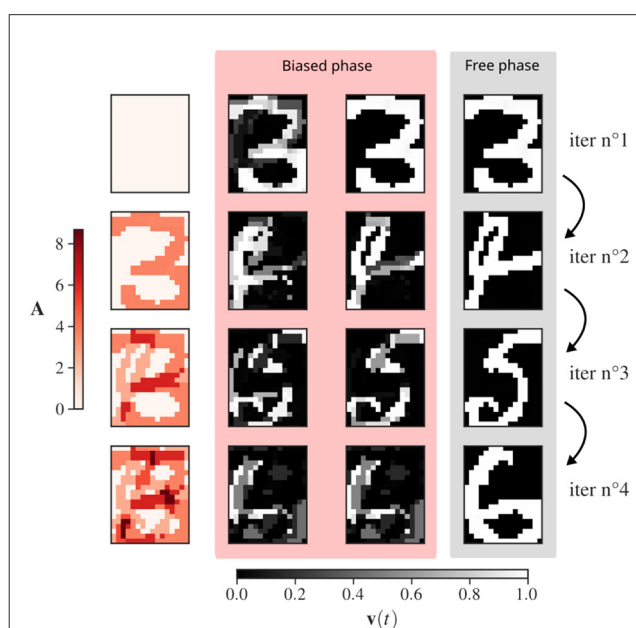


FIGURE 3

Recovery of three binary-patterns representations of handwritten digits from the MNIST dataset, “3”, “4”, “6”, “5”. The network is of size 20×16 as each neuron codes for a pixel. **Left** (in red): Evolution of adaptation of each neuron A_i . **Right** (blue and yellow): snapshots of the evolution of the rates $v_i(t)$ for each unit of the network. Each row corresponds to the start of a new memory retrieval in the CHN. The free phase corresponds to the removal of the inhibitory drive biasing the activity of each neurons as described in Algorithm 3. The panels illustrate how the biased phase “orients” the evolution, and then, the free phase provides the precise convergence to a stored pattern. An exaggerated large inhibitory coefficient $\beta = 2$ is used to illustrate the stable state deformation at the end of the biased phase, highlighting the need of the free phase.

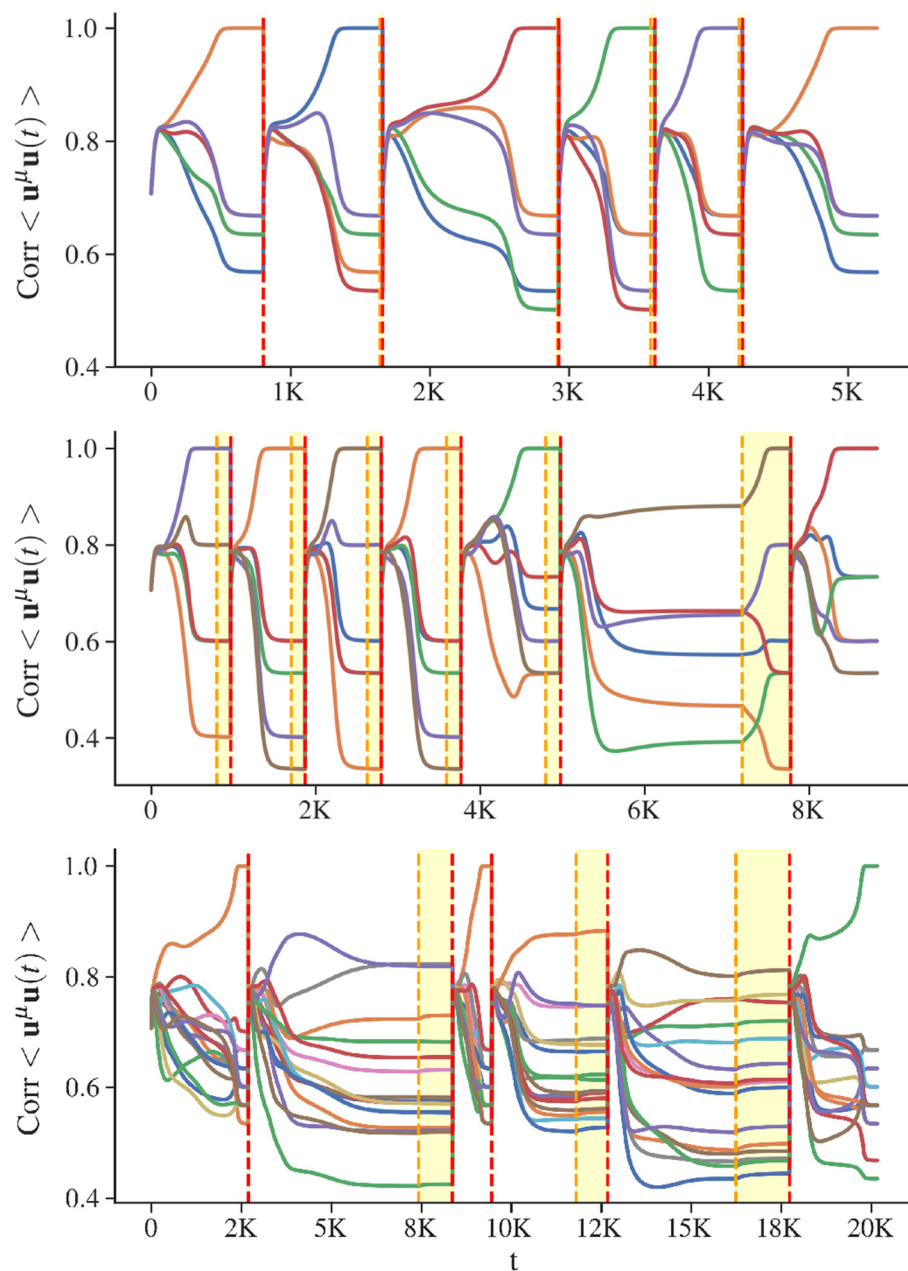


FIGURE 4

Visualization of AR (Algorithm 3). The x-axis shows the number of time steps for simulations of the CHN, with all query iterations of AR concatenated. The y-axis shows the correlation between the state of the network and any stored state, with each color corresponding to a specific memory. Vertical red dashed lines separate successive memory retrieval sequences. At the beginning of a memory recall adaptation A is updated and the initial state is the neutral one. The yellow dashed lines indicate the end of the biased phase. Between yellow and red lines is the free phase, during which self-inhibition is removed, to converge precisely into a stored state. Considered patterns have minimal correlation as they are generated from random binary vectors. **(Top)** Recovery dynamics for a low-load network: 5 patterns for a network of 60 units. **(Middle)** Recovery dynamics for a critical-load network: 5 stored patterns for a network of size 30. **(Bottom)** Recovery dynamics for a high-load network: 16 stored patterns for a network of size 60.

the AR, the network remains in a mixed state characterized by ambiguous correlations with multiple stored patterns. The free phase enables the network to resolve this ambiguity and ultimately converge to a stored state. At high loads (see Figure 4 bottom), the network successfully retrieves some stored states but mostly fall into spurious attractors. Under these conditions, even the free phase cannot rescue the recovery dynamics.

These autonomous recovery dynamics are reminiscent of memory replays observed in the central nervous system of mammals during quiescent states, such as sleep or rest. This process is believed to function as a consolidation mechanism that mitigates or modulates memory forgetting (Robins, 1995; Louie and Wilson, 2001; Peyrache et al., 2009; Fauth and Van Rossum, 2019; Tononi and Cirelli, 2014).

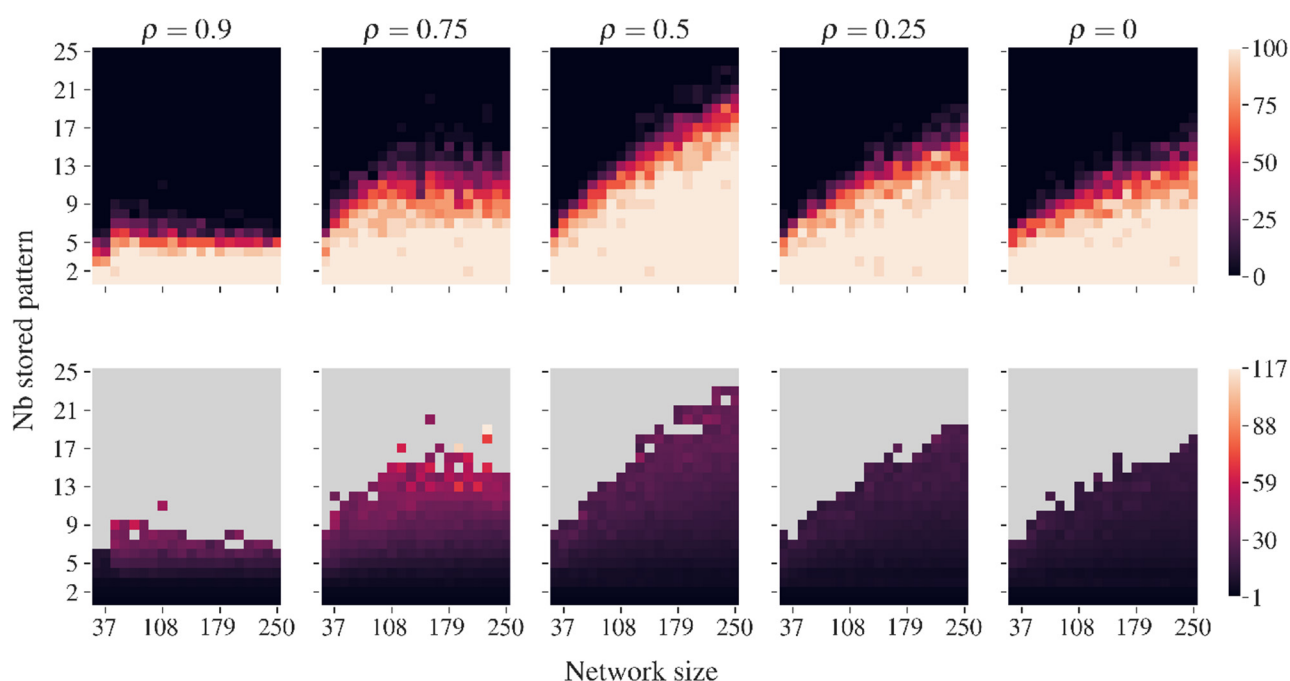


FIGURE 5

Recovery capacity of AR for various correlations, modulated by ρ . Data are averaged over 20 simulations with different pattern sets. Correlated patterns are generated using Algorithm 5 (Appendix 3). For networks of various sizes and numbers of stored patterns, we employ Algorithm 3 until all patterns are recovered or until a spurious state is found. **(Top row)** The percentage of 'full retrieval', i.e., simulation runs where no spurious state occurs before recovering all stored patterns. **(Bottom row)** Number of iterations required to recover all patterns. Recovery is best for $\rho = 0.5$, which denotes equal number of correlated and uncorrelated bits. For small ρ values with few correlated bits, performance worsens. For highly correlated sets, only very low loads are recovered without false memories.

The use of recurrent plastic inhibitory synapses has been tested and shows the same qualitative dynamics as that observed for networks with units that undergo adaptation. The results and the model are detailed in Appendix 4. As implementing adaptation requires less computation and parameters, the following work focuses only on the adaptation model.

3 Results

In this section, we evaluate the ability of our algorithm to retrieve correlated patterns from a given network. We will consider the retrieval of pattern sets with various amounts of correlation. Each set gets assigned a random binary pattern, the parent pattern. Each pattern of a given set is generated by randomly choosing and randomizing a fraction $(1 - \rho)$ of bits from the parent pattern as described in Algorithm 2 (Appendix 3). Therefore, a higher ρ induces more correlation, while a lower ρ results in less correlation. Networks with various loads will be tested. All retrieval dynamics are considered without free phases as it has been observed that, for the retrieval of random binary patterns and the use of small inhibitory potentiation values β , the free phase does not significantly improve retrieval performance.

Figure 5 illustrates, for various correlations, the loads for which AR allows systematic recovery of all stored patterns without external cues or memory lists. This property enables the continuous incorporation of new correlated memories, as

described in Algorithm 2. The retrieval improves with network size—larger networks can reliably retrieve more stored patterns without encountering false memories. We can observe the rather surprising feature that a moderate amount of correlation ($\rho \approx 0.5$) tends to improve pattern retrieval compared to highly correlated and minimally correlated pattern sets. In fact, this finding contrasts with traditional associative memory models, where correlation typically degrades performance (Fontanari and Theumann, 1990).

Our interpretation is that an intermediate amount of correlation helps the system to be driven in the “good” direction in early stages of the evolution, i.e., while still rather close to the neutral state, and the energy landscape is rather featureless. Once driven to a point of the state space proximal to all the stored patterns, the system can finish the convergence. As illustrated in Appendix 5, Appendix Figure 3, this push toward the good direction can be measured through the average correlation between the synaptic drive of each unit and the stored patterns.

As emerges from our simulations, to limit the appearance of false memories, the values of β must be relatively small compared to the target potential u_{target} . In our case, we found it convenient to keep β smaller than 0.1. By keeping the nudging of the convergence dynamic small, the emergence of false memories that might otherwise result from deformations in the energy landscape is reduced. Figure 6 indicates the existence of a trade-off: Smaller β values require more iterations to retrieve all patterns but provide greater stability, while larger values accelerate pattern retrieval at the cost of increasing the probability of encountering a spurious

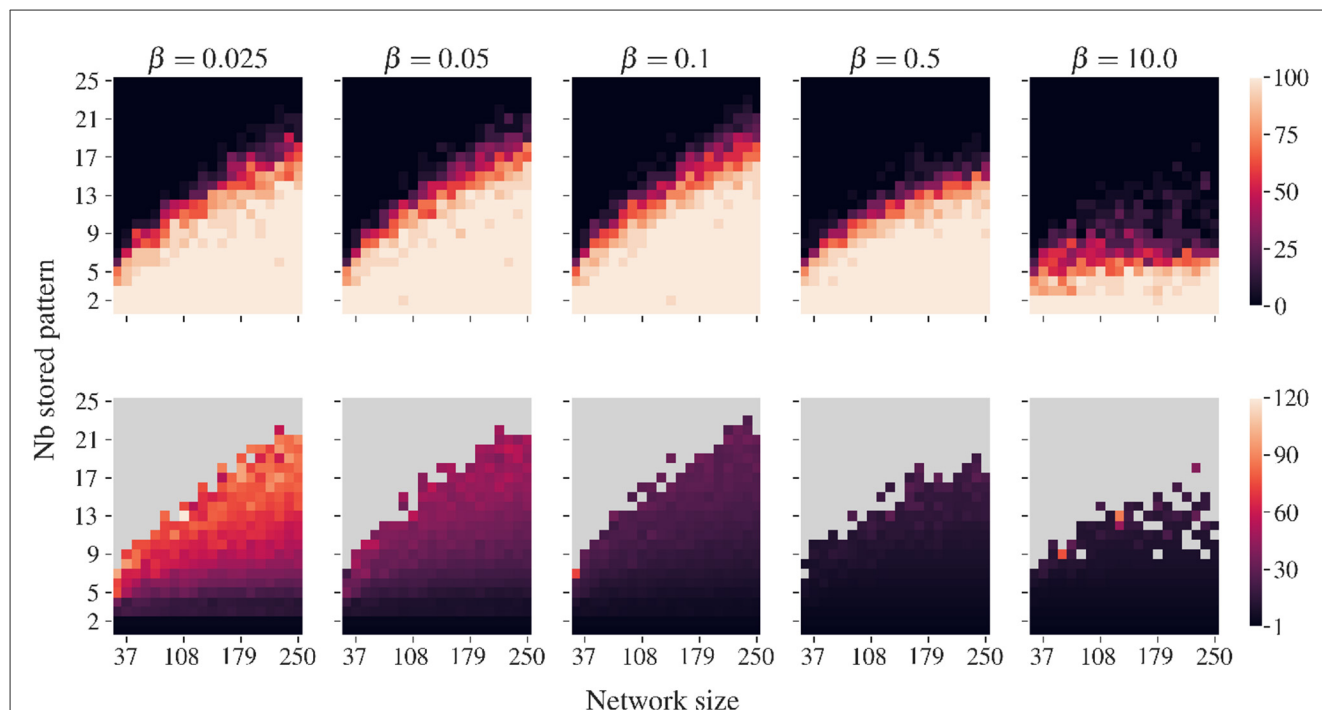


FIGURE 6

Similar to Figure 5 but for various β values. Correlated patterns are generated with a ρ of 0.5. Lower β values require more iterations to recover the complete set of stored patterns. Higher β values diminish the number of iteration needed but increase the probability of finding spurious state when the load is important.

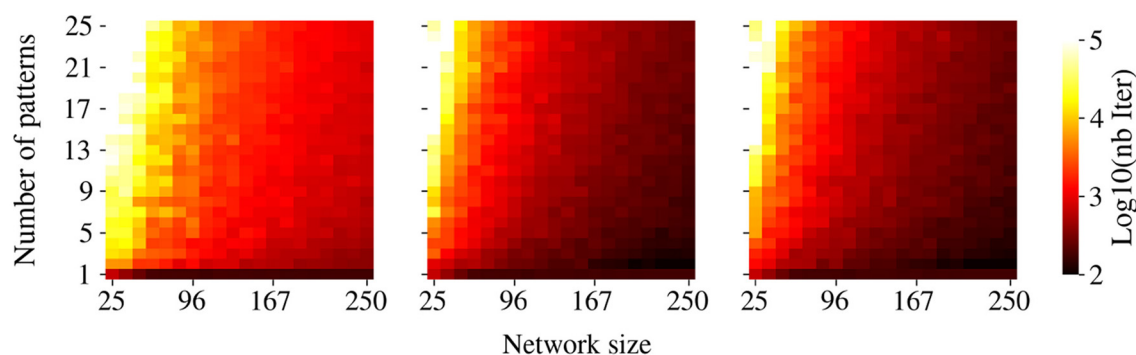


FIGURE 7

Convergence time of the GDA. Color intensity shows the number of iterations required to store a given number of patterns using GDA (Algorithm 1). Data are averaged over 20 simulations for various pattern sets.

state. Higher values of β than those considered in this study lead to catastrophic degradation of recovery dynamics for which no stored patterns are recovered. Lower values of β only lead to the need for more iterations when recovering the pattern set.

We shall now describe some qualitative information on the efficiency of our algorithm. As expected, the number of iterations required to successfully store patterns using Algorithm 1 increases with the memory load (Figure 7). Moreover, the higher the correlation between patterns, the higher the number of iterations needed for storage. Overall, our method requires significantly more iterations than previously documented for associative memory tasks in DHNs (Diederich and Oppen, 1987).

We can argue that this increase in computational cost may come from two factors. First, training a CHN through GDA (Algorithm 1) inherently requires more iterations than training DHNs. Second, we observed that a smaller convergence parameter ϵ in Algorithm 1, while computationally more demanding, yields superior retrieval performance. We hypothesize that this tighter convergence criterion induces stronger competition between pattern attractors at the neutral state. This competition results in enhanced network responsiveness to subtle modifications of attractor basins induced by W' during retrieval. These observations led to the adoption of a very small $\epsilon = 10^{-6}$, which demanded more iterations when performing the GDA.

4 Discussion

Our work introduces a biologically inspired mechanism for the continuous incorporation of correlated patterns in associative networks. CL is made possible through the autonomous recovery of all stored patterns during a retrieval phase. By systematically retrieving memories, the network can incorporate new patterns while mitigating the forgetting of the ones already stored. Autonomous retrieval is made possible by adaptation, avoiding the necessity of recalling patterns from an external list.

Previous work in computational neurosciences indicates that inhibitory circuits may play a critical role in the regulation of neural activity and plasticity (Vogels et al., 2011; Barron et al., 2016). Here, we demonstrate that inhibitory plasticity or SFA could be one of the key mechanisms that allow the sequential reactivation of memories observed during sleep and resting states (Wilson and McNaughton, 1994; Peyrache et al., 2009). A property of associative networks highlighted by our approach is that subtle changes in self-inhibition can drive substantial shifts in network dynamics without disrupting the fundamental structure of stored attractors. Inhibition, therefore, allows context-dependent activity of the network as observed in experimental setups (Kuchibhotla et al., 2017). Biological neural circuits might, therefore, employ similar mechanisms to navigate complex, correlated, memory spaces. Our results indicate that larger networks experience fewer spurious state visits, suggesting improved reliability with scale. However, understanding how these dynamics extend to networks of biologically relevant sizes would require further investigation.

Traditional associative memory models typically suffer from decreased capacity when storing correlated patterns (Amit et al., 1985b). However, our findings revealed the rather unexpected feature that moderate correlation levels actually improve pattern retrieval in the context of autonomous retrieval. We argue how this result may arise from the way that correlated structures influence the geometry of the attractor basins and the ensuing flow toward them. A more thorough theoretical analysis of this phenomenon could provide information on the factors that influence the robustness of recovery dynamics in both artificial and biological systems.

5 Conclusion

In this work, we demonstrate how adaptation can enable autonomous exploration of attractor landscapes in continuous Hopfield networks. Our key finding reveals that, under a critical load, inhibitory plasticity allows networks to systematically retrieve the entire set of stored memories, even for highly correlated sets. This property allowed us to propose and test an algorithmic scheme for continuous learning leveraging the ability of gradient descent to store correlated patterns. The capacity for self-directed pattern exploration, emerging from inhibitory modulation, offers insights for both biological memory consolidation and neuromorphic computing.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

PS: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Software, Visualization, Writing – original draft, Writing – review & editing. MR: Funding acquisition, Investigation, Methodology, Project administration, Resources, Supervision, Validation, Writing – review & editing.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. This project received financial support from the Ile-de-France region through the program DIM AI4IDF MR acknowledges support from the French ANR project MemAI ANR-23-CE30-0040-01 and the CNRS MITI program Osez2025.

Conflict of interest

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

Generative AI statement

The author(s) declare that no Gen AI was used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fncom.2025.1655701/full#supplementary-material>

References

- Amit, D. J., Gutfreund, H., and Sompolinsky, H. (1985a). Storing infinite numbers of patterns in a spin-glass model of neural networks. *Phys. Rev. Lett.* 55, 1530–1533. doi: 10.1103/PhysRevLett.55.1530
- Amit, D. J., Gutfreund, H., and Sompolinsky, H. (1985b). Spin-glass models of neural networks. *Phys. Rev. A* 32, 1007–1018. doi: 10.1103/PhysRevA.32.1007
- Barron, H. C., Vogels, T. P., Emir, U. E., Makin, T. R., O'Shea, J., Clare, S., et al. (2016). Unmasking latent inhibitory connections in human cortex to reveal dormant cortical memories. *Neuron* 90, 191–203. doi: 10.1016/j.neuron.2016.02.031
- D'amour, J., and Froemke, R. (2015). Inhibitory and excitatory spike-timing-dependent plasticity in the auditory cortex. *Neuron* 86, 514–528. doi: 10.1016/j.neuron.2015.03.014
- Diederich, S., and Oppen, M. (1987). Learning of correlated patterns in spin-glass networks by local learning rules. *Phys. Rev. Lett.* 58, 949–952. doi: 10.1103/PhysRevLett.58.949
- Fauth, M. J., and Van Rossum, M. C. (2019). Self-organized reactivation maintains and reinforces memories despite synaptic turnover. *Elife* 8:e43717. doi: 10.7554/eLife.43717
- Fontanari, J. F., and Theumann, W. K. (1990). On the storage of correlated patterns in Hopfield's model. *Journal de Physique* 51, 375–386. doi: 10.1051/jphys:01990005105037500
- Ha, G. E., and Cheong, E. (2017). Spike frequency adaptation in neurons of the central nervous system. *Exp. Neurobiol.* 26, 179–185. doi: 10.5607/en.2017.26.4.179
- Haxby, J. V., Gobbini, M. I., Furey, M. L., Ishai, A., Schouten, J. L., Pietrini, P., et al. (2001). Distributed and overlapping representations of faces and objects in ventral temporal cortex. *Science* 293, 2425–2430. doi: 10.1126/science.1063736
- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. Nat. Acad. Sci.* 79, 2554–2558. doi: 10.1073/pnas.79.8.2554
- Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Nat. Acad. Sci.* 81, 3088–3092. doi: 10.1073/pnas.81.10.3088
- Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proc. Nat. Acad. Sci.* 114, 3521–3526. doi: 10.1073/pnas.1611835114
- Kodangattil, J. N., Dacher, M., Authement, M. E., and Nugent, F. S. (2013). Spike timing-dependent plasticity at GABAergic synapses in the ventral tegmental area. *J. Physiol.* 591, 4699–4710. doi: 10.1113/jphysiol.2013.257873
- Kothapalli, V. (2023). Neural collapse: a review on modelling principles and generalization. *arXiv [preprint] arXiv:2206.04041*. doi: 10.48550/arXiv.2206.04041
- Kriegeskorte, N. (2008). Representational similarity analysis and connecting the branches of systems neuroscience. *Front. Syst. Neurosci.* 2:2008. doi: 10.3389/neuro.06.004.2008
- Kuchibhotla, K. V., Gill, J. V., Lindsay, G. W., Papadopoulos, E. S., and Field, R. E. (2017). Parallel processing by cortical inhibition enables context-dependent behavior. *Nat. Neurosci.* 20, 62–71. doi: 10.1038/nn.4436
- LeCun, Y., Cortes, C., and Burges, J. C. C. (1998). *The MNIST Database of Handwritten Digits*. Available online at: <http://yann.lecun.com/exdb/mnist/>
- Louie, K., and Wilson, M. A. (2001). Temporally structured replay of awake hippocampal ensemble activity during rapid eye movement sleep. *Neuron* 29, 145–156. doi: 10.1016/S0896-6273(01)00186-6
- McCallum, S. (1998). Catastrophic forgetting and the pseudorehearsal solution in hopfield-type networks. *Conn. Sci.* 10, 121–35. doi: 10.1080/095400998116530
- McCloskey, M., and Cohen, N. J. (1989). Catastrophic interference in connectionist networks: the sequential learning problem. In: *Psychology of Learning and Motivation*. London: Elsevier (1989). p. 109–165.
- Peron, S., and Gabbiani, F. (2009). Spike frequency adaptation mediates looming stimulus selectivity in a collision-detecting neuron. *Nat. Neurosci.* 12, 318–326. doi: 10.1038/nn.2259
- Peyrache, A., Khamassi, M., Benchenane, K., Wiener, S. I., and Battaglia, F. P. (2009). Replay of rule-learning related neural patterns in the prefrontal cortex during sleep. *Nat. Neurosci.* 12, 919–926. doi: 10.1038/nn.2337
- Quiroga, R. Q., Reddy, L., Kreiman, G., Koch, C., and Fried, I. (2005). Invariant visual representation by single neurons in the human brain. *Nature* 435, 1102–1107. doi: 10.1038/nature03687
- Robins, A. (1995). Catastrophic forgetting, rehearsal and pseudorehearsal. *Conn. Sci.* 7, 123–146. doi: 10.1080/0954009950039318
- Shen, Y., Dasgupta, S., and Navlakha, S. (2023). Reducing catastrophic forgetting with associative learning: a lesson from fruit flies. *Neural Comput.* 35, 1797–1819. doi: 10.1162/neco_a_01615
- Tononi, G., and Cirelli, C. (2014). Sleep and the price of plasticity: from synaptic and cellular homeostasis to memory consolidation and integration. *Neuron* 81, 12–34. doi: 10.1016/j.neuron.2013.12.025
- Vogels, T. P., Sprekeler, H., Zenke, F., Clopath, C., and Gerstner, W. (2011). Inhibitory plasticity balances excitation and inhibition in sensory pathways and memory networks. *Science* 334, 1569–1573. doi: 10.1126/science.1211095
- Wilson, M. A., and McNaughton, B. L. (1994). Reactivation of hippocampal ensemble memories during sleep. *Science* 265, 676–679. doi: 10.1126/science.8036517