Bio-inspired Continuous Learning in Associative Memory Networks

Paul Saighi

université PARIS-SACLAY

Bio-inspired Continuous Learning in Associative Memory Networks

Apprentissage continu bio-inspiré dans les réseaux à mémoire associative

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 564, Physique en Île-de-France (PIF)
Spécialité de doctorat : Physique
Graduate School : Physique. Référent : Faculté des sciences d'Orsay

Thèse préparée dans l'unité de recherche
Laboratoire de Physique des Solides (Université Paris-Saclay, CNRS),
sous la direction de **Marcelo ROZENBERG**, Directeur de Recherche au CNRS

Thèse soutenue à Paris-Saclay, le 1 Décembre 2025, par

Paul SAIGHI

Composition du jury

Membres du jury avec voix délibérative

Sabir JACQUIR Professeur, Université Paris-Saclay	Président
Carina CURTO Directrice de recherche, Brown University	Rapporteur & Examinatrice
Nicolas ROUGIER Directeur de recherche, Inria	Rapporteur & Examinateur
Alex ROXIN Directeur de recherche, Centre de Recerca Matemàtica	Examinateur
Sara A SOLLA Professeur, Northwestern University	Examinatrice
Samuel BOTTANI Professeur, Université Paris 7 Diderot	Examinateur

THESE DE DOCTORAT
NNT : 2025UPASP146

université PARIS-SACLAY | ÉCOLE DOCTORALE Physique en Île-de-France (PIF)

Titre : Apprentissage continu bio-inspiré dans les réseaux à mémoire associative
Mots-clés : Apprentissage continu, Réseaux de neurones, Réseaux de Hopfield, Mémoire associative, Matériel neuromorphique, Bio-inspiré

Résumé :

L'apprentissage continu, c'est-à-dire la capacité d'acquérir de nouveaux souvenirs ou compétences tout en préservant des informations cruciales déjà apprises, constitue une faculté fondamentale du cerveau. Pourtant, cette capacité demeure difficile à reproduire dans les réseaux de neurones artificiels, car ceux-ci souffrent d'oubli catastrophique, c'est-à-dire de la perte soudaine de connaissances antérieurement acquises lors de l'apprentissage de nouvelles données.

Cette thèse porte sur les mécanismes de l'apprentissage continu et les causes de l'oubli catastrophique dans les réseaux à mémoire associative (RMA), des modèles proposés comme analogues aux circuits hippocampiques en raison de leur architecture récurrente et de leur dynamique d'attracteurs. Plus précisément, l'apprentissage continu est étudié dans le cadre de la mémorisation séquentielle de souvenirs fortement corrélés. Ces souvenirs reposent sur des représentations neuronales qui se recouvrent, c'est-à-dire que les mêmes groupes de neurones participent à l'encodage de multiples souvenirs. Ce recouvrement est considéré comme un principe organisationnel clé du cerveau : il permet la généralisation et la mise en évidence de régularités entre expériences. Toutefois, dans les RMA, la mémorisation séquentielle de représentations neuronales qui se recouvrent induit généralement des phénomènes d'interférence entre souvenirs, pouvant conduire à l'oubli catastrophique.

Nous introduisons un mécanisme qui atténue l'oubli catastrophique dans les RMA en permettant la réactivation autonome des souvenirs. En intégrant un mécanisme de plasticité auto-inhibitrice dans des réseaux de Hopfield continus, nous montrons que le réseau peut explorer spontanément son paysage d'attracteurs, revisitant séquentiellement l'ensemble des souvenirs précédemment appris, sans intervention externe. Lorsqu'il est couplé à un algorithme de descente de gradient, ce mécanisme de réactivation permet la construction de configurations synaptiques capables de soutenir simultanément anciens et nouveaux souvenirs. En limitant les phénomènes d'interférence entre souvenirs, ce mécanisme prévient l'oubli catastrophique lors de l'incorporation séquentielle de données corrélées.

Notre approche respecte des contraintes biologiques essentielles : toutes les règles d'apprentissage sont locales, ne s'appuyant que sur l'information disponible à chaque synapse, et le système fonctionne de manière autonome, sans nécessiter l'accès externe à un catalogue de souvenirs déjà mémorisés.

Ces contributions visent à enrichir à la fois le domaine des neurosciences et celui de l'intelligence artificielle. Le mécanisme proposé a pour objectif d'améliorer notre compréhension des dynamiques de consolidation mnésique présentes dans le cerveau, tout en guidant le développement de systèmes d'apprentissage artificiel plus robustes, notamment dans le contexte des implémentations neuromorphiques où la localité des dynamiques est essentielle.

université PARIS-SACLAY | ÉCOLE DOCTORALE Physique en Île-de-France (PIF)

Title : Bio-inspired Continuous Learning in Associative Memory Networks
Keywords : Continuous learning, Neural networks, Hopfield Networks, Associative memory, Neuromorphic Hardware, Bio-inspired

Abstract :

Continuous learning, the ability to acquire new memories and skills while preserving previously learned information, is a fundamental capability of the brain. However, replicating this capability in artificial neural networks remains problematic, as they typically suffer from catastrophic forgetting (CF), the abrupt and uncontrolled loss of previously stored information when learning new data.

This thesis investigates continuous learning and catastrophic forgetting dynamics in associative memory networks (AMNs), computational models proposed as analogues of hippocampal circuits due to their recurrent architecture and attractor dynamics. Specifically, we study continuous learning in the challenging scenario of sequential storage of highly correlated memory patterns. These patterns rely on overlapping neural representations, where the same groups of neurons participate in encoding multiple memories. Such overlapping representation is considered a fundamental organizational principle of the brain, enabling generalization and the extraction of regularities across experiences. However, in AMNs, the sequential storage of overlapping memory representations typically induces interferences between stored memories, ultimately resulting in catastrophic forgetting.

We introduce a novel mechanism that mitigates catastrophic forgetting in AMNs by enabling autonomous rehearsal of stored patterns. By incorporating biologically inspired plastic self-inhibition into continuous Hopfield networks, we demonstrate that the network can spontaneously explore its attractor landscape, sequentially retrieving all previously stored memories without external intervention. Using a gradient descent algorithm, the rehearsal mechanism allows the network to construct synaptic configurations that simultaneously support both old and new memories. By limiting interferences between stored memories, this mechanism prevents catastrophic forgetting during the sequential incorporation of new correlated patterns.

Our approach satisfies key biological constraints : all learning rules are local, relying only on information available at individual synapses, and the system operates autonomously without requiring an external catalog of previously stored patterns. The resulting framework scales effectively with network size and successfully handles patterns with varying degrees of correlation.

This work aims to contribute to both neuroscience and artificial intelligence by proposing a biologically plausible mechanism for continuous learning that could inform our understanding of memory consolidation in the brain and guide the development of more robust artificial learning systems, particularly for neuromorphic implementations where local computation is essential.

À ma famille.

Acknowledgment

First and foremost, I am deeply grateful to my supervisor, Prof. Marcelo Rozenberg, for his guidance and support over these three years. Through his remarkable open-mindedness, he allowed me to explore research directions that, while related, departed from the original subject of my thesis. Our discussions were always stimulating and constructive, and he knew how to channel my hyper-exploratory tendencies into focused, productive work. His encouragement gave me the confidence to pursue my ideas and ultimately publish my first paper.

I would also like to thank Adrien d'Hollande. Our discussions and his attentive listening provided invaluable moral support during this final year. I am confident that his curiosity and open-mindedness will enable him to achieve his career and personal goals, whatever they may be.

My sincere thanks go to Véronique Bellu. Her assistance with administrative tasks and her warmth have been a great support for someone who, like myself, suffers from an irrational fear of paperwork.

Finally, I thank my family and friends, without whom I would never have reached this point in my journey.

Table des matières

Liste des acronymes

AMN — Associative Memory Network

ANN — Artificial Neural Network

AR — Autonomous Retrieval

CAM — Content-Addressable Memory

CF — Catastrophic Forgetting

CHN — Continuous Hopfield Network

CI — Continuous Incorporation

CL — Continuous Learning

DHN — Discrete Hopfield Network

GDA — Gradient Descent Algorithm

HN — Hopfield Network

MNIST — Modified National Institute of Standards and Technology (dataset)

MR — Memory Replay

NN — Neural Network

1 - Introduction

Humans demonstrate a remarkable capacity for lifelong learning. An individual's memory corpus is continually updated and refined to support its adaptation to the new challenges arising throughout its entire life. Importantly, despite this continuous adaptation to new contexts, certain skills are maintained without being actively recruited over extended periods of time.

The subjective experience of suddenly recalling long-dormant memories or effortlessly performing skills unused for years can catch us by surprise. Consider someone who learned to ride a bicycle as a child, but has not cycled for decades. Upon mounting a bike again, after a short period of adjustment, they may find themselves pedaling with remarkable stability and control, their body automatically maintaining balance and coordinating movements despite years without practice. This unexpected preservation and retrieval of motor coordination can be genuinely startling. This phenomenon extends beyond motor skills. Over the years, numerous experiences shape a person's repertoire of skills and memories. Countless pieces of information are forgotten due to lack of conscious rehearsal, and most daily cognition is related to current preoccupations. However, certain memories, such as the face of a relative not seen for decades, the scent of a childhood bedroom, or the route from an old home to the nearest subway station, appear to be preserved against the erosion of time.

This ability, observed in humans, to acquire new knowledge over time while retaining crucial information previously learned is known as continuous learning (CL). From an evolutionary perspective, being able to perform CL represented a survival advantage for our ancestors. The ability to adapt to new environmental challenges without forgetting vital long-term memories, such as the locations of reliable water sources, safe shelter sites, and the skills required to hunt particular prey, would have directly influenced survival rates and reproductive success. Consequently, natural selection has shaped and refined the mechanisms that enable CL across diverse environments and tasks.

In mammals, such remarkable behavioral adaptability is considered to be a function of the nervous system and, more specifically, the brain. This organ allows us to acquire complex skills and retain information. In humans, the brain represents a significant metabolic investment, consuming approximately 20% of the body's energy despite only accounting for about 2% of total body mass. This disproportionate allocation of energy, along with the extended developmental period required for the human brain to reach maturity, suggests the critical importance of the learning and processing capabilities it provides.

The ability of humans to perform CL is even more impressive when we consider that current state-of-the-art machine learning algorithms based on artificial neural networks (ANNs) typically struggle with CL tasks. ANNs involved in large language models for text generation and diffusion models for image or video synthesis are typically trained in a single large-scale training run (batch training) on all available data and then deployed with minimal further updating. Attempts to continuously adapt their skill set through the learning of new data induce a phenomenon called catastrophic forgetting (CF), in which previously learned knowledge is forgotten, causing severe performance degradation.

Consequently, a substantial body of artificial intelligence (AI) research is devoted to solving the problem of CF. The focus of this thesis is a class of solutions to CF inspired by the phenomenon of memory replay (MR) observed in the brain. MRs are well-documented events during which the brain revisits previously encoded memories during spontaneous activity. While MRs occur primarily during sleep, they also occur during quiet waking periods, such as when someone is resting on a couch. Numerous theoretical frameworks propose that MRs protect crucial knowledge from degradation induced by new learning. This protection is thought to arise from the unconscious rehearsal of important information. To illustrate this concept, we can return to our earlier example of bicycle riding. The theory proposes that cycling skills persist despite extended periods without practice because memories related to bicycling are periodically reactivated during sleep. Just as conscious rehearsal helps us remember material before an exam, unconscious rehearsal induced by MRs may underlie the maintenance of our long-term memories.

Although there is substantial empirical evidence that MR can facilitate learning and mitigate CF in ANNs, relatively few studies investigate how ANNs might generate replays through their own dynamics. From a computational neuroscience perspective, this question is central, as the brain has to produce replays via its intrinsic activity (e.g., autonomously during sleep and quiet wakefulness). To address this gap, this thesis proposes a model that enables autonomous MR in associative memory networks (AMNs), a class of ANN. The proposed model demonstrates how replay experiences can be generated using only the network's internal dynamics, addressing the fundamental constraint that the brain faces. In the proposed model, spontaneous MRs generated by the intrinsic dynamics of the network enable memory rehearsal, thus mitigating CF.

The work presented here serves a dual purpose. First, we develop concrete solutions for generating and leveraging MR in ANNs to tackle the pervasive challenges of CL. Second, by constraining our approach to biologically plausible dynamics, we propose a mechanistic hypothesis for how the brain generates memory replay during periods of spontaneous activity.

1.1 . Thesis organization

The primary contribution of this PhD is presented in Chapter 4, which introduces an AMN model capable of generating MRs through its internal dynamics. This work demonstrates that MRs effectively mitigate CF in such models. We particularly focus on enabling the storage and replay of strongly correlated memories for two key reasons :

- Both the brain and ANNs rely on correlated memory representations to perform core functions such as knowledge generalization and feature detection.
- The storage of such correlated memories typically induces stronger CF dynamics than the storage of uncorrelated representations.

Throughout this work, we have also prioritized the locality of neuronal dynamics, including synaptic plasticity. Respecting this constraint not only enhances the biological plausibility of our model, but also increases its potential for implementation in neuromorphic hardware.

To provide the necessary context for this contribution, Chapter 2 reviews the theoretical and empirical background essential for understanding Chapter 4. The chapter begins with Section 2.1, which examines the architecture, dynamics, and functions of the nervous system and explores how they relate to the ANN models considered in this work. Building on this biological foundation, Subsection 2.2.3 discusses the locality constraint and its importance for biological plausibility, while Subsection 2.2.4 examines how these principles translate to neuromorphic hardware implementations and their advantages over traditional hardware.

Section 2.3 then presents AMNs, which is the class of ANNs that forms the core object of this thesis. Their computational properties and advantages over traditional memory storage solutions are explored in detail through Subsection 2.3.1 and Subsection 2.3.2, respectively. Subsection 2.3.3 further examines how AMNs relate to neural circuits present in the brain, while Subsection 2.3.4 introduces the paradigmatic model of AMNs that constitutes the main subject of study in this thesis : Hopfield networks.

Section 2.4 reviews how memory rehearsal methods mitigate CF to enable CL in ANNs, with a focus on biologically plausible models. Section 2.5 broadens the discussion to other types of bio-plausible approaches to CL.

Having established both the biological and theoretical foundations, Chapter 3 addresses the intertwined challenges of storing correlated memories and allowing CL in AMNs. This analysis sets the stage for understanding the specific problems that our model in Chapter 4 aims to solve.

Finally, the Conclusion reflects on the limitations of this work and outlines promising avenues for future improvement and development.

2 - Biological and Computational Foundations

This chapter reviews the theoretical and empirical background underpinning the work presented in this thesis.

In Sec. 2.1, we begin by examining the structure, basic dynamics, and functions of the nervous system, which provides the inspirational basis for the architecture of associative memory networks (AMN).

In Sec. 2.2, we explore the core computational paradigm of neural circuits. This paradigm can be decomposed into distinct principles such as distributivity, overlapping memory representation, and the locality of processing dynamics. These principles define the design constraints for biologically plausible learning algorithms. They have guided our design choices and conditioned the models we present throughout this thesis.

In Sec. 2.3, We properly introduce AMN, with particular emphasis on Hopfield networks that serve as the primary model investigated in this work. These networks exemplify how biological principles translate into computational architectures capable of content-addressable memory.

In Sec. 2.4, We continue with the examination of memory replays and rehearsal dynamics observed in biological systems and their implementation in artificial networks, establishing the conceptual framework for the autonomous rehearsal approach developed in Chap. 4.

In Sec. 2.5, the chapter concludes with a review of other methods than rehearsal used to achieve bio-inspired continuous learning (CL) in neural network models. These methods lie slightly outside the scope of this thesis, but constitute other serious candidates to explain how the brain could perform CL and should therefore be mentioned. They do not necessarily constitute competing theories, but can also be interpreted as complementary mechanisms that the brain could use in synergy with memory replays.

2.1 . The Nervous System

2.1.1 . Structure and Function

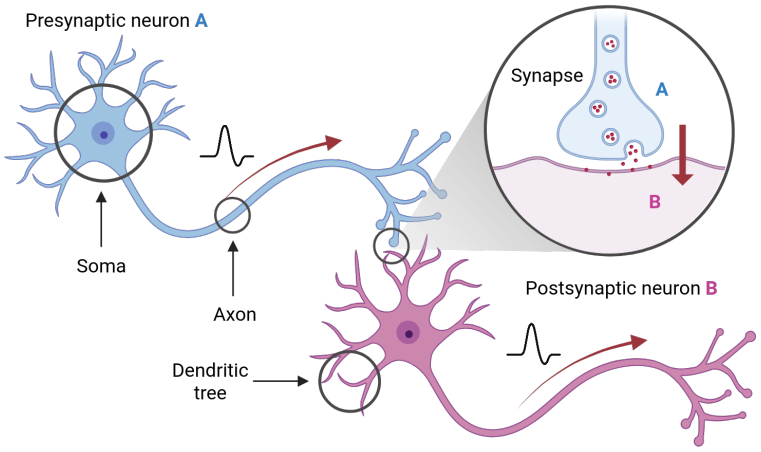

Figure 2.1 – **Two neurons making synapse.** The red arrows indicate the direction in which the signal (action potential) is transmitted. We can write $A \rightarrow B$ to make explicit that it is neuron A which modulates the activity of neuron B. A zoom is made on the synaptic site showing the release of neurotransmitters (red dots) that relay the signal from neuron A to neuron B.

The nervous system is a functional collection of cells, tissues, and organs that regulates the body's responses to internal and external stimuli. The cells of the nervous system can be divided into two main categories : neuronal cells (neurons) and support cells called glial cells. This discussion will focus exclusively on neurons that are specialized in the generation and conduction of electrical and chemical signals. Neurons exhibit a characteristic branching morphology. The dendritic tree serves as the input structure, receiving signals from other neurons through specialized connections called synapses, with a single neuron potentially contacted by up to 100,000 synapses (Hoxha et al., 2016). These incoming signals converge to the soma (cell body) of the neuron, where they are integrated and transformed into output signals. The output signal is then transmitted through the axon, where it is carried to further synapses which influence the dendritic trees from other neurons (Fig. 2.1).

This signal propagates as an action potential (AP), a rapid change in electrical potential that travels from the soma along the axon. When the action potential reaches the axon terminal, the synapse transmits the signal through neurotransmitter release (Fig. 2.1). These neurotransmitters influence the electrical potential of the contacted dendrite, effectively relaying the signal from the presynaptic neuron (the signal-sending neuron) to the postsynaptic neuron (the signal-receiving neuron).

The key takeaway is that neurons can transmit signals through their axons and dendrites, sometimes across considerable distances throughout the brain and body. This allows them to influence the activity of other neurons across the entire nervous system.

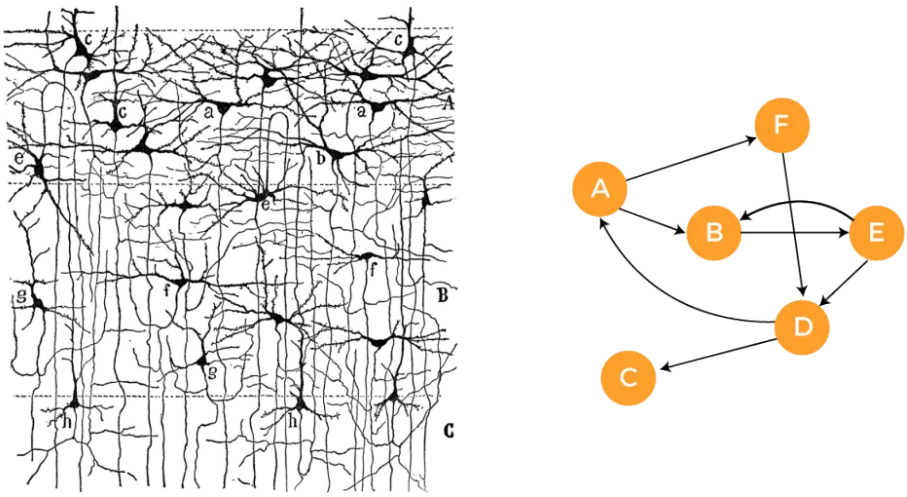

Figure 2.2 – **The nervous system as a directed graph. (left)** Drawing from Santiago Ramón y Cajal's second lecture at Clark University, 1899. The drawing illustrates neurons in layers 4, 5, and 6 of the visual cortex of a human infant. **(right)** A directed graph as the chosen abstraction to model the functional interactions between neurons in a network such as the one depicted by Cajal. The two networks do not map onto each other, their representation side by side is only for illustrative purposes.

Neurons are often considered to function as computational nodes that integrate signals from the numerous inputs received through synapses. In this view, synaptic connections establish the edges that connect these computational nodes throughout the system. Rather than following a clear hierarchical arrangement, the nervous system exhibits a complex distributed connectivity pattern that motivates its modeling as a network or a graph (Fig. 2.2). As synapses conduct the signal only in one direction, from the presynaptic cell to the postsynaptic cell, this graph is therefore directed.

The nervous system performs an extraordinarily broad range of functions, from coordinating voluntary and involuntary movements, regulating physiological processes through hormonal signaling and autonomic control, to supporting higher cognitive functions such as emotional processing, executive planning, problem solving, language comprehension, and conscious awareness. This remarkable functional diversity is often considered to emerge from the coordinated activity of billions of interconnected neurons working in parallel and hierarchical processing streams (Felleman & Van Essen, 1991; Zeki, 2015; Ungerleider, 1994).

In this thesis, we will focus specifically on the formation and maintenance of memories, a fundamental phenomenon that underlies learning and adaptive behavior.

Memory can be defined as the faculty by which the nervous system encodes, stores, and retrieves information, a process that fundamentally represents the persistence of learning over time. From a computational perspective, memory is an abstraction that encompasses various mechanisms through which past experiences modify future behavior and cognition. This includes not only conscious recollections (declarative memory) but also unconscious changes in performance (procedural memory), emotional associations, and perceptual priming effects (Squire & Dede, 2015).

The formation and maintenance of appropriate memories is crucial for survival, as it enables organisms to learn from past experiences, recognize patterns in their environment, anticipate future events, and adapt their behavior accordingly. Without memory, an organism would be trapped in a perpetual present, unable to benefit from learning derived from previous encounters with predators, food sources, or environmental hazards. Memory systems allow for the accumulation of knowledge across timescales, from rapid single-trial learning of dangerous situations to gradual skill acquisition through repeated practice. This adaptive value has driven the evolution of sophisticated memory systems across species, from simple habituation in invertebrates to the complex episodic and semantic memory systems found in humans (Rankin et al., 2009; Greenberg & Verfaeilli, 2010). The formation of memories, along with learning dynamics in general, is considered to be primarily supported by synaptic plasticity, as will be discussed in the next section.

AMNs, which are the models discussed in this thesis (Sec. 2.3), omit most of the biophysical details of neural dynamics. Mechanisms such as action potential generation, their propagation through the axon, the release of neurotransmitters and receptor binding, and the molecular cascades underlying signal transmission are not taken into account. Instead, these models preserve what can be considered the core constituents of the nervous system's structure and behavior : the network topology and the ability of nodes (neurons) to influence the state of other nodes through their connecting directed edges (synapses). This abstraction provides a tractable framework for investigating how network topology and dynamics give rise to memory formation and retrieval, without requiring detailed simulation of every biophysical process.

Inhibition and excitation

Neurons in the nervous system are broadly divided into two categories based on their effect on postsynaptic targets : excitatory neurons and inhibitory neurons. When an excitatory neuron is activated, it releases excitatory neurotransmitters into the synaptic cleft. These neurotransmitters bind to postsynaptic receptors and increase the probability that the postsynaptic neuron emits action potentials. Conversely, when an inhibitory neuron is activated, it releases inhibitory neurotransmitters into the synaptic cleft, decreasing the probability that the postsynaptic neuron emits action potentials. These two categories therefore appear to have a complementary role, excitation up-regulates activity, and inhibition down-regulates it.

In the mammalian cortex, excitatory neurons constitute approximately 70–80% of the neuronal population (Lodato & Arlotta, 2015). They participate in both local circuit computations and long-range signal transmission. Projections connecting distant cortical areas and those extending to subcortical and peripheral targets are predominantly excitatory (Lodato & Arlotta, 2015).

Inhibitory neurons, which represent a smaller fraction of cortical neurons (on the order of 10–20%), predominantly project locally toward surrounding neurons and are therefore commonly referred to as interneurons (Lodato & Arlotta, 2015; Tremblay et al., 2016). Although long-range inhibitory projections of cortical origin have been identified, they remain a minority compared to excitatory long-range pathways (Tamamaki & Tomioka, 2010; Caputi et al., 2013).

Inhibitory neurons play a central regulatory role in cortical and hippocampal circuits, maintaining the balance between excitation and inhibition necessary for stable network operation (Hernández-Frausto et al., 2023; Isaacson & Scanziani, 2011). Disruptions of this balance can lead to pathological states such as epileptic seizures, in which unchecked excitatory activity propagates uncontrollably through neural circuits (Isaacson & Scanziani, 2011). Beyond homeostatic regulation, inhibitory neurons also shape the selectivity and temporal precision of excitatory responses, gate information flow between brain areas, and generate neural oscillations that coordinate distributed computations (Kepecs & Fishell, 2014; Roux & Buzsáki, 2015; Isaacson & Scanziani, 2011). They have further been shown to participate in memory-related processes such as the formation of engrams and the regulation of engram size (Giorgi & Marinelli, 2021; Morrison et al., 2016; Barron et al., 2017; Tomé et al., 2024; Hou et al., 2024). Distributed computations, engrams, and memory representations, which are mentioned here for context, are formally introduced in Sec.2.2 and Sec.2.1.2.

In AMNs (Sec. 2.3), the main class of models considered in this thesis, the strict distinction between excitation and inhibition is abstracted away. In these networks, neurons are not defined as exclusively excitatory or inhibitory; rather, a single neuron can influence its targets through both inhibitory and

excitatory synapses. Consequently, we say that these models do not respect Dale's law. This principle, attributed to Henry Hallett Dale, states that a neuron releases the same set of transmitters at all its synapses, which in many circuits implies that it acts predominantly as either excitatory or inhibitory

This modeling freedom is often justified by a disynaptic (auxiliary-interneuron) interpretation : a circuit in which a single unit can have both excitatory and inhibitory effects can be implemented by a functionally equivalent, albeit more complex, circuit that respects Dale's law. In this framework, an effective inhibitory influence originating from an excitatory unit is treated as a modeling shorthand for an indirect pathway, where the excitatory neuron recruits an intermediate inhibitory interneuron that in turn suppresses the target (Parisien et al., 2008; Abbott & Dayan, 2005). However, it is important to acknowledge counterarguments emphasizing that imposing Dale's law (i.e., sign-constrained connectivity) can substantially affect recurrent network dynamics and should therefore be treated as a priority when biological faithfulness is sought (Song et al., 2016).

2.1.2 . Plasticity and learning

Plasticity refers to the ability of the nervous system to modify its structure and behavior in response to experience and homeostatic needs (Citri & Malenka, 2008). The structural changes considered as forms of plasticity are diverse and occur at multiple levels of organization. At the synaptic level, changes include alterations in the number and sensitivity of neurotransmitter receptors, modifications of the postsynaptic density that anchors these receptors, and changes in the probability of neurotransmitter release from presynaptic terminals (Malinow & Malenka, 2002; Sheng & Kim, 2011; Zucker & Regehr, 2002). Beyond individual synapses, plasticity is manifested through synaptogenesis (formation of new connections between neurons), synaptic pruning (elimination of existing ones), morphological changes in dendritic spines, and neurogenesis in specific brain regions (Fu & Zuo, 2011; Ming & Song, 2011). These mechanisms operate on timescales from milliseconds to years, working in concert to shape neural circuits.

Plasticity serves multiple roles : It underlies developmental processes that refine neural circuits based on sensory experience, supports homeostatic regulation that maintains stable network activity despite perturbations, enables recovery and compensation following neural injury, and allows continuous adaptation of behavior to changing environmental demands (Turrigiano, 2012; Tetzlaff et al., 2011; Morris et al., 1986; Takeuchi et al., 2014). In this work, we will focus on framing synaptic plasticity as the primary mechanism allowing the formation of memories.

In most cases, synaptic plasticity is considered to involve modulating the strength of connections between neurons, that is, adjusting how effectively

a presynaptic neuron can influence the activity of its postsynaptic partner. A strong synapse allows the presynaptic neuron to exert substantial influence on the postsynaptic neuron's activity (its output signal), potentially driving significant changes in its firing rate, the frequency at which it produces action potentials. Conversely, a weak synapse produces only modest effects on postsynaptic activity, contributing less to variations in the target neuron's output. The relative strength of a synapse is sometimes referred to as its weight, with a large weight corresponding to strong synapses and a small weight corresponding to weak ones. The strengthening of synapses is also called potentiation, and the weakening of synapses, depression (Citri & Malenka, 2008).

Through experience-dependent potentiation and depression, neural circuits can encode information and adapt their computational properties (Takeuchi et al., 2014). Synaptic plasticity is considered to be the main mechanism underlying the formation and refinement of memories (Josselyn & Tonegawa, 2020).

An engram refers to the enduring modifications in neural circuits that constitute the physical substrate of a memory. When Richard Semon first introduced the concept in 1904, the physical basis of such memory traces was entirely speculative, with little understanding of the biological mechanisms that could support the storage and retrieval of information in neural tissue (Semon, 1921). Subsequent research has shown that engrams are supported by distributed patterns of synaptic changes across neural networks, although additional processes such as intrinsic neuronal plasticity and synaptic pruning are also likely to contribute (Shim et al., 2018; Peter R., 1979). These modifications allow for the selective reactivation of neuronal ensembles when the organism encounters appropriate retrieval cues, thus resulting in the phenomenon of memory recall (Josselyn & Tonegawa, 2020).

Memory recall (or retrieval) therefore represents the process by which the engram is translated back into active neural patterns. When retrieval cues match aspects of the original encoding context, they trigger the reactivation of the modified synaptic pathways, reconstructing a pattern of neural activity that resembles the original experience. This reconstruction allows stored information to influence current behavior and cognition (Takeuchi et al., 2014).

For instance, the taste of a madeleine dipped in tea may serve as a powerful retrieval cue, reactivating engrams formed during childhood. The taste of the madeleine can reactivate neural circuits linked to the sensory qualities of a grandmother's kitchen, the feeling of Sunday mornings, or the atmosphere of a childhood home. This phenomenon demonstrates how a simple sensory cue can trigger the reconstruction of rich, multifaceted memories (Proust, 1913).

Compelling evidence has emerged for the link between synaptic plasticity and memory storage, as studies have shown that artificially manipulating the

strength of synapses can directly modulate memory expression. Nabavi et al. (2014) demonstrated this causal relationship using optogenetic protocols to directly strengthen or weaken synapses in the amygdala. After establishing a fear memory through conditioning, the researchers applied optical stimulation which induced depression in the previously potentiated synapses, causing the animals to lose their fear response, effectively erasing the memory. Remarkably, a subsequent protocol that re-potentiated the same synapses restored the fear memory, causing the animals to again display freezing behavior. Recent technological advances have allowed researchers to identify and manipulate specific engram cells, demonstrating that memories are encoded in distributed populations of neurons whose connectivity is modified by experience (Liu et al., 2012; Ryan et al., 2015). In the following section, we will discuss Hebbian plasticity, a major mechanism through which experience shapes connectivity in neural circuits leading to engram formation.

Hebbian plasticity and associativity

Donald Hebb's postulate, formulated in 1949, describes one of several mechanisms governing synaptic plasticity in neural networks and is often summarized as "cells that fire together, wire together" (Hebb, 1949). Hebbian plasticity represents a specific form of activity-dependent modification in which the temporal correlation between presynaptic and postsynaptic activity determines changes in synaptic strength.

Hebbian plasticity can take several forms and can be broadly characterized as follows : when two neurons increase their activity in close temporal proximity, the synaptic connection between them tends to potentiate. In certain cases, it is the firing rates (the frequency of action potentials events) that must be jointly elevated in both neurons ; in other cases, a single, precisely timed pair of action potentials from the presynaptic and postsynaptic neurons is sufficient to elicit potentiation. A wide variety of experimental observations support both views (Bliss & Lømo, 1973a; Markram et al., 1997).

Hebbian mechanisms are thought to underlie associative memory, the ability to link distinct stimuli or experiences that occur together in time and/or space (Antonov et al., 2001; Bliss & Collingridge, 1993). Associative memory allows organisms to predict relationships between events and forms the basis for conditioning. The canonical example is Pavlovian conditioning, where a neutral stimulus (such as a bell) becomes associated with a biologically significant stimulus (such as food) through repeated pairing. In Pavlov's original experiments, a dog is presented with food which naturally induces salivation. Pavlov then repeatedly rang a bell just before presenting food to the dog. Initially, the bell alone did not produce a salivation response. However, after multiple pairings of the bell sound followed by food presentation, the dog began to salivate upon hearing the bell alone, even when no food was pre-

sented. The previously neutral stimulus (the bell) had become a conditioned stimulus capable of eliciting a conditioned response (salivation), demonstrating that the nervous system had formed an associative link between the two stimuli.

At the physiological level, associative learning is considered to occur through Hebbian potentiation of synapses that connect neurons or groups of neurons that represent the paired stimuli. Specifically, neurons that encode the bell sound and neurons that encode the salivation response become active in close temporal succession during each pairing. According to Hebbian principles, this repeated coactivation strengthens the synaptic connections between these neural populations. As a result, after sufficient repetitions, activation of the 'bell' neurons alone becomes sufficient to modulate the activity of the 'salivation' neurons through the potentiated synaptic pathways, thereby triggering the salivation response even in the absence of actual food.

These principles of plasticity, associative learning, and engram formation have inspired computational approaches aiming to understand and replicate intelligent behavior. AMNs models discussed in Sec. 2.3 draw direct inspiration from these biological principles of associative memory induced by plasticity dynamics. Sec. 2.3.4 discusses how Hebbian plasticity is implemented in discrete Hopfield networks (DHN), a class of AMN. In computational models such as these, Hebbian plasticity is often abstracted as any type of plasticity that relies on the product of pre and postsynaptic activity.

In the next section, we show that these plasticity principles apply to both excitatory and inhibitory synapses, and that their functional outcomes can differ depending on the synaptic type.

Inhibitory plasticity and adaptation

In AMNs, inhibitory connections play a constitutive role in memory storage. Both excitatory and inhibitory synaptic weights shape the engrams such that silencing of specific units is an integral part of the stored memory. In the nervous system, a growing body of evidence supports a comparable role for inhibition : inhibitory engrams, formed through plasticity on GABAergic synapses, have been shown to complement excitatory engrams across several brain regions (Barron et al., 2017; Tomé et al., 2024; Hou et al., 2024).

Returning to the Pavlovian conditioning example, the conditioned response to the bell may involve not only excitatory potentiation but also the learned inhibition of specific neuronal populations and therefore a diminishing of their activity. Inhibitory plasticity is therefore a core component of the associative memory function discussed in the preceding section.

The model presented in this thesis (Chap.4) also relies on a second form of inhibition that serves a distinct function. Rather than participating in the encoding of individual engrams, this inhibition enables the network to sequentially explore its stored memories.

More specifically, the exploration of stored memories is induced by plastic self-inhibition, whereby a neuron or a group of neurons suppresses its own responsiveness in proportion to its recent activity. As a consequence, the network dynamics are redirected away from recently visited attractors and toward other stored memories upon subsequent memory retrieval attempts. Successive cycles of retrieval and self-inhibition therefore allow the network to explore its full memory repertoire.

From a biological standpoint, this self-inhibition can be interpreted either as Hebbian plasticity on inhibitory synapses, or as spike frequency adaptation (SFA), a widespread property of neurons in the mammalian central nervous system (Ha & Cheong, 2017).

Under sustained stimulation, a neuron that exhibits SFA progressively reduces its firing rate, typically through the accumulation of slow hyperpolarizing currents such as calcium-activated potassium currents (Benda & Herz, 2003). The use of adaptation to promote the sequential exploration of attractors has several precedents in the computational literature (Hopfield, 2010; Mi et al., 2014).

The formalization of this mechanism and its application to continuous learning are the subject of Chap.4.

2.1.3 . Conclusion

In summary, the core constituents of the nervous system are neurons connected through plastic synapses, forming complex networks that can be abstracted as directed graphs. Synaptic plasticity is considered to be the primary mechanism underlying engram formation. The formation of engrams in the nervous system enables organisms to learn from experience, that is, to form new memories.

As will be seen in Sec.2.3, this graph-based abstraction enables the development of models that capture the remarkable capacity of the nervous system to encode, store, and retrieve information.

As shown in Chap.4, additional mechanisms such as various forms of inhibitory plasticity or neuronal adaptation can be incorporated into these models to expand their computational repertoire and capture additional properties of the nervous system.

In the following section, we examine in greater detail how computation is organized across the neural substrate, focusing on the principles that shape information processing and storage throughout these networks.

2.2 . Nervous system computational paradigm

This section describes some of the core principles of how neural circuits organize and perform computation. These include distributivity, overlapping representations, and locality constraints. Understanding these organizational features shapes our view of both biological information processing and the development of artificial systems that emulate neural computations.

The effort to faithfully embrace the core organizational principles of the nervous system has also driven the creation of neuromorphic technologies that implement these principles in specialized hardware architectures. Discussion of neuromorphic implementations is relevant here, as the principal work presented in Chap. 4 focuses on developing solutions compatible with neuromorphic implementations.

2.2.1 . Distributivity of Neural Computations

The nervous system exhibits extensive distributed processing. Cognitive tasks, behavioral manifestations, and conscious experiences depend on patterns of activity distributed throughout the entire brain (Pessoa, 2014; Thomas Yeo et al., 2011). Critically, no well-defined subnetwork or localized area can be isolated as the single locus responsible for the computations underlying complex tasks such as recognizing a face, thinking of an abstract concept, or even bicycling (Haxby et al., 2001; Binder et al., 2009; Penhune & Steele, 2012).

This extensive distributivity supports what can be characterized as a decentralized computational paradigm, in which populations of neurons and their synaptic connections operate as computational units. These subnetworks perform specific computations in parallel and communicate their outputs to other regions, ultimately giving rise to coherent behavioral and cognitive modalities. This decentralized dynamic can be illustrated through an analogy with collective decision-making processes. Each neuron, neuronal population, or subnetwork (depending on the scale of analysis) functions analogously to an individual participant casting a vote based on the information available through its inputs. Once a voting unit has determined its decision (manifested as a change in neural activity), it conveys this information to other participants, influencing their subsequent decisions. The resulting behavioral or cognitive output therefore represents a concerted choice rather than the isolated expression of a single specialized subgroup. This phenomenon is enabled by the fact that each subnetwork encodes, within its synaptic weights and connectivity patterns, partial information necessary for the global computations that drive adaptive behavior.

Consider a dog that has learned to avoid a specific type of plant after experiencing its bitter taste and rough texture. When the dog encounters this plant again : visual information in occipital regions processes the distinctive

leaf shape and color ; olfactory processing detects the particular scent of the plant ; somatosensory areas may process the texture as the dog's nose briefly touches it. These converging sensory inputs induce an activity pattern supported by the engram formed during the original aversive experience. This engram corresponds therefore to the synaptic modifications that associated these specific sensory features with the unpleasant taste. This distributed pattern of activity triggers, through memory recall, the appropriate behavior : avoidance. All the computations taking place in the different subcircuits are potentially done in parallel and all individually weigh on the behavioral outcome.

The pattern of neural activity associated with memory recall, together with the engram that supports it, constitutes what is termed a memory representation. A memory representation can therefore be understood as the complete functional unit underlying a memory. One primary characteristic of these memory representations in the brain is that they overlap extensively. Overlap in this context means that individual neurons or subnetworks involved in one memory representation typically participate in numerous other representations corresponding to distinct experiences. The role of these overlapping representations is discussed in the following section.

2.2.2 . Overlapping memory representation

Overlapping memory representations constitute a fundamental organizational principle in neural systems. In the brain, an overlapping memory representation occurs when different memories are encoded using partially shared populations of neurons and synapses. Rather than dedicating separate neural circuits for each memory, the brain reuses subnetworks to participate in representing multiple memories (Haxby et al., 2001; Cai et al., 2016; Josselyn & Tonegawa, 2020).

This feature is not incidental but is considered to serve important computational functions. Overlapping representations allow for generalization between related experiences, thus supporting the extraction and consolidation of features and abstract concepts from specific episodic events (McClelland et al., 1995; Haxby et al., 2001; Kriegeskorte, 2008; Quiroga et al., 2005). This ability to generalize across similar experiences is critical for adaptive behavior, as it enables organisms to apply past learning to new, yet related, situations (Rigotti et al., 2013).

A compelling illustration of this principle comes from studies of face processing in the human brain. Research using high-dimensional face representations has demonstrated how face-selective regions encode different faces through overlapping neural patterns (Kriegeskorte et al., 2007; Chang & Tsao, 2017; Freiwald et al., 2009). In these studies, researchers found that individual neurons in face-selective areas increase their firing rate for multiple faces that

share similar features. For instance, the same neurons might show elevated activity for different faces that have similar eye shapes or nose structures. This means that the neural representation of one face partially overlaps with representations of other faces sharing those features, the greater the similarity between faces, the more neurons they share in their representation.

In this thesis, Chap. 3 and 4 focus on studying neural networks that store highly overlapping memory representations, consistent with observations in the nervous system.

2.2.3 . Locality

An aspect of neural computation that has been implicit in our discussion thus far, but deserves to be made explicit, is the fundamental constraint of locality in neural processing (Crick, 1989). Unlike distributivity and overlapping representations, which can be understood as organizational strategies, locality is an inescapable physical constraint that shapes how biological neural networks must operate.

Locality implies that the information required for neural or synaptic updates is accessible through local modulatory signals. In biological networks, local events such as changes in calcium concentration in both the axon terminal and the dendrite are considered the primary factors modulating synaptic strength (Neher & Sakaba, 2008; Malenka et al., 1992). Therefore, when a synapse undergoes modification during learning, it relies predominantly on neural activity patterns that can trigger these local events, such as presynaptic activity (the activity of the signal-sending neuron) and postsynaptic activity (the activity of the signal-receiving neuron). As will be introduced in Sec. 3.1.2, additional information is considered physically available at the synaptic site through local homeostatic regulatory mechanisms (Turrigiano et al., 1998; Turrigiano, 2012).

Other modulating factors also play an important role, notably volume-transmitted neuromodulators that convey information about broader network states (Özçete et al., 2024; Marder, 2012). These signals are critical for setting network states, modulating broad patterns of activity, and setting plasticity windows. However, the fine-grained specificity required to encode detailed information, such as memories of specific smells or faces, is generally considered to rely predominantly on the precise dynamics of individual synapses shaped by local activity-dependent and homeostatic processes (Engert & Bonhoeffer, 1999; Harvey & Svoboda, 2007). However, this view has been nuanced by work showing that neuromodulatory signaling can exhibit more spatial specificity than traditionally assumed, with, for instance, dopaminergic projections gating plasticity for targeted synaptic populations in a projection-specific manner (Lammel et al., 2011; Seamans & Yang, 2004).

This locality constraint has profound implications on the way biological

neural networks learn and adapt. It implies that learning rules depend primarily on locally available information, which is consistent with plasticity mechanisms such as Hebbian learning, where connections strengthen between neurons that fire together. In recognition of this fundamental constraint, the principal work presented in this thesis (Chap. 4) has been designed with most of the dynamics respecting locality, ensuring biological plausibility.

In the next section, we will introduce neuromorphic technologies, which take inspiration from the architecture and dynamics of the nervous system to perform their computations.

2.2.4 . Neuromorphic implementation

The computational principles embodied by the nervous system discussed above serve as design directives for neuromorphic technologies. Although detailed discussion of neuromorphic systems lies beyond the scope of this thesis, understanding their fundamental characteristics is essential, as the main work presented here (Chap. 4) emphasizes solutions compatible with neuromorphic implementations.

Neuromorphic technologies are specialized hardware architectures that emulate the computational principles, structure and dynamics of the nervous system (Mead, 1990; Indiveri & Liu, 2015; Schuman et al., 2017). Most of these systems implement distributed processing through arrays of artificial neurons and synapses, support parallel computation across multiple processing elements, and encode information in ways inspired by biological neural communication (Merolla et al., 2014; Davies et al., 2018; Furber, 2016; Benjamin et al., 2014). By directly instantiating neural dynamics in silicon, neuromorphic chips inherently respect the computational paradigm of biological neural networks, including distributed computation and overlapping representations.

Crucially, certain neuromorphic architectures typically enforce locality constraints, as no central processing unit (CPU), orchestrates computations across the network. Each computational element operates based primarily on locally available information from its directly connected neighbors, reflecting biological constraints (Boahen, 2000; Merolla et al., 2014; Kim et al., 2024; Wu & Rozenberg, 2024). Although certain hybrid architectures permit limited nonlocal operations for practical purposes (Furber, 2016; Kim et al., 2024), pure neuromorphic designs maintain strict locality to preserve biological fidelity and enable massively parallel processing. By ensuring that the model developed in this thesis relies exclusively on local dynamics, we therefore enhance not only its biological plausibility but also its potential for neuromorphic hardware implementation.

These technologies promise to better leverage bio-inspired network architectures by directly embedding computational constraints and modalities into hardware (Merolla et al., 2014; Davies et al., 2018; Schuman et al., 2017). For ins-

tance, neuromorphic chips are expected to consume significantly less energy and operate faster than traditional computers when running neural network algorithms, as they avoid emulating neural dynamics on classical CPU-based architectures. A primary source of these performance advantages is the potential to circumvent the von Neumann bottleneck discussed in Sec. 2.3.2, where data movement between memory and processing units creates computational inefficiencies.

2.2.5 . Conclusion

This section has examined some of the core principles of how neural circuits organize and perform computation : distributivity, which enables parallel processing across decentralized populations of neurons ; overlapping memory representations, which allow efficient storage and generalization ; and locality, which constrains neural and synaptic computations to rely primarily on locally available information. Together, these principles define a computational paradigm in which complex behaviors and cognitive functions emerge from the collective activity of interconnected neural populations operating under physical constraints.

These properties are not unique to biological systems but also characterize many models of artificial neural networks, including deep neural networks (DNNs) and AMNs. However, the degree to which artificial systems implement these principles varies considerably. Although distributivity and decentralization are embodied in virtually all neural network models, forming the foundation of their computational architecture, the locality constraint is respected only by certain models that prioritize biological plausibility or compatibility with neuromorphic implementations.

The following section introduces AMNs, which represent a class of models that effectively implements all these principles to perform associative memory, enabling the storage and retrieval of memories from partial cues while maintaining both biological plausibility and compatibility with neuromorphic hardware.

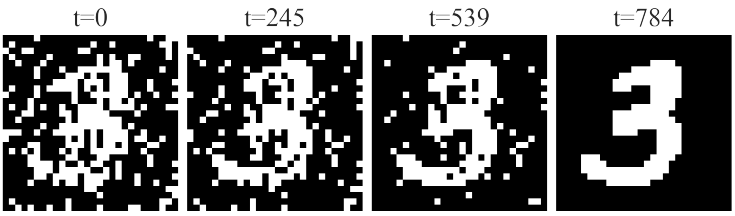

Figure 2.3 – **Recovery dynamics of a discrete Hopfield network (DHN).** Each pixel represents the state of a neuron in the network, with pixel color indicating the corresponding neuron's state. The network stores a memory pattern representing the digit "3". Starting from a noisy, partial cue of the stored pattern, the dynamics evolve over time steps t, where each step corresponds to a network state update. The network successfully recovers the complete stored pattern once the dynamics converge. DHNs are discussed in detail in Sec. 2.3.4.

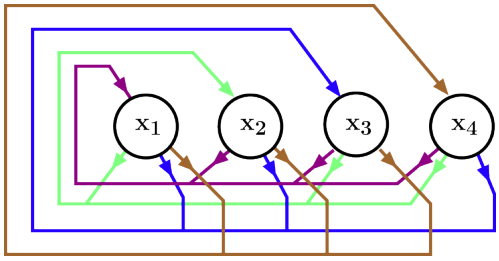

Figure 2.4 – **Typical architecture of an AMN.** Illustration of a Hopfield network (HN) architecture with multiple neurons interconnected through recurrent synapses (shown in color). Each neuron has an activity state x_i.

Having examined the structure of the nervous system and its computational paradigm, we now turn to associative memory networks (AMNs). AMNs are at the core of this thesis and provide an illustrative example of how drawing inspiration from the nervous system enables the development of functional models. AMNs constitute a class of recurrent neural network models in which memories correspond to activity patterns stored as attractor states within the network dynamics.

When presented with a partial or corrupted version of a stored memory (also called a query), they generate a trajectory in state space that converges toward the complete, uncorrupted version of the stored memory. It is this convergence dynamic that characterizes the stored states as attractors. We can say that the network state is "attracted" toward the stored pattern. This

process is known as pattern completion. However, this attraction does not operate from every possible initial state. The set of initial states from which the network dynamics converge to a given attractor is called its basin of attraction. Pattern completion occurs when the query falls within the basin of attraction of the target pattern. The wider the basin, the more degraded or incomplete a cue can be, while still leading to successful recovery.

Fig. 2.3 illustrates the typical recovery dynamics of an AMN, in this case, a discrete Hopfield network (DHN). The stored memory corresponds to the binarized image of the digit "3" from the MNIST dataset (LeCun et al., 1998). The network is initialized with a noisy cue that provides only partial information about the stored pattern. As the network state is iteratively updated, it progressively converges toward the stored representation of the digit "3". This cue therefore lies within the attraction basin of the stored pattern. DHNs are presented in detail in Sec. 2.3.4.

For most of the models presented in this thesis, memories are stored through modifications of synaptic connection strengths between neurons. The connectivity scheme is typically highly recurrent, as illustrated in Fig. 2.4. We say that a network is recurrent when neurons that send synapses to other groups of neurons, potentially influencing their activity state, can also receive synapses from these same neurons. Recurrence creates feedback loops that influence the network's dynamics over time and is, in the case of most AMNs, the cause of attractor states. This recurrent synaptic structure is therefore crucial for the storage and retrieval properties of the network (Hopfield, 1982).

Having the storage and retrieval of patterns (memories) supported by modifications of synaptic weights aligns with the concept of engrams as physical substrates for memory in the nervous system. This mapping between synaptic changes and stored content is supported by extensive engram literature demonstrating that reactivation of specific neuronal groups is both sufficient and often necessary for memory recall (Tonegawa et al., 2015; Liu et al., 2012).

2.3.1 . Content-Addressable Memory

One of the defining strengths of AMNs lies in their ability to serve as content-addressable memory (CAM). As shown in Fig. 2.3, the network is able to retrieve the memory using only partial information about the content of the memory itself. This property, also called pattern matching or associative memory, aligns well with how animals routinely recognize objects, situations, or other animals despite occlusion, varying lighting conditions, or other sources of noise.

A prototypical example is a prey spotting a predator in a bush : some information about the color or outlines of the predator is available, but most of the information must be reconstructed due to occlusion in the visual scene. The result of this reconstruction is a complete representation of the predator,

which can then be directly used for classification (identifying it as a predator), decision-making (initiating flight), or other behaviorally relevant tasks. CAMs have indeed been applied to such tasks ; for example, AMNs have been used to improve image classification performance in DNNs (Yang & Ding, 2020).

To understand why this characteristic is remarkable, it is useful to compare CAM with the way conventional computer memory operates. Conventional computers use Random Access Memory (RAM), which relies on location-based addressing : information is stored at specific numerical addresses that must be known exactly for retrieval. This means that each fragment of data (or memory) is isolated in a specific location within the overall memory system. Stored in this way, the memory cannot be queried based on its content. In the previous example, providing only a partial cue, such as a noisy picture of a '3', would not elicit the retrieval of the complete matching piece of information (the full '3' picture), since there are no inherent associative properties in this storage scheme. In such architectures, another component external to the memory system, such as the central processing unit (CPU) in a conventional computer, must handle any form of comparison between stored items. To achieve pattern matching, the CPU would first need to know the addresses of all candidate memories, retrieve them one by one, and then run an explicit algorithm to compare them with the query. This process is sequential, time-consuming, and entirely dependent on external control (Zou et al., 2021), in stark contrast to CAM systems, where associative matching is an inherent property of the memory itself that occurs through its dynamics.

In summary, CAMs integrate within their architecture both the capacity to store memories for later retrieval and the computational ability to match patterns, allowing the stored information to be directly exploited for recognition, classification, or decision-making tasks (Yang & Ding, 2020; Liu & Mukhopadhyay, 2018; Ramsauer et al., 2021).

2.3.2 . Von Neumann bottleneck

AMN are not only robust to noise and capable of content-based retrieval, they are also, at least in theory, superior to traditional computer architectures in that they offer a potential way to bypass a limitation known as the von Neumann bottleneck (Zou et al., 2021).

This bottleneck arises from the fundamental design of the von Neumann architecture, in which memory and computation are physically separated. In such systems, as mentioned previously, the central processing unit (CPU) must continuously fetch data from memory, perform the necessary operations, and then write the results back (Zou et al., 2021; Sze et al., 2017). For pattern matching, this requires retrieving each candidate memory one by one, transferring it to the CPU, and explicitly comparing it to the query. Because all these operations require data switching back and forth along the same connection

between the processor and memory, this link quickly becomes a bottleneck, allowing only part of the data needed for the computation to transit at any given moment, which in turn slows down the overall computation time (Sze et al., 2017). This architectural constraint, known as the von Neumann bottleneck, is particularly detrimental for tasks that require rapid associative retrieval (Pagiamtzis & Sheikholeslami, 2006).

By implementing AMNs on a neuromorphic substrate whose hardware directly mirrored the network's dynamics (Sec. 2.2.4), this bottleneck could be avoided (Indiveri & Liu, 2015; Sebastian et al., 2020). In such an implementation, storage and computation would share the same physical substrate : neurons and their connecting synapses would each be realized as dedicated circuits, simultaneously holding the information that encodes the memories and performing the computations required for retrieval (Sebastian et al., 2020). Presenting a partial cue would automatically drive the network toward the matching memory through its own dynamics, without the need for an external control loop or repeated data transfers. Because these operations would occur in parallel across all neurons and synapses, each acting as a small computational unit, the process would inherently distribute both data transfer and computation over the entire network. This avoids the constant back-and-forth of information between separate memory and processing units that slows conventional von Neumann systems, and therefore allows fast, efficient associative matching (Indiveri & Liu, 2015; Pagiamtzis & Sheikholeslami, 2006). We will not delve further into this topic as it lies outside the scope of this thesis, but being aware of the von Neumann bottleneck helps explain the strong interest surrounding AMN.

The strict reliance on implementation modalities, specifically the need for neuromorphic hardware to fully exploit the advantages of AMN, is a key consideration in this thesis. For this reason, we focus on learning rules and network dynamics that are local and simple, ensuring compatibility with neuromorphic implementation.

2.3.3 . A model for the hippocampus

AMNs have long been considered as potential models for hippocampal architecture and function, reflecting the central role of this brain structure in memory formation and retrieval (Marr, 1971; Treves & Rolls, 1994). The hippocampus, particularly the CA3 region with its extensive recurrent connections, exhibits architectural characteristics that are remarkably similar to those assumed in AMN models (Li et al., 1994; Miles & Wong, 1986). The dense interconnectivity, synaptic plasticity mechanisms, and pattern completion capabilities observed in hippocampal circuits provide biological grounding for the computational and structural features of AMNs (Miles & Wong, 1986; Bliss & Lømo, 1973b; Neunuebel & Knierim, 2014).

Recent experimental evidence has shown that hippocampal neurons form distinct groups or assemblies that represent specific memories. Retrieval of a given memory has been associated with the reactivation of its distinct hippocampal assembly (Liu et al., 2012; Gelbard-Sagiv et al., 2008). The phenomenon of pattern completion in the hippocampus, where partial sensory cues can trigger the recall of complete memory representations, directly parallels the fundamental operation of AMNs (Guzman et al., 2016).

Understanding AMNs thus provides a valuable framework for investigating how biological neural circuits implement memory functions. By establishing models that capture the essential features of hippocampal computation, we can generate testable predictions about memory storage capacity, retrieval dynamics, and the conditions under which biological memory systems fail or exhibit pathological behavior (Rolls & Kesner, 2006; Kesner & Rolls, 2015).

In the following sections, we examine the architecture of discrete Hopfield networks (DHNs), which serve as a paradigmatic model of AMNs. We will begin by introducing the McCulloch–Pitts neuron, the basic computational unit of DHNs, before describing in detail their architecture, memory storage mechanisms, and memory retrieval dynamics.

2.3.4 . Discrete Hopfield Networks

McCulloch and Pitts model : the abstracted neuron

For the AMNs presented here using binary units, such as DHNs, the foundational neuron model is the McCulloch and Pitts neuron (M-P neuron). Warren S. McCulloch and Walter Pitts introduced one of the earliest minimalist models for a neuron in their seminal paper, "A Logical Calculus of the Ideas Immanent in Nervous Activity" (McCulloch & Pitts, 1943). By simplifying biological neurons to logical processing units, McCulloch and Pitts showed how neural activity could be interpreted in terms of propositional logic, helping to bridge the gap between neuroscience and computational sciences. The M-P

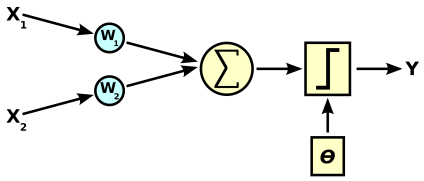

Figure 2.5 – **Schematic of a McCulloch–Pitts neuron** (McCulloch & Pitts, 1943), showing binary inputs, synaptic weights, summation, thresholding, and binary output. Source : *Wikipedia*.

neuron (Fig. 2.5) is defined as :

$$y = H(h)$$

with h the neuronal drive

$$h = \sum_{i=1}^{N} w_i x_i - \theta$$

and $H(x)$ the Heaviside function

$$H(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

Therefore, M-P neurons are characterized by the following elements :

- **Inputs** $\{x_i\}$: A set of binary inputs $x_i \in \{0,1\}$ with $i \in \{1,\ldots,N\}$ where N is the number of inputs. They correspond to variations in membrane potential at the soma or dendritic tree induced by signals from other neurons carried by synapses.
- **Weights** $\{w_i\}$: Each input x_i is associated with a weight w_i. In the original M-P model, the weights were often restricted to 1 or -1, which correspond, respectively, to excitatory and inhibitory connections. Modern variants allow for real-valued weights. The weight corresponds to

the variable intensity of the depolarization induced by presynaptic activity. The larger the weight, the more the associated input will influence the state of the neuron.

- **Threshold** θ : A constant that determines whether the sum of weighted inputs is sufficient to "activate" the neuron.
- **Drive** h : Corresponds to the net input integrated by the neuron with $h = \sum_i w_i x_i$. This net input "drives" the activity of the neuron toward activated (1) or inactivated (0) states.
- **Output** y : A binary output $y \in \{0, 1\}$ such that :

$$y = H(h) = \begin{cases} 1 & \text{if } h \geq \theta, \\ 0 & \text{otherwise.} \end{cases}$$

Neurons are not the only components of the nervous system that exhibit the general characteristics of the M-P neuron. The dendritic tree of cortical pyramidal neurons can perform nonlinear integration of presynaptic signals and, therefore, may be modeled using M-P neurons (London & Häusser, 2005; Poirazi et al., 2003). In addition, entire populations of neurons can be considered as functional units, integrating signals from various sources and generating nonlinear outputs. Therefore, computational nodes from the network models considered in this thesis could correspond to any of these biological constituents. However, we will still refer to them as neurons for readability throughout the remainder of this work.

In the next section, we discuss DHNs, for which the fundamental computational units are M-P neurons.

Discrete Hopfield networks (DHNs) constitute one of the most influential AMN models. Introduced by John Hopfield in 1982 (Hopfield, 1982), DHNs are recurrent neural networks designed to store and retrieve memory states through their dynamics. The network consists of a fully connected population of neurons that update their states according to a deterministic rule. Fully connected means that every one of the neurons sends a synapse to every other neuron, as shown in Fig. 2.4.

As discussed in Sec. 2.3, a key feature of Hopfield networks (HNs) is their ability to perform pattern completion. For pattern completion to occur, the network is initialized with a partial or noisy version of a stored pattern (the query). The network dynamics are then simulated until convergence. Convergence corresponds to the state at which, despite further simulating the dynamics of the network, the state of the network does not change anymore. If the convergence is successful, the state of the network corresponds precisely to the stored pattern. If the input query (the initial state) corresponds to a noisy version of the stored pattern, the final state of the network is therefore a "denoised" version of the query. The whole procedure is visualized in Fig. 2.3.

These dynamics arise from the fact that, following the storage procedure, the network states corresponding to the stored patterns constitute attractors toward which the network state evolves when updated. Although originally formulated for binary neurons, the model has been extended to continuous variables and has found applications in optimization problems and pattern recognition, serving as a foundational framework for understanding memory storage in neural systems (Hertz et al., 1991; Hopfield & Tank, 1985). The main work of this thesis presented in Chap. 4 uses continuous Hopfield networks (CHNs), which, rather than using binary M-P neurons, use continuously valued units. Despite being a different model, all the properties highlighted in this section hold for CHNs.

Since DHNs serve as a key example for understanding the challenges of storing correlated patterns in AMNs (Chap. 3), the next section discusses their architecture and update scheme.

Structure and dynamics of DHNs

DHNs are composed of many M-P neurons interconnected by real-valued synapses. For N binary units, we define the **state vector** $\mathbf{s} = (s_1, \ldots, s_N) \in \{-1, 1\}^N$ and the weight matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$, with W_{ij} the weight of the connection from unit j toward unit i, such that :

$$h_i(t) = \sum_j W_{ij} s_j(t) \tag{2.1}$$

where h_i is the net input (or drive) of unit i. In the standard asynchronous update scheme, a randomly chosen unit i is updated according to the following rule :

$$s_i(t+1) = \begin{cases} 1 & \text{if } h_i(t) \geq 0, \\ -1 & \text{otherwise,} \end{cases} \tag{2.2}$$

where 0 is the activity threshold. The update dynamics are detailed in Sec. 2.3.4.

Classic DHNs enforces the symmetry of the weight matrix, which means that $\forall i, j \quad W_{ij} = W_{ji}$. Under these circumstances, the dynamics are guaranteed to converge to a fixed point, which therefore constitutes an attractor (Hopfield, 1982). Some implementations also remove synapses that connect a neuron to itself (autapses) by setting the diagonal terms $W_{ii} = 0$. This facilitates certain analytical studies of the network's properties and slightly improves storage performance (Hopfield, 1982). In our case, we will consider that these synapses can have non-zero values, as this facilitates some of the analytical approaches used in Chap. 3.

Moreover, classic DHNs operate with bipolar neuron states $(-1, +1)$, in contrast to the original M-P neurons that use binary states $(0, 1)$. At first glance, this choice appears to be less biologically plausible, since real neurons cannot exhibit negative firing rates. However, simple transformations, such as affine rescaling of the activation function or an appropriate offset in the synaptic input, can reconcile this discrepancy (Davey et al., 2004). These adjustments preserve the fundamental convergence properties of the model while allowing it to operate with binary (0/1) neuron states, thus improving its biological interpretability without sacrificing mathematical tractability.

Plasticity as a Memory Storage Mechanism

For DHNs to serve as models for associative memory, a mechanism is needed to allow the storage of memories for subsequent retrieval. The mechanisms considered in this thesis are mainly based on modifying synaptic weights $\mathbf{W}$ to establish the state of the network associated with stored memories as attractors in network dynamics, with the exception of the modern Hopfield networks that are quickly discussed in Chap. 3.

A 'memory' here corresponds to the information encoded in the network by an explicit storage process such as the modification of the connection weights. When using neural networks (NNs) as associative memories, it is typically the information the user wants to recover from partial queries. The terms 'stored pattern', 'memory' and 'memory state' will be used interchangeably throughout this work.

The first mechanism used to allow the storage of memories in DHNs, introduced by Hopfield in his seminal paper, is a form of Hebbian learning (Hopfield, 1982). Hebbian learning refers to a class of plasticity rules where the change in synaptic strength between two neurons is proportional to the product of their activities as introduced in Sec. 2.1.2. Given a set of M patterns $\{\mathbf{x}^\mu\}^M$ to be stored as memories, the weight matrix is constructed as :

$$W_{ij} = \frac{1}{N} \sum_{\mu=1}^{M} x_i^\mu x_j^\mu \tag{2.3}$$

where N is the network size. N serves as a normalization factor that does not affect the dynamics of the network, but facilitates the subsequent analytical treatment.

Eq. 2.3 can be viewed as a sum of independent plastic events, where each pattern μ contributes to the weight matrix through :

$$\Delta W_{ij} = \frac{1}{N} x_i^\mu x_j^\mu$$

Each stored pattern thus modifies the same set of synaptic weights. A parallel can be drawn between these plastic events and engram formation as discussed in Sec. 2.1.2. Each plastic event can be seen as the formation of a new engram underlying a new memory.

It is important to note that this plasticity occurs only during the storage phase, when patterns are encoded into the network weights. During retrieval, when the network dynamics induces pattern completion from partial queries, the weights remain fixed, and only the neuron states evolve according to Eq. 2.2 for an asynchronous scheme (more details on the different update schemes are given in the next section).

The storage capacity of DHNs is limited. When the number of stored patterns exceeds approximately $0.14N$ (where N is the network size), the network

enters a spin glass phase characterized by many strange attractors that do not correspond to any stored pattern (Amit et al., 1985a). In this regime, retrieval becomes unreliable as the network may converge to spurious states rather than to the stored memories. Spurious states are discussed in Sec. 3.1.1.

The apparition of a spin glass phase is a form of catastrophic forgetting (CF) for which the network's ability to retrieve previously stored patterns degrades quickly as new patterns are added above the capacity limit. This capacity-induced forgetting differs from the correlation-induced forgetting that arises when storing strongly overlapping patterns, which will be the primary focus of Chap. 3.

Once patterns are stored, the network dynamics must be simulated to enable retrieval from partial cues. The next section describes the two main strategies used to simulate the retrieval dynamics.

Retrieval dynamics

A typical retrieval procedure begins by initializing the network with a partial query associated to a stored pattern $\mathbf{s}(0) = (\mathbf{x}^\mu)'$, with $(\mathbf{x}^\mu)'$ the partial query associated to the stored pattern $\mathbf{x}^\mu$. $(\mathbf{x}^\mu)'$ can be a noisy version of the stored pattern, as shown in Fig. 2.3.

After this initialization, an update scheme is applied to simulate the dynamics of the network until convergence.

Two main update schemes are used to simulate DHN dynamics : asynchronous and synchronous updates. Each has distinct computational properties and convergence behaviors.

For asynchronous updates (Alg. 1), only one randomly selected neuron changes its state per time step. This guarantees convergence to a fixed point for symmetric networks (Hopfield, 1982).

Synchronous updates (Alg. 2) change all neurons' states simultaneously according to the current state of the network. Synchronous updates can be computationally less demanding, as convergence is usually achieved in few updates (such as 4), but may lead to oscillatory behavior rather than successful convergence to a fixed point (Hertz et al., 1991).

Successful retrieval occurs when the state after convergence corresponds to the stored state $\mathbf{x}^\mu$ associated to the query $(\mathbf{x}^\mu)'$.

The simulation shown in Fig. 2.3 is based on the asynchronous update scheme.

Algorithm 1 Asynchronous Update

Initialize the network with partial query $\mathbf{s}(0) = (\mathbf{x}^{\mu})'$
$t \leftarrow 0$
while not converged **do**
    Select random unit $i \in \{1, \ldots, N\}$ with equal probability
    $h_i \leftarrow \sum_j W_{ij} s_j(t)$
    $s_i(t+1) \leftarrow \text{sign}(h_i)$
    $t \leftarrow t + 1$
end while
return $\mathbf{s}(t)$

Algorithm 2 Synchronous Update

Initialize the network with partial query $\mathbf{s}(0) = (\mathbf{x}^{\mu})'$
$t \leftarrow 0$
while not converged **do**
    for $i = 1$ to N **do**
        $h_i \leftarrow \sum_j W_{ij} s_j(t)$
        $s_i(t+1) \leftarrow \text{sign}(h_i)$
    end for
    $t \leftarrow t + 1$
end while
return $\mathbf{s}(t)$

2.3.5 . Conclusion

AMNs exemplify how some of the core aspects of neural architecture can lead to functional models : distributed processing, parallel operation, and emerging functionalities arising from local interactions. Through their recurrent dynamics, these networks achieve information storage and retrieval in a manner that is considered to mirror biological memory systems.

Rather than localizing the information specific to a single memory in an isolated part of the memory system, AMNs encode memories through distributed synaptic modifications across all neurons in the network. This distributed organization enables the decentralized computational paradigm described in Sec. 2.2.1, where each neuron acts like a voter casting its decision based on local inputs. Pattern completion dynamics emerge from these collective choices.

AMNs present computational advantages, especially when considering their implementation in hardware architectures that can leverage parallelization, such as neuromorphic hardware. Unlike conventional von Neumann systems that separate memory and computation, these brain-inspired architectures can exploit the distributed nature of AMNs to achieve efficient, parallel pro-

cessing for pattern storage and retrieval.

Beyond their computational potential, AMNs are considered to emulate properties of parts of the nervous system such as the hippocampus, offering opportunities to deepen our understanding of memory storage and retrieval mechanisms in the brain.

To provide a concrete example of AMNs, we presented DHNs. DHNs are considered the prototypical implementation of AMNs and will be extensively discussed throughout this thesis. The main contribution of this thesis, presented in Chap. 4, is based on continuous Hopfield networks (CHNs), which are a variant of Hopfield networks that exhibit similar fundamental properties.

While AMNs demonstrate capabilities for storing and retrieving patterns, they face a fundamental challenge when required to continuously learn new information : the storage of new patterns can interfere with and overwrite previously stored memories. This phenomenon, known as catastrophic forgetting (CF), affects not only AMNs but also most ANN models. The following section explores how rehearsal techniques mitigate this problem in ANNs and the parallels that can be drawn between these computational strategies and the memory replay dynamics observed in the brain.

2.4 . Memory replay and rehearsal methods

When artificial neural networks (ANNs) learn new information, they tend to forget previous learning in an uncontrolled manner. This phenomenon is called catastrophic forgetting (CF).

Extensive research in machine learning demonstrates that ANNs benefit from revisiting previously learned data when learning new information. This process, called rehearsal, helps mitigate CF (Robins, 1995; Shin et al., 2017; Rolnick et al., 2019).

In parallel to these computational considerations, it has been found that the brain revisits activity patterns that correspond to past experiences, mainly during sleep (Wilson & McNaughton, 1994; Lee & Wilson, 2002; Foster & Wilson, 2006; Jadhav et al., 2012). This phenomenon is called memory replay (MR).

These biological observations, combined with computational considerations, motivate the following hypothesis : the brain could use MR to mitigate CF, similar to how artificial neural networks benefit from rehearsal.

In this chapter, we examine the phenomenon of MR in the brain and its functional interpretations through the lens of machine learning approaches. Understanding MR, its biological basis, and its application in machine learning techniques provides the essential background necessary to appreciate the motivations behind the work presented in Chap. 4.

2.4.1 . Memory replay in the brain

A memory replay (MR) is the reactivation of an activity pattern in a population of neurons that mirrors the activity pattern originally observed in this population during a specific experience. This reactivation occurs without re-exposure to any stimulus that could have initially induced the activity pattern. We therefore say that this phenomenon is spontaneous.

A well-studied example occurs in the hippocampus, a brain region crucial for memory formation (Scoville & Milner, 1957). The first direct evidence came from recordings in the CA1 region of the hippocampus in rats : groups of neurons that were active during the exploration of an environment tended to be reactivated during subsequent sleep periods (Wilson & McNaughton, 1994). In later work, replay was reported during rapid eye movement (REM) sleep, a phase during which vivid dreaming occurs and memory processing is often discussed (Louie & Wilson, 2001; Nir & Tononi, 2010). MR has also been reported in the same region during other phases of sleep, such as slow-wave sleep (Lee & Wilson, 2002). Additional work showed that replay in the hippocampus can coincide with activity in the primary visual cortex (V1), indicating coordination between large populations of neurons in the central nervous system (Ji & Wilson, 2007). Replay has also been observed in hippocampal CA1 during quiet wakefulness, when animals are immobile or pausing between movements (Foster & Wilson, 2006; Jadhav et al., 2012).

MRs are especially common during events that organize cortico-hippocampal communication, notably sleep spindles : brief bursts of rhythmic activity (9–16 Hz) produced by interactions between the thalamus and cortex. These events are easily visible on an electroencephalogram (Fernandez & Lüthi, 2020; Clawson et al., 2016). During spindles, the influence of external inputs, such as sensory modalities, is reduced, and intracortical information flow is favored, thus priming autonomous replay-like activity (Ngo, 2020).

Historically, one of the first hypotheses to emerge proposed that replay events facilitate long-term memory storage by transferring newly formed memories from the hippocampus to the neocortex, where they can be maintained over long periods (McClelland et al., 1995). In support of this view, studies have documented a progressive shift from hippocampal to neocortical structures that underlies successful memory recall after offline (sleep-related) consolidation periods (Takashima et al., 2006). This view is entirely focused on the role of replaying recent experiences for the long-term consolidation of newly learned information. It is important to mention this hypothesis, as it represents the first widespread theory on the role of replay during sleep and already implicates replay in a continuous learning (CL) framework, namely the ability to store memories long-term to prevent forgetting.

The above view emphasizes the role of revisiting recent experiences for the consolidation of newly learned information. However, converging evidence indicates that replay is not limited to recent experiences. Replay can target previously encountered remote experiences, including trajectories or locations that are significant but have not been recently experienced (Karlsson & Frank, 2009; Gillespie et al., 2021). The replay dynamics therefore appear able to sample from the archive of past experiences. This observation has motivated a complementary functional interpretation. In this interpretation, some replays act as rehearsal and serve to stabilize established memories against drift and interferences induced by new learning, rather than consolidating newly acquired information (Káli & Dayan, 2004; Ólafsdóttir et al., 2018).

Machine learning and computational neuroscience studies have explored the advantages offered by implementing a rehearsal mechanism inspired by MRs for the training of ANNs. The following sections discuss these methods.

2.4.2 . Rehearsal and continuous learning in artificial networks

Rehearsal is a technique that allows ANNs to mitigate catastrophic forgetting (CF). CF occurs when an ANN abruptly and drastically forgets previously learned information upon learning new data. This phenomenon poses a fundamental challenge for CL, in which networks must adapt to new contexts while preserving potentially important information acquired from past experiences.

By revisiting previously learned data while also training on new data, ANNs can selectively strengthen the synapses that support the encoding of both old and newly acquired information (Robins, 1995; McCallum, 1998; Rebuffi et al., 2017). We say that rehearsals allow an ANN to find the joint synaptic distribution that supports the retention of past and current experiences (Shin et al., 2017). The emergence of such a joint synaptic distribution in ANNs is illustrated in the next section. This restructuring of synaptic weights can be interpreted as a form of credit assignment, determining which connections should be preserved, modified, or strengthened to accommodate old and new knowledge.

The necessity for rehearsal arises from the fundamental architecture of neural networks : the same neurons and synapses that encode existing knowledge must be recruited to learn new information, as discussed in Sec. 2.2.2. This overlapping substrate can cause new learning to disrupt previously established connections. These disruptions are known as interferences. When such interferences severely impair the network's function, the phenomenon is termed catastrophic interference. Catastrophic interference is the primary cause of CF in ANNs (McCloskey & Cohen, 1989; French, 1999). Without a mechanism to compensate for these disruptions, the network's performance on earlier tasks degrades severely. As will be seen in Chap. 3, in the case of AMNs, rehearsal offers the opportunity to reduce interferences, which explains its efficacy in mitigating CF. For other types of ANNs, such as deep neural networks (DNNs), it would be overly reductive to claim that rehearsal mitigates CF solely by reducing interferences, but this issue lies outside the scope of this thesis. In any case, the goal remains to find a joint synaptic distribution that supports the retention of past and current experiences, as mentioned above.

In conventional training of DNNs used for state-of-the-art applications, such as large language models (LLMs) for text generation or diffusion networks for image synthesis, rehearsal is implicitly present during learning. These networks typically loop through their training datasets multiple times (epochs) or employ heterogeneous sampling strategies that allow the same information to be interleaved repeatedly throughout training, a method called batch learning (Bottou, 2010). This continuous re-exposure to diverse data helps maintain performance across the full range of learned information. Batch learning is therefore generally considered to fully mitigate CF. Matching its

performance under incremental learning constraints can thus be seen as a natural benchmark for continuous learning methods.

When training on new data and without the ability to rehearse examples from their original training distribution, state-of-the-art architectures experience strong CF (Goodfellow et al., 2013; Kirkpatrick et al., 2017). In these large-scale networks, CF is particularly dramatic, as the network not only forgets anecdotal information (such as a specific date), but also tends to forget abstract concepts or learned features underlying its ability to perform complex tasks.

In the context of ANNs, learning from new data to improve a model's performance on a specific task that the original dataset does not sufficiently cover is called fine-tuning. Typically, fine-tuning must be performed without access to the original dataset for rehearsal, enabling users to specialize their models on their own data without requiring access to the entire initial training set. Extensive fine-tuning typically induces CF (Goodfellow et al., 2013; Kirkpatrick et al., 2017). It is important to note that for large models, a small amount of fine-tuning does not induce CF initially, but the extensive fine-tuning often necessary to achieve strong performance on new data does (Goodfellow et al., 2013; Kirkpatrick et al., 2017). From this perspective, extensive fine-tuning represents a CL scenario in which existing approaches fail to adequately preserve crucial knowledge from the original training dataset when assimilating new data.

In summary, rehearsal is a method that allows networks to circumvent CF. However, existing methods require the network to have access to all or part of the previous training dataset when learning new data. This external source acts as a teacher that reviews previous material with its students to prevent forgetting when introducing new information : the network is "given" the data considered necessary to avoid CF. In that sense, the network is not autonomous when performing rehearsal.

If we consider the ANN as a learning agent, rather than merely a component within a larger system that includes external data storage, this lack of autonomy poses a fundamental problem for CL. Under this perspective, "true CL" requires the agent to preserve important information from previously learned data. If the agent can simply access past data from an external source whenever needed, there is no genuine requirement to mitigate forgetting, as it can always retrain on those data after learning new tasks. Thus, continuous access to past data effectively eliminates the core challenge of CL. Autonomy in memory management is therefore an essential, though often implicit, requirement for genuine CL that deserves explicit recognition.

From this perspective, the brain provides an instructive example. In essence, the brain is an autonomous agent that cannot rely on an external memory system to rehearse information. In fact, through spontaneous MRs, the

brain appears to generate its own rehearsal internally. These observations motivate the hypothesis that the brain may fine-tune new information during wakefulness and rehearse past information during sleep to avoid CF.

We can imagine a similar scenario for an ANN. We perform a small amount of fine-tuning on new data (analogous to awake activity). We then let the network "sleep," which induces rehearsal on internal memory representations to prevent forgetting. We then continue fine-tuning during subsequent awake phases and allow MRs during sleep phases. This interleaving of awake (fine-tuning) and sleep (rehearsal through MRs) periods would allow CL from an ongoing stream of data without requiring any portion of the previously encountered dataset to be made available at any time.

These considerations motivate the development of new models capable of autonomously revisiting all or part of previously stored information through the dynamics of the neural network itself. Such networks would be able to orchestrate rehearsal autonomously and thus seamlessly integrate new learning with existing knowledge.

In the next section, we examine two specific studies that demonstrates how bio-inspired networks can implement autonomous rehearsal by spontaneously reactivating previously learned activity patterns.

2.4.3 . Autonomous rehearsal

In this section, we review two studies that demonstrate how bio-inspired networks can mitigate CF during CL through internally generated activity.

Both approaches employ a form of pseudorehearsal (Robins, 1995) : the activity patterns produced by the network's internal dynamics during spontaneous activity phases do not necessarily correspond exactly to any specific stored memory. As discussed in the previous section, rehearsal on the exact stored data is generally considered to fully mitigate CF. Therefore, we can hypothesize that the partial efficacy of these methods is due to the fact that training on pseudo-replays does not allow the network to fully recover information lost during CL.

In contrast, the mechanism proposed in the main work of this thesis (Chap. 4) enables, with high probability and under favorable conditions, the retrieval of exactly the stored patterns and no other. We refer to this property as *perfect retrieval*. When all stored patterns are recovered during spontaneous activity, the learning algorithm has access to the complete set of stored patterns for training, as in batch learning, thus completely preventing CF.

Despite this difference between pseudo-reharsal and proper rehearsal, the method presented here shares many common points with the main work of this thesis and therefore deserves proper review.

Catastrophic Forgetting and the Pseudorehearsal Solution in Hopfield-type Networks

The thesis by Simon McCallum (McCallum, 1998), titled *Catastrophic Forgetting and the Pseudorehearsal Solution in Hopfield-type Networks*, is the closest precursor to the work presented in Chap. 4. McCallum mitigates CF in discrete Hopfield networks (DHNs) by training the network not only on new patterns to be stored, but also on states recovered during spontaneous activity phases analogous to sleep.

McCallum employs the following protocol each time a new pattern is added to the network :

1. Before storing a new pattern, the network is probed with a large number of random input states, each relaxed to convergence under the network dynamics (Sec. 2.3.4).
2. This variety of recovered states is then combined with the new pattern to form the complete training set.
3. Then a learning algorithm is applied to store the complete training set on the network.

The network, when storing a new pattern does not have access to past memories, it is therefore a CL protocol.

The learning algorithm used to store the complete training set is called delta learning rule in the work of Mccallum, and we formalize it as the Gradient descent algorithm (GDA) in the context of this thesis. This algorithm/plasticity rule is detailed in Sec. 3.1.2.

By including the recovered states in the training set, the network is trained both on the new pattern and on states that implicitly encode information about the existing attractor landscape. The forgetting induced by the storage of the new pattern is thereby mitigated, as the learning algorithm tends to preserve the basins of attraction (Sec. 2.3) of previously stored patterns.

The convergence from random initial states employed during pseudorehearsal does not guarantee that the recovered states correspond to stored patterns. The network may also converge to spurious attractors, i.e. stable states that do not correspond to any stored pattern (Sec. 3.1.1, Fig. 2.7). The population of recovered states therefore contains two types : *true items*, which correspond to stored patterns (true memories), and *pseudoitems*, which are spurious stable states (false memories). The presence of pseudoitems in the recovered population gives the procedure its name : *pseudo*rehearsal.

McCallum showed that training on pseudoitems alone, by removing any true item from the recovered population, partially preserves the basins of attraction of previously stored patterns. This surprising result is at the core of pseudorehearsal as formalized by Robins (1995). A network is able to partially

prevent the degradation of true memories by generating 'dreams' that do not correspond to the actual stored memories but constitute a form of proxy.

However, training on the recovered population remains substantially less effective than *perfect rehearsal*, in which the learning algorithm is applied to the exact set of stored patterns (containing all the true memories and no others).

As shown in Fig. 2.6, pseudorehearsal partially mitigates CF, maintaining a plateau of stable patterns that scales with the number of recovered states used for rehearsal. A pattern is said to be stable if, when the network is initialized in that state, the dynamics leave it unchanged. Stability is a necessary condition for a pattern to be considered as stored, but it is a weak criterion : a stable pattern may possess a narrow basin of attraction that makes it unretrievable from a degraded cue.

The recovered population only constitutes an approximate and incomplete representation of the set of stored patterns, which limits the efficacy of the rehearsal process. The work presented in Chap. 4 pursues the same objective of mitigating CF through autonomous rehearsal, but differs in a fundamental respect : rather than training on a mixture of true items and pseudoitems, our method aims to recover exactly the stored patterns, and no other, a property we refer to as *perfect retrieval*. As we demonstrate, for small memory loads, the autonomous retrieval mechanism presented in Chap. 4 indeed recovers only the stored patterns with high probability. No pseudo-items are generated.

A detailed reproduction and extension of McCallum's experiments, together with a comparison between our method and McCallum's pseudorehearsal, is presented in Sec. 4.8.5.

In the next section, we review a second study pursuing a similar objective in a different architecture : Golden et al. (Golden et al., 2022) demonstrate that sleep-like spontaneous activity can mitigate CF in feedforward spiking neural networks.

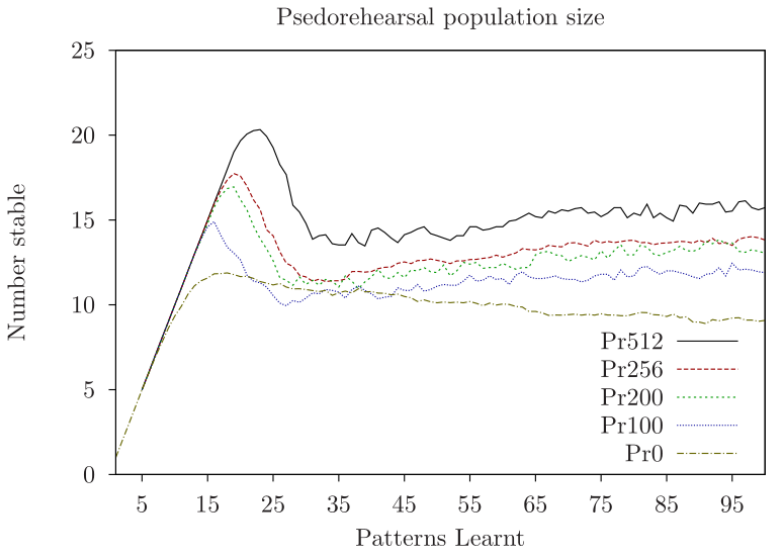

Figure 2.6 – **Pseudorehearsal in a DHN (adapted from McCallum (1998)).** Number of stable patterns as a function of the total number of patterns learned, for a network of size $N = 100$. Prk denotes the number of pseudo-items used for rehearsal : for each new pattern to be incorporated, at most k states recovered from the network dynamics are combined with the new pattern for training. Pro corresponds to the no-rehearsal baseline.

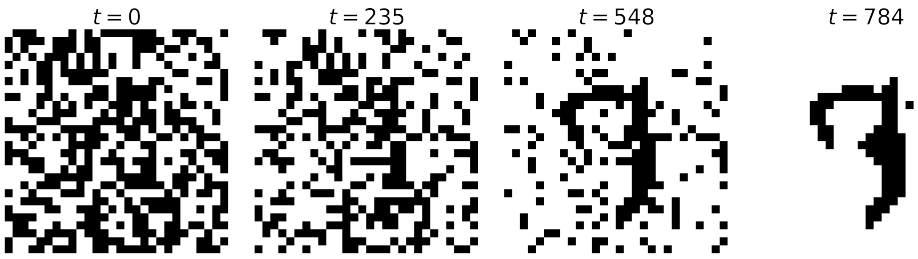

Figure 2.7 – A discrete Hopfield network (DHN) converging to a spurious attractor when initialized from a random state. In McCallum's pseudorehearsal protocol, many of the recovered states correspond to this kind of spurious attractor.

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

Here we focus on the work of Golden et al. (2022), which also inspired the research presented in this thesis.

Before describing their work, it is important to clarify the terminology and training protocols used in this computational neuroscience context. Golden and colleagues use the term "sequential training" to describe a scenario where a network learns one task during an extended period before moving on to the next, without the ability to rapidly alternate between learning phases on different tasks. As discussed in the Introduction, this constraint mirrors natural learning conditions faced by animals, including humans. For instance, during summer an animal may master specific foraging techniques, then must adapt to entirely different strategies when winter arrives; there is no opportunity to practice both skill sets simultaneously because the relevant environmental conditions do not coexist. This temporal separation of learning experiences is precisely the challenge addressed by "incremental learning" in the AMN literature (Sec. 3.2.1). Both sequential and incremental learning describe specific instances of the broader CL paradigm discussed in the Introduction. We maintain the terminology native to each field throughout this thesis to facilitate engagement with the respective literature.

In their specific implementation, the default sequential training protocol involves training on Task 1 (T1) for an extended period, then training on Task 2 (T2) for a subsequent extended period, reflecting the constraint of having access to only one task at a time. For this network, as in most neural network models, this sequential protocol typically leads to CF, whereby learning T2 causes the network to lose the information necessary to perform T1. In contrast, non-sequential learning corresponds to training on both tasks in a rapidly interleaved manner, mimicking a scenario in which training on both tasks occurs in close temporal proximity. In this configuration, the interleaved protocol enables the successful learning of both tasks without forgetting.

The key innovation in their approach was the implementation of sleep phases that effectively reproduce the benefits of non-sequential interleaved learning without requiring training on both tasks in close temporal proximity. During these sleep periods, the input layer was silenced while hidden-layer neurons received Poisson-distributed spike trains calibrated to maintain firing rates similar to those observed during task training. Crucially, this noise-driven activity allowed the network to spontaneously replay previously learned activity patterns, essentially creating an internally generated training phase that supported learning of both tasks despite the temporal separation constraint.

Fig. 2.8(a) presents a simulation during which a network is trained on T1 and T2. The simulation begins with a training phase on T1, during which the task is presented many times (blue area). At this point, the network performs

well on T1 but not on T2, as T2 has not yet been learned. The simulation continues with the network learning T2 for a certain period (red area). This new learning induces forgetting of T1. As the simulation progresses, rather than only presenting T1 for an extended period (which would induce forgetting of T2), a sleep phase is introduced. Interleaving sleep periods (shown in gray) with training on T1 preserves performance on T2 while enabling successful learning of T1. This occurs because during sleep periods the network revisits activity patterns reminiscent of T2, which prevents forgetting of T2 while learning T1. The study revealed that this sleep-like replay allows the synaptic weight configuration to evolve toward the intersection of the two manifolds in weight space that support learning of both T1 and T2 (Fig. 2.8(b)). In simpler terms, the network can find a configuration of synaptic weights that retains the information necessary to perform both T1 and T2.

In this setup, the network does not rely on accessing past data to enable learning of new tasks while maintaining performance on old ones. The model is a self-contained, autonomous, adaptive system that can continuously fit both old and new data without requiring external storage of previously encountered training examples. The network generates its own internal replay through sleep-like states. This property enables the system to operate in dynamic environments where previous training data may be unavailable.

Similarly to McCallum's pseudorehearsal (Sec. 2.4.3), the activity patterns generated during sleep-like phases in Golden et al.'s framework do not, in general, correspond exactly to any specific previously learned representation. Both methods therefore rely on a form of pseudorehearsal : the internally generated activity provides an approximate representation of previously learned information. Learning/plasticity on this activity partially mitigates CF but does not eliminate it entirely.

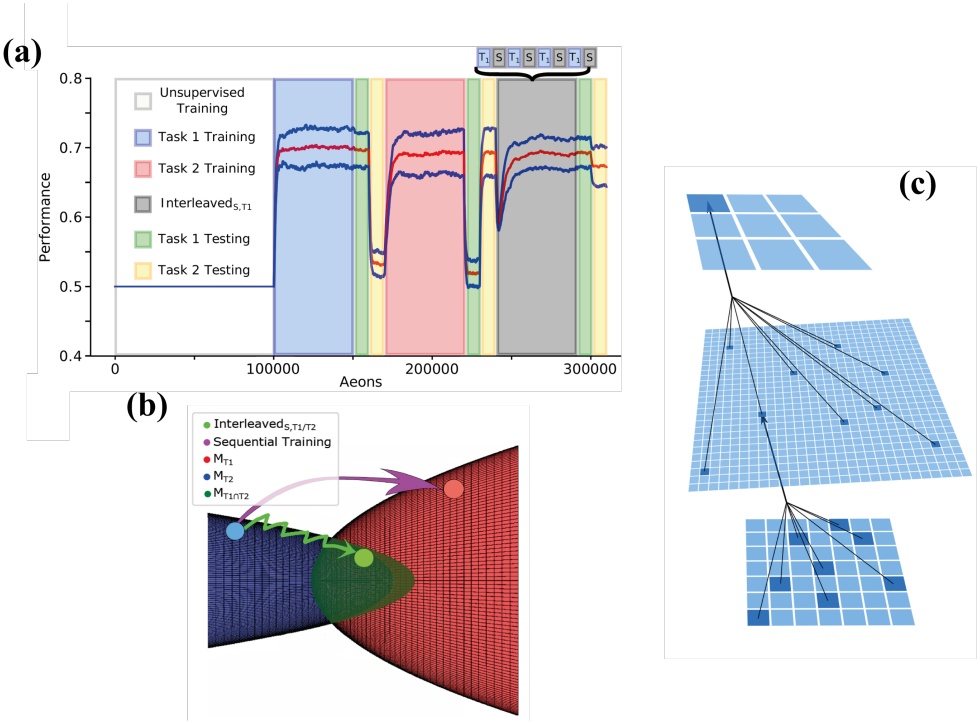

Figure 2.8 – **Autonomous rehearsal through sleep prevents catastrophic forgetting in spiking neural networks (adapted from Golden et al. (2022)). (a)** Performance dynamics during sequential task learning with interleaved sleep periods. The Red line shows mean performance and dark blue lines indicate variability (e.g., ± 1 SD) on the current task. During Task 1 (T1) training (blue shaded area), the network is trained and tested on T1 to obtain the performance score; similarly for Task 2 (T2) training (red shaded area). T1 testing (green shaded areas) reveals catastrophic forgetting—performance on T1 drops dramatically after T2 training. However, the *Interleaved S,T1* period (gray shaded area), consisting of alternating sleep and T1 training, allows the network to recover performance on T1 while maintaining T2 performance through autonomous rehearsal dynamics (yellow shaded areas indicate T2 testing). **(b)** Synaptic weight-space visualization showing task-specific manifolds MT1 (red) and MT2 (blue) and their intersection MT1 $\cap$ MT2 (green). Sequential training (purple arrow) causes direct jumps between manifolds, leading to forgetting, whereas *Interleaved S,T1/T2* training (green trajectory) guides the network toward the intersection where both tasks can be supported. **(c)** Feedforward architecture with three layers—input (7×7), hidden (28×28), and output (3×3)—connected through plastic synapses.

2.4.4 . Conclusion

Rehearsal, defined as revisiting previously learned data, effectively prevents catastrophic forgetting (CF) in ANNs by enabling the discovery of synaptic configurations that accommodate both old and new knowledge. Similarly, evidence demonstrates that the brain spontaneously replays memory patterns during sleep and quiet wakefulness, suggesting that biological neural networks may employ comparable rehearsal strategies to mitigate CF. Rehearsal strategies in ANNs typically require access to all or part of the training dataset from an external source. In contrast, the brain must operate autonomously when performing rehearsal, relying exclusively on self-generated neural activity.

Moreover, autonomy represents not merely a constraint affecting learning in biological systems but emerges as a fundamental requirement for CL. Relaxing this requirement essentially eliminates the need to address CF altogether, as agents could simply retrain on externally stored data whenever necessary, rather than genuinely preserving previously acquired knowledge within the network itself.

From a practical perspective, autonomous rehearsal would offer significant advantages for real-world applications. It would enable third parties, such as enterprises, to fine-tune models without requiring access to proprietary training datasets, which is a critical consideration for large language models (LLMs) trained on vast portions of the internet and sensitive human-generated data. Furthermore, autonomous rehearsal would empower embedded AI systems in robots, autonomous vehicles, and edge devices to continuously adapt to new environments while preserving essential capabilities, all without dependency on external data storage or connectivity.

Perhaps more importantly, understanding the mechanisms that enable autonomous rehearsal in ANNs could offer significant insights into how the brain performs memory replay and mitigates CF.

As mentioned in this section, CF arises from the way memories are stored in both ANNs and the brain : the same neurons and synapses that encode existing knowledge must be recruited to learn new information. This characteristic of memory storage in neural networks induces interferences that degrade information related to previously learned data. In Chapter 3, we will discuss how and why interferences occur in AMNs and how to mitigate them. The next chapter first provides a brief review of alternative methods that enable CL in bio-inspired network architectures.

2.5 . Alternative Continuous Learning Approaches

The previous section reviewed how rehearsal can mitigate CF by enabling the discovery of synaptic configurations that support both old and new memories, which constitutes the core topic of this thesis. This section, by contrast, broadens the scope to examine other biologically plausible CL solutions.

Rather than relying on rehearsal, these approaches address the tension between learning and memory preservation through alternative mechanisms.This tension, formalized by Grossberg as the *stability–plasticity dilemma* (Grossberg, 2013), captures a fundamental constraint faced by any learning system that uses shared resources to encode multiple experiences : excessive plasticity allows new learning but erases existing memories, whereas excessive stability preserves old memories but prevents adaptation. The two methods reviewed in this section address this dilemma through strategies that regulate plasticity rather than rehearsal : Adaptive Resonance Theory, a family of architectures that gates learning through a resonance mechanism, and Elastic Weight Consolidation, which selectively reduces plasticity on synapses deemed important for past tasks.

2.5.1 . Adaptive Resonance Theory

Adaptive Resonance Theory (ART), introduced by Grossberg and later developed with Carpenter into a family of neural network architectures (Grossberg, 1980; Carpenter & Grossberg, 1987), was explicitly designed to address the stability–plasticity dilemma.

In its simplest form, an ART network can be understood as an AMN composed of an input neuron population and a recognition population, connected through bottom-up and top-down plastic synapses. The architecture of the network and its principal dynamics are illustrated in Fig. 2.9. The principal steps of the ART dynamics are summarized below :

1. First, an input presented to the network induces an activity pattern in the input population.
2. This input activity activates a specific group of neurons in the recognition population through bottom-up plastic synapses. This group of neurons correspond to a category.
3. Once activated, the category neurons project back an expectation through top-down synapses onto the input population. This top-down influence corresponds to a learned template associated with that category. The template is then compared with the actual input.
4. If the match between the template and the input exceeds a threshold set by a *vigilance parameter*, the system enters a resonant state in which learning is permitted and the template is refined to incorporate the features of the new input through synaptic plasticity. If the match is insuffi-

cient, the current category neurons are inhibited, and a search process is initiated until a better-matching category is found.

5. If no existing category meets the vigilance criterion, a new category is created by recruiting a new group of neurons in the recognition population and potentiating the corresponding synapses.

To illustrate the network dynamics in an ecological scenario, consider a hunter-gatherer who has learned to recognize a set of edible berries. Each type of berry is represented by a category associated with a template encoding its characteristic features (color, shape, texture). When the forager encounters a new fruit, the input pattern of visual and tactile features is propagated through bottom-up synapses, potentially activating the appropriate category neurons whose template best matches the current stimulus. If the fruit closely resembles a known berry, the match exceeds the vigilance threshold : the corresponding template is refined through plasticity to incorporate the minor variations of this new exemplar. If, however, the fruit differs substantially from all stored templates (for instance, an unfamiliar mushroom), the match falls below the vigilance threshold for every existing category. The search process then recruits a previously uncommitted group of neurons (or a single neuron) into the recognition population, which becomes the substrate of a new category. The bottom-up and top-down connection weights are updated between these neurons and the input neurons that encode the distinguishing features of the mushroom, effectively allocating a dedicated neural population for this novel experience.

This match-based learning process ensures that existing memories can only be modified when incoming data are sufficiently consistent with stored expectations. In contrast, genuinely novel inputs are allocated to dedicated categories, preventing destructive interference with established memories.

Grossberg developed ART as a theory of cortical processing, and the architecture was designed to reflect key organizational principles of the brain (Grossberg, 2013). The reciprocal connectivity between the input and recognition populations mirrors the feedforward and feedback projections that connect cortical areas, and all synaptic updates rely exclusively on locally available information, consistent with the locality principle discussed in Sec. 2.2.3.

While the ART architecture provides a principled solution to the stability–plasticity dilemma, it faces several well-documented limitations. First, the vigilance parameter creates a direct trade-off between specificity and generality : high vigilance leads to *category proliferation*, in which the number of stored categories grows excessively (Brito da Silva et al., 2019). Various strategies have been proposed to mitigate this issue (Carpenter et al., 1991; Isawa et al., 2008), but to our knowledge, no account in the literature on practical applications of ART suggests that category proliferation has been fully resolved. Second, extending ART dynamics to deep hierarchical representations is nontrivial, as

the architecture was not originally designed for multi-layer feature learning. Recent efforts such as DeepART (Petrenko et al., 2025) and Deep ARTMAP (Melton et al., 2025) attempt to leverage ART dynamics in deeper models, but published evidence of competitive performance on large-scale benchmarks remains limited.

As a side note, it is worth considering that the difficulty in performing good quality credit assignment in deep architectures also applies to the most biologically plausible rehearsal methods. For instance, Sleep Replay Consolidation (Tadros et al., 2022) generates replays from the network's own intrinsic dynamics using local Hebbian plasticity rules, but still relies on backpropagation during the supervised learning phase to solve credit assignment. Moreover, on the MNIST class-incremental benchmark, this method achieves approximately 61% accuracy under incremental learning, far below the approximately 90% reached by batch learning. By comparison, non-biologically plausible approaches such as buffer-based rehearsal, which explicitly store and replay past training examples, achieve substantially stronger performance on challenging continuous-learning benchmarks. For instance, iCaRL (Rebuffi et al., 2017) reports strong results on CIFAR-100, while Dark Experience Replay (DER/DER++) (Buzzega et al., 2020) achieves competitive performance on sequences such as Tiny ImageNet. If biologically plausible rehearsal methods relying exclusively on local plasticity rules and internally generated replay can be brought closer to the performance of buffer-based approaches, they would offer a compelling path toward autonomous continuous learning in deep architectures.

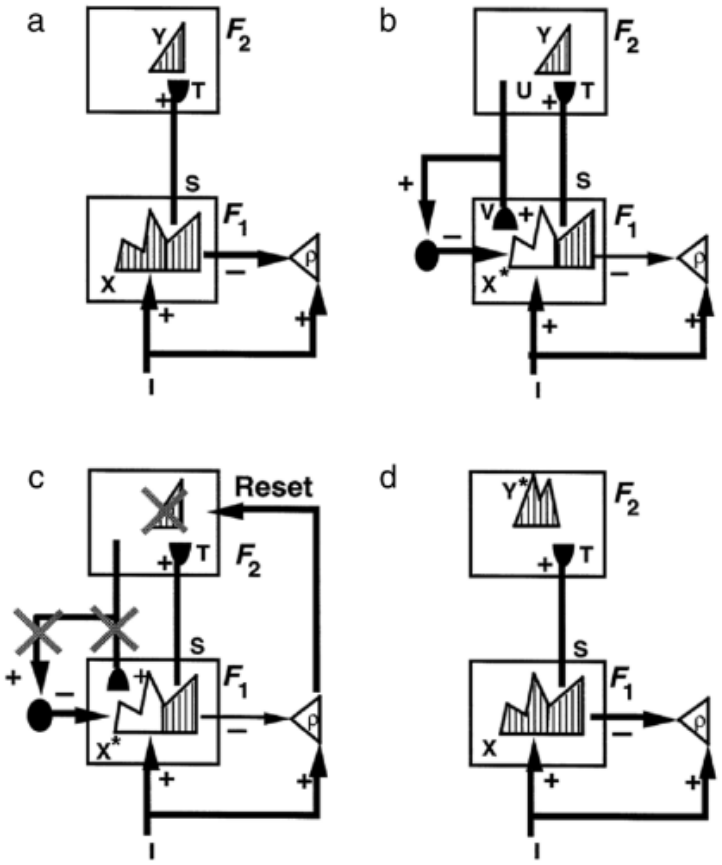

Figure 2.9 – **ART architecture and search cycle.** (a) An input pattern I activates a pattern of activity X across the input population F_1, which propagates through bottom-up adaptive synapses to the recognition population F_2, selecting a candidate category Y. (b) The selected category Y sends a top-down expectation V back to F_1. If the match between V and the input I exceeds the vigilance parameter ρ, a resonance is established and learning proceeds. (c) If the match is below ρ, the orienting subsystem emits a reset signal that inhibits Y. (d) After reset, a new candidate category Y^* is activated, and the matching process repeats until either a sufficiently matching category is found or an uncommitted group of neurons is recruited. Adapted from Grossberg (2013).

2.5.2 . Elastic Weight Consolidation

Elastic Weight Consolidation (EWC), introduced by Kirkpatrick et al. (2017), takes a different approach to the stability–plasticity dilemma. Rather than gating learning through an architectural mechanism as in ART, EWC selectively reduces the plasticity of individual weights according to their estimated importance for previously learned tasks. After training on a given task, the method computes the diagonal of the Fisher information matrix, which quantifies the sensitivity of the network's performance to changes in each weight. Weights with large Fisher information values are deemed important for the current task. When learning a new task, a quadratic penalty added to the loss function discourages modifications to these important weights, anchoring them near their previously learned values while leaving less important weights free to adapt.

The principle of selectively reducing plasticity at specific synapses finds several parallels in biology. In mouse motor cortex, learning different motor tasks elicits Ca^{2+} spikes confined to different apical dendritic branches of the same layer-V pyramidal neuron, which persistently potentiates only the spines that were active during the spike, providing a cellular example of localized consolidation (Cichon & Gan, 2015). This finding provided inspiration for EWC (Kirkpatrick et al., 2017). In cascade models of synaptic memory, repeated potentiation events push a synapse into progressively more stable internal states without altering its observable efficacy (Fusi et al., 2005; Benna & Fusi, 2016). More broadly, metaplasticity, the activity-dependent modification of plasticity rules, provides a general biological principle by which individual synapses adjust their susceptibility to future modification (Abraham, 2008). None of these mechanisms map directly onto the Fisher information computation used in EWC, but they indicate that the brain employs multiple strategies to protect important synaptic configurations.

However, EWC faces several important limitations. First, the method requires explicit knowledge of task boundaries : the Fisher information matrix is computed at the end of each task, and the optimal parameter values must be stored as a reference for the penalty term. Biological learning typically occurs as a continuous stream of experience without clearly delineated task transitions, making this requirement difficult to reconcile with neurobiology. Notably, online variants of EWC mitigate this issue by merging past constraints into a single regularizer that can be updated continuously, making the approach more compatible with settings where task identities (and even sharp task boundaries) are unknown (Huszár, 2018). Second, extending EWC beyond two tasks introduces a systematic bias : maintaining separate penalties anchored at each successive optimum leads to double-counting of past data, which favors earlier tasks (Huszár, 2018). Third, as the number of tasks grows, the accumulated penalties effectively freeze an increasing fraction of the net-

work's parameters. This progressive loss of plasticity eventually prevents the network from learning new tasks altogether, a failure mode that Kirkpatrick et al. (2017) has likened to CF in overloaded Hopfield networks.

In contrast to EWC, rehearsal-based approaches keep the entire synaptic substrate plastic. Rather than penalizing changes to important weights, rehearsal re-exposes the network to previously stored information, forcing the optimization to pass through regions of weight space that support both old and new tasks. The network can thus discover joint synaptic configurations without permanently freezing any subset of its parameters.

2.5.3 . Conclusion

The methods reviewed in this section address the stability–plasticity dilemma by regulating when and where learning is permitted : ART gates plasticity through a matching criterion, while EWC selectively reduces it at synapses deemed important for past tasks. Each carries specific limitations, from ART's category proliferation and credit assignment difficulties in deep architectures to EWC's reliance on explicit task boundaries (partly alleviated by online variants) and progressive parameter freezing.

The following chapter returns to the setting at the core of this thesis : shallow AMNs without hidden neuronal populations. In these networks, overlapping memory representations produce interferences known as crosstalk, and specific methods have been developed to mitigate them, including the gradient descent algorithm (GDA) presented in Section 3.1.2.

3 - Overlapping Memory Representations and Incremental Learning in Hopfield Networks

In this chapter, we analyze the specific challenges that arise when storing correlated patterns in Hopfield Networks (HNs). These correlated patterns correspond to highly overlapping memory representations, as discussed in Sec. 2.2.2. We then present and evaluate existing solutions for storing correlated patterns, discussing their respective advantages and limitations. Finally, we demonstrate that the challenge of storing correlated patterns becomes particularly acute in CL scenarios, where patterns must be incrementally incorporated into the network.

3.1 . The challenge of correlated memories in Hopfield networks

In Sec. 2.2.2, we have been discussing overlapping memory representations, that is, when different memories are encoded using partially shared populations of neurons. The presence of overlapping memory representations in the brain is considered a fundamental organizational principle that supports generalization, feature extraction, and other key properties of NNs.

However, the advantages conferred by overlapping representations come with conceptual and practical challenges. Most synapses remain plastic throughout life, allowing new experiences to reshape existing neural connections. This continuous plasticity implies that every new experience can trigger plasticity events within a neural substrate involved in the storage and processing of other memories (Fu & Zuo, 2011; Fusi & Abbott, 2007), potentially disrupting previously stored information. Together, these two fundamental features, representational overlap and ongoing synaptic plasticity, raise a critical question : How does the brain incorporate new information while safeguarding memories essential for survival? In neural networks (NNs), the degradation of existing memories upon the acquisition of new knowledge is known as interferences. An important disruption of NN functions induced by interferences is called catastrophic interference (in reference to CF). Understanding how NNs can mitigate catastrophic interference remains a central challenge in computational neuroscience and AI (Kirkpatrick et al., 2017).

In HNs, the overlap of memory representations is strictly related to the amount of correlation between binary activity patterns :

Non-overlapping memory representations correspond to anti-correlated binary vectors where different memories assign opposite states to all/most neurons :

$$\mathbf{x}_1 : [+1\ +1\ +1\ -1\ -1\ -1\ -1\ -1]$$
$$\mathbf{x}_2 : [-1\ -1\ -1\ +1\ +1\ +1\ +1\ +1]$$

Partially overlapping representations can occur with uncorrelated binary patterns in which neurons have independent states across memories :

$$\mathbf{x}_1 : [+1\ +1\ +1\ +1\ -1\ +1\ -1\ -1]$$
$$\mathbf{x}_2 : [+1\ +1\ +1\ -1\ +1\ +1\ -1\ +1]$$

Highly overlapping memory representations emerge from correlated binary vectors where many neurons maintain the same states across memories :

$$\mathbf{x}_1 : [+1\ +1\ +1\ +1\ -1\ -1\ +1\ -1]$$
$$\mathbf{x}_2 : [+1\ +1\ +1\ -1\ -1\ -1\ +1\ +1]$$

The degree of correlation between the stored patterns directly determines the amount of interferences in an HN. Storing highly correlated patterns produces greater interferences, while anti-correlated patterns minimize them. Interferences can be quantified through the crosstalk term, as discussed in the following section.

3.1.1 . Crosstalk between memory patterns

Definition of crosstalk

When HNs store multiple patterns, crosstalk quantifies the interferences among stored memories. Here we adopt the definition of crosstalk given in *Introduction to the Theory of Neural Computation* (Hertz et al., 1991). We consider a network that has stored M patterns $\{x^\mu\}_{\mu=1}^{M}$ using the Hebbian rule :

$$W_{ij} = \frac{1}{N} \sum_{\mu=1}^{M} x_i^\mu x_j^\mu .$$

When we present the pattern $\mathbf{x}^\nu$ to the network and compute the drive h_i for unit i, we obtain :

$$h_i = \sum_{j=1}^{N} W_{ij} x_j^\nu = \frac{1}{N} \sum_{j=1}^{N} \sum_{\mu=1}^{M} x_i^\mu x_j^\mu x_j^\nu .$$

We now separate the sum over μ into the special term $\mu = \nu$ and the remaining terms :

$$h_i = \frac{1}{N}\sum_{j=1}^{N} x_i^{\nu}(x_j^{\nu})^2 + \frac{1}{N}\sum_{j=1}^{N}\sum_{\mu\neq\nu} x_i^{\mu}x_j^{\mu}x_j^{\nu}.$$

Since $x_j^{\nu} \in \{-1, 1\}$, we have $(x_j^{\nu})^2 = 1$, yielding

$$h_i = \frac{1}{N}\sum_{j=1}^{N} x_i^{\nu} + \frac{1}{N}\sum_{j=1}^{N}\sum_{\mu\neq\nu} x_i^{\mu}x_j^{\mu}x_j^{\nu}.$$

Therefore,

$$h_i = x_i^{\nu} + \underbrace{\frac{1}{N}\sum_{j=1}^{N}\sum_{\mu\neq\nu} x_i^{\mu}x_j^{\mu}x_j^{\nu}}_{\text{crosstalk } \eta_i^{\nu}}.$$

The first term represents the desired signal that reinforces pattern ν, while the second term is the crosstalk, which corresponds to interferences from all other stored patterns. For successful retrieval of ν, the stability condition requires

$$\text{sign}(h_i) = x_i^{\nu}.$$

A sufficient condition is that the crosstalk does not overpower the signal term :

$$|\eta_i^{\nu}| < 1.$$

The magnitude of crosstalk depends critically on the number of stored patterns and on the overall correlation structure of the stored set. When the M patterns are uncorrelated, the crosstalk terms tend to average out. However, when the stored patterns share systematic similarities—i.e., when certain groups of neurons tend to have the same state across multiple patterns—the crosstalk terms can constructively interfere, leading to retrieval failure. Fig. 3.1 shows three uncorrelated random patterns typically used in theoretical analyses of HNs, where each unit independently takes values in $\{-1, +1\}$ with equal probability. In contrast, Fig. 3.2 presents three MNIST digit patterns that exemplify a set of correlated patterns. The MNIST patterns exhibit clear structural similarities, most notably the shared black background (representing $x_i = -1$) that surrounds each digit. This common feature creates systematic correlations between patterns, as large groups of neurons maintain the same state across different stored memories.

The impact of these correlations on crosstalk is quantified in Fig. 3.3, which compares the average crosstalk magnitude in a network storing random patterns with that in a network storing MNIST patterns. In both cases, patterns are stored using the Hebbian plasticity rule. The higher crosstalk observed for MNIST patterns shows that structured real-world data induce significantly

greater interferences than the idealized random patterns extensively examined in theoretical work.

In biological terms, correlated memory patterns would translate into populations of neurons that are consistently activated (spiking or having elevated firing rates) or suppressed across various memories. If we consider that biological NNs might operate according to principles similar to those governing HNs, then these overlapping memory representations would necessarily lead to significant crosstalk. In the following section, we examine in detail how the crosstalk induced by correlated patterns poses fundamental challenges for memory storage and retrieval in HNs.

Figure 3.1 – **Three uncorrelated random binary patterns typically used in theoretical analyses of Hopfield networks**. Each pattern is generated by independently assigning states $x_i \in \{-1, +1\}$ with equal probability to each neuron. White pixels represent $x_i = +1$ and black pixels represent $x_i = -1$.

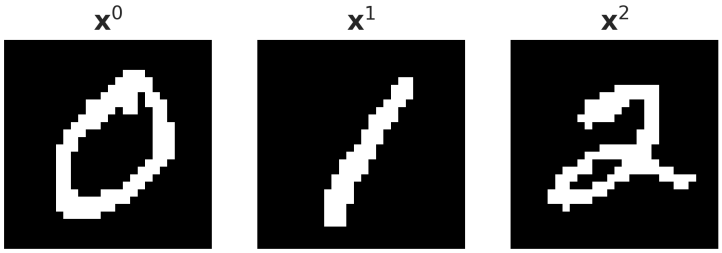

Figure 3.2 – **Three MNIST digits stored in a network**. Analogous to Fig. 3.1 but with binarized images of handwritten digits. Unlike the random patterns, these digit images exhibit structure and are correlated ; most notably, the shared black background surrounding each digit creates systematic overlap between patterns, leading to increased crosstalk in the network.

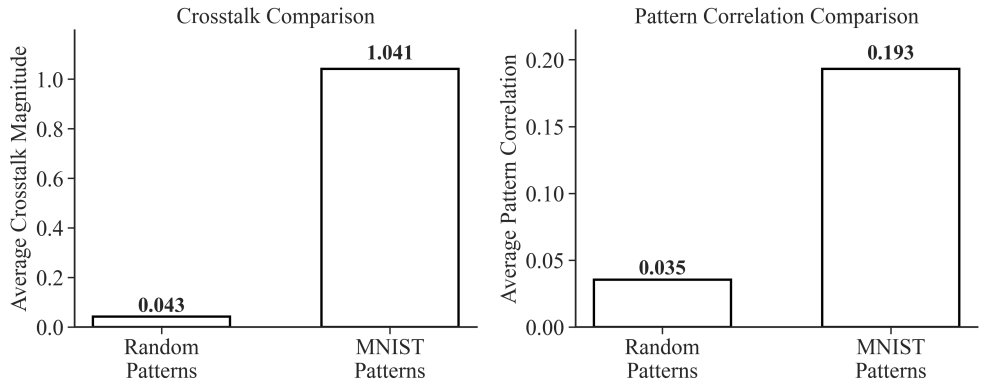

Figure 3.3 – **Crosstalk and pattern correlation comparison.** For networks storing sets of 3 patterns using the Hebbian plasticity rule, the average crosstalk magnitude is computed as $\frac{1}{MN}\sum_{\mu=1}^{M}\sum_{i=1}^{N}\left|h_i^\mu - x_i^\mu\right|$, where $h_i^\mu = \sum_j W_{ij}x_j^\mu$ is the local field at unit i when pattern μ is presented, and x_i^μ is the signal term. This measures the average interferences across all units and all stored patterns. Pattern correlation is the mean absolute Pearson correlation coefficient between all pattern pairs. The random patterns are those in Fig. 3.1, while the MNIST patterns are those in Fig. 3.2. MNIST patterns induce higher average crosstalk due to greater correlations, demonstrating how a structured, correlated pattern set can severely amplify interferences in HNs.

Spurious patterns and catastrophic forgetting

Crosstalk between stored patterns in HNs leads to a fundamental problem : the emergence of spurious attractors. These are stable states that were not explicitly stored during learning but are generated by interferences among memories. They can be seen as false memories that occupy the network's memory space and hinder the retrieval of true memories, compromising performance.

A pattern $\mathbf{x}^\mu$ is stable if, when the pattern is presented to the network, the condition $\text{sign}(h_i) = x_i^\mu$ holds for all i. Even if a plasticity rule, such as Hebbian plasticity, can enforce this stability for a given pattern, the crosstalk induced by storing multiple patterns can drive some units' local fields h_i to flip sign ("bit flips"). This phenomenon can lead to the emergence of unwanted fixed points—so-called spurious attractors (or spurious stable states).

There are two important families of spurious attractors :

- **k-mixture states.** These arise even when storing uncorrelated memories. For an odd k, a typical mixture has the form

$$\mathbf{x}^{\text{mix}} = \text{sign}\big(\pm\mathbf{x}^{\mu_1} \pm \mathbf{x}^{\mu_2} \pm \cdots \pm \mathbf{x}^{\mu_k}\big),$$

  and can satisfy the stability condition as the crosstalk contributions align constructively in the local fields. Mixture states are a classic and well-characterized source of spurious patterns (Amit et al., 1985a). Adding stochasticity to the dynamics of HNs is sufficient to eliminate these spurious stable states (Hertz et al., 1991). In this work, we do not discuss solutions based on stochastic networks.
- **Prototype (cluster–center) states.** When stored memories are correlated, Hebbian learning tends to stabilize unlearned states that approximate the mean of a class or cluster of related patterns (Gorman et al., 2017; McAlister et al., 2024). Fig. 3.4 illustrates a prototype spurious state in a network that stores three MNIST patterns. When presented with a noisy cue, the network fails to converge to any of the original stored patterns, instead settling into a spurious state that resembles an average of the stored patterns. These spurious stable states emerge specifically due to the correlated structure of the MNIST dataset, which generates significant crosstalk between stored patterns. Adding stochasticity to the network dynamics is insufficient to eliminate these stable states (McAlister et al., 2024).

The emergence of prototype states presents a significant challenge for the storage of highly overlapping representations in HNs (Marzo & Iannelli, 2023; Hertz et al., 1991). Understanding how to mitigate them is therefore a necessary step toward implementing more robust AMNs. In the next section, we explore methods that reduce crosstalk between stored memories in HNs and, therefore, mitigate the emergence of spurious attractors.

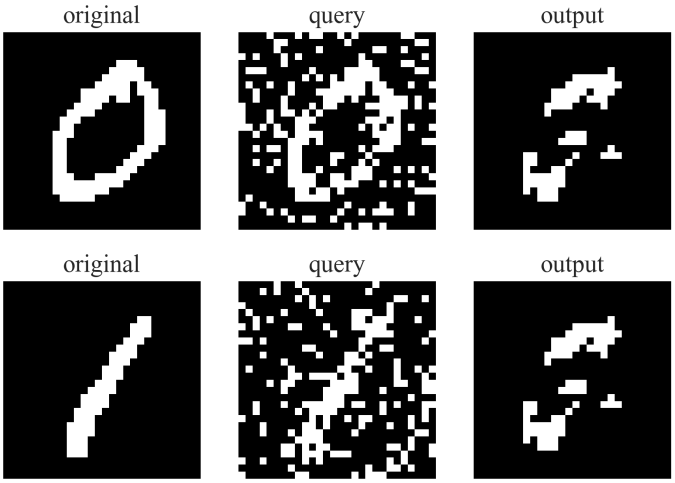

Figure 3.4 – **Spurious pattern formation in a Hopfield network storing correlated patterns.** Two retrieval attempts from a network storing three MNIST digits (the three stored digits are those in Fig. 3.2). The top row corresponds to the first attempt and the bottom row to the second attempt. Each row shows : (left) the original stored pattern, (middle) a corrupted query with 40% noise, and (right) the network output after convergence. In both cases, instead of recovering the original pattern, the network converges to the same prototype spurious mixture state, averaging features from multiple stored digits—illustrating how correlated patterns lead to false memories.

3.1.2 . Solutions to crosstalk

Multiple methods have been developed to reduce crosstalk between correlated memories, each with distinct strengths and limitations. Before examining specific approaches, we establish the evaluation criteria that guide our selection and assessment of these methods.

1. **Locality of dynamics.** We require that all computations rely exclusively on information accessible to individual neurons and their immediate synaptic connections. This excludes operations that require simultaneous access to global network states or distant, unconnected network components.
2. **Limited complexity of the plasticity rules.** We prioritize plasticity rules with limited complexity, in line with Occam's razor : simpler theories should be explored before adding complexity if necessary. A full justification for this epistemic stance is beyond the scope of this thesis. In addition, simplicity is crucial for neuromorphic implementations. Hardware realizations of NNs face constraints in circuit complexity, power consumption, and fabrication precision. Simple plasticity rules can be implemented with basic circuit elements, require fewer transistors, and thereby reduce energy requirements and improve the overall feasibility of neuromorphic systems.
3. **Efficiency in reducing crosstalk.** We evaluate methods based on their effectiveness in reducing interferences between correlated memories. Methods that only marginally improve on Hebbian learning are considered insufficient for storing highly correlated patterns.

Suitability for continuous learning (CL) will be addressed in the next section. Considering this final criterion provides the full context for the main work of this thesis, presented in Chap. 4.

We now examine three specific methods : the pseudoinverse method (nonlocal, simple, and highly effective), the Storkey update rule (local, simple, but less effective), and the gradient descent algorithm (local, simple, and highly effective). We also briefly review some discarded approaches and why they were discarded.

This overview does not aim to provide an exhaustive survey of all methods that improve the storage of correlated patterns in HNs. Rather, it illustrates the types of techniques commonly employed and their typical advantages and limitations, establishing the context necessary to understand the solutions presented in this thesis.

Pseudo-inverse method

The pseudo-inverse method was introduced in the Hopfield framework by Personnaz et al. (1985), who developed the "projection rule" for error-free pattern storage, and later formalized by Kanter & Sompolinsky (1987), who demonstrated exact recall for linearly independent patterns. This method is often regarded as the theoretical gold standard for eliminating crosstalk in Hopfield networks (HNs).

The method constructs the weight matrix by directly solving the linear system that makes each stored pattern a fixed point of the network dynamics. Given a set of M patterns $\{\mathbf{x}^\mu\}_{\mu=1}^M$ to be stored, it seeks weights W_{ij} such that

$$\mathbf{x}^\mu = \mathbf{W}\mathbf{x}^\mu \quad \forall \mu.$$

Arranging all patterns as columns in a matrix $\mathbf{X} = [\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^M]$, we obtain

$$\mathbf{X} = \mathbf{W}\mathbf{X}.$$

A solution is

$$\mathbf{W} = \mathbf{X}\mathbf{X}^+ = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top,$$

where $\mathbf{X}^+$ denotes the Moore–Penrose pseudoinverse of $\mathbf{X}$. When the patterns are linearly independent and $M < N$, an equivalent form uses the pattern correlation matrix

$$C_{\mu\nu} = \frac{1}{N}\sum_{k=1}^{N} x_k^\mu x_k^\nu,$$

yielding

$$\mathbf{W} = \frac{1}{N}\mathbf{X}\mathbf{C}^{-1}\mathbf{X}^\top \quad \Longrightarrow \quad W_{ij} = \frac{1}{N}\sum_{\mu,\nu=1}^{M} x_i^\mu\,(\mathbf{C}^{-1})_{\mu\nu}\,x_j^\nu.$$

This method is considered a gold standard because it precisely enforces the equations that ensure zero crosstalk among stored memories. By construction, when pattern $\mathbf{x}^\nu$ is presented to the network, the local field (drive) satisfies $h_i = \sum_j W_{ij}x_j^\nu = x_i^\nu$ exactly, eliminating all interferences. This holds for all stored patterns simultaneously, achieving optimal storage without interferences (Kanter & Sompolinsky, 1987).

However, the method is explicitly nonlocal : the matrix inversion $(\mathbf{X}^\top\mathbf{X})^{-1}$ is a global computation that depends on the entire pattern set and cannot be accessed by local neural dynamics. Each synaptic weight W_{ij} depends not only on the activities of neurons i and j across patterns, but also on the full correlation structure of the memory set via $(\mathbf{C}^{-1})_{\mu\nu}$.

Despite its efficiency, the pseudo-inverse rule must be discarded under our criteria : because it requires global computations, a biological implementation or a neuromorphic realization is unrealistic.

Storkey update rule

The Storkey learning rule (Storkey, 1997) provides a method for storing patterns while partially compensating for crosstalk. When adding a new pattern $\mathbf{x}^{M+1}$ to a network that already stores M patterns, the weight update is

$$W_{ij}^{M+1} = W_{ij}^{M} + \frac{1}{N} x_i^{M+1} x_j^{M+1} - \frac{1}{N} x_i^{M+1} h_{ij}^{M+1} - \frac{1}{N} h_{ji}^{M+1} x_j^{M+1}, \tag{3.1}$$

where W_{ij}^{M} is the synaptic matrix after storage of M patterns, and the local fields are defined as

$$h_{ij}^{M+1} = \sum_{\substack{k=1 \\ k \neq i,j}}^{N} W_{ik}^{M} x_k^{M+1}.$$

The Storkey rule incrementally subtracts crosstalk from previous memories using local neuronal fields. The terms $x_i^{M+1} h_{ij}^{M+1}$ and $h_{ji}^{M+1} x_j^{M+1}$ are corrections that compensate for interferences induced by learning the new pattern $\mathbf{x}^{M+1}$. These correction terms are computed from the local fields h_{ij}^{M+1}, which capture the network's response to the new pattern given the existing connections, thereby reducing first-order crosstalk.

This method respects locality : each weight update involves only the states of the presynaptic neuron j and the postsynaptic neuron i (x_j^{M+1} and x_i^{M+1}), together with local fields (h_{ij}^{M+1}) computed at the postsynaptic site.

The computation of h_{ij}^{M+1} requires only information about the existing weights connecting neuron i to its neighbors and the current pattern to be stored. This information is considered available at the synaptic level without nonlocal coordination. Experimental studies indicate that analogous local fields are computed and used by biological neurons to modulate synaptic plasticity. Typically, homeostatic mechanisms such as synaptic normalization depend on such computations (Turrigiano et al., 1998; Turrigiano, 2012).

The rule has limited complexity : it requires no global computations or complex normalization procedures. However, it also has important limitations. Although it eliminates first-order interferences, residual higher-order interferences remain. The rule does not fully remove crosstalk, particularly when patterns are highly correlated or when the number of stored patterns approaches the network's capacity. These higher-order terms, though smaller in magnitude than the primary crosstalk, can accumulate and still lead to retrieval errors, especially in networks that store highly correlated patterns (Storkey & Valabregue, 1999).

Gradient descent algorithm

The gradient descent algorithm (GDA) iteratively adjusts synaptic weights to minimize the error between the desired and actual network states for all stored patterns (Diederich & Opper, 1987). The following procedure stores the set of patterns $\{\mathbf{x}^\mu\}_{\mu=1}^M$:

Algorithm 3 Gradient descent for storing correlated patterns in DHNs

Initialize $W_{ij} = 0$ for all i, j
repeat
  $\delta \leftarrow 0$
  for each pattern μ **do**
    for each neuron i **do**
      Compute the current drive : $h_i^\mu = \sum_j W_{ij} x_j^\mu$
      Compute the error : $e_i^\mu = x_i^\mu - h_i^\mu$
      for each synapse j **do**
        $\Delta W_{ij} \leftarrow \alpha\, e_i^\mu\, x_j^\mu$
        Update weights : $W_{ij} \leftarrow W_{ij} + \Delta W_{ij}$
        $\delta \leftarrow \max(\delta, |\Delta W_{ij}|)$
      end for
    end for
  end for
until $\delta < \varepsilon$

Typically, $\alpha < 0.01$ and $\varepsilon < 0.01$, which allows the convergence of the learning process and avoids oscillations near the optimum. This method is highly effective, typically achieving near-zero crosstalk, with performance comparable to the pseudo-inverse method. By iteratively minimizing the retrieval error for each pattern, the algorithm finds a weight configuration that ensures that all stored patterns are stable fixed points with minimal crosstalk.

The method is local because each synaptic update depends only on the current local field h_i^μ at neuron i, the states of the presynaptic and postsynaptic neurons (x_j^μ and x_i^μ), and the error e_i^μ computed at that neuron. The update rule $\Delta W_{ij} = \alpha\, e_i^\mu\, x_j^\mu$ uses only information available at the synapse : the postsynaptic local field and the pre and postsynaptic activity. The postsynaptic local field is considered to be available as a homeostatic regulatory signal, as in the Storkey learning rule.

To understand how GDA reduces crosstalk, consider the general case where the network's drive deviates from the desired pattern. When presenting pattern μ, the drive of neuron i can be decomposed as

$$h_i^\mu = x_i^\mu + \mathscr{D}_i^\mu,$$

where $\mathscr{D}_i^\mu$ represents any deviation from the target state x_i^μ. If any crosstalk exists between patterns, it is included in this deviation term.

The error computed by the algorithm is

$$e_i^\mu = x_i^\mu - h_i^\mu(\text{old}) = -\mathscr{D}_i^\mu,$$

where $h_i^\mu(\text{old})$ is the drive of neuron i for pattern μ before the weight update.

Thus, the error directly measures the negative of the deviation. The weight update then becomes

$$\Delta W_{ij} = \alpha\, e_i^\mu\, x_j^\mu = -\alpha\, \mathscr{D}_i^\mu\, x_j^\mu.$$

After this update, the new drive for pattern μ at neuron i is

$$h_i^\mu(\text{new}) = h_i^\mu(\text{old}) + \sum_j \Delta W_{ij} x_j^\mu = h_i^\mu(\text{old}) + \alpha\, e_i^\mu \sum_j (x_j^\mu)^2.$$

Assuming bipolar coding $x_j^\mu \in \{\pm 1\}$ (so $(x_j^\mu)^2 = 1$) and N inputs (including the diagonal), we obtain

$$h_i^\mu(\text{new}) = h_i^\mu(\text{old}) + \alpha N\, e_i^\mu = (1 - \alpha N)\, h_i^\mu(\text{old}) + \alpha N\, x_i^\mu.$$

To see how this drives the deviation to zero, substitute $h_i^\mu = x_i^\mu + \mathscr{D}_i^\mu$ into the update equation :

$$x_i^\mu + \mathscr{D}_i^\mu(\text{new}) = (1 - \alpha N)\big(x_i^\mu + \mathscr{D}_i^\mu(\text{old})\big) + \alpha N\, x_i^\mu.$$

Expanding and simplifying the x_i^μ terms yields

$$\mathscr{D}_i^\mu(\text{new}) = (1 - \alpha N)\, \mathscr{D}_i^\mu(\text{old}).$$

This is the key result : each iteration multiplies the deviation by the factor $(1 - \alpha N)$. For convergence, we require $0 < \alpha N < 1$, which constrains the learning rate to $\alpha < 1/N$. After t iterations, the deviation becomes

$$\mathscr{D}_i^\mu(t) = (1 - \alpha N)^t\, \mathscr{D}_i^\mu(0).$$

As $t \to \infty$, the term $(1 - \alpha N)^t \to 0$, ensuring $\mathscr{D}_i^\mu \to 0$ and thus $h_i^\mu \to x_i^\mu$. The convergence is exponential : for instance, with $\alpha N = 0.1$, the deviation is reduced to $0.9^{10} \approx 0.35$ of its initial value after 10 iterations, and to $0.9^{50} \approx 0.005$ after 50 iterations.

If this deviation $\mathscr{D}_i^\mu$ corresponds to crosstalk between patterns, the algorithm effectively removes this crosstalk, driving it to zero through repeated iterations. Through this process across all patterns, GDA modifies the weight matrix to minimize $|\mathscr{D}_i^\mu|$ for all μ, achieving very low crosstalk even for highly correlated pattern sets.

The GDA satisfies the constraints stated earlier : it maintains locality of computation, uses a simple error-driven plasticity rule, and achieves excellent efficiency in reducing crosstalk. The simplicity of the update rule—a product of local error and presynaptic activity—makes it suitable for neuromorphic implementation.

Importantly, the need to loop multiple times over all patterns can be considered a form of rehearsal, as introduced in Sec. 2.4.2. We can say that the GDA organizes an interleaved learning scheme of the form ABCABCABC. . ., here for the storage of three patterns : A, B, and C.

Fig. 3.5 illustrates the performance of each method by comparing residual crosstalk across different plasticity rules. The results demonstrate that GDA and the pseudo-inverse method achieve minimal crosstalk, nearly eliminating interferences between stored patterns. The Storkey rule reduces crosstalk significantly compared to standard Hebbian learning but retains measurable interferences. This empirical comparison underscores why GDA provides a good balance between biological plausibility, ease of implementation on a neuromorphic substrate, and storage performance.

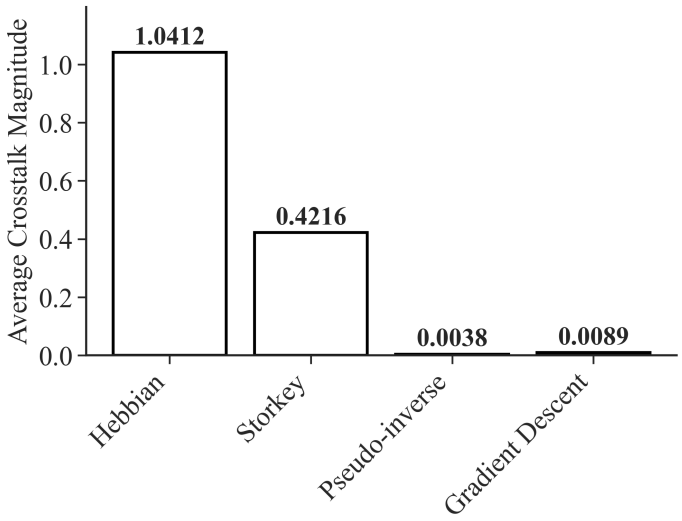

Figure 3.5 – **Comparison of residual crosstalk across different plasticity rules.** For each method, the average crosstalk magnitude for 3 stored MNIST patterns is shown. GDA and the pseudo-inverse method achieve minimal crosstalk, while Hebbian and Storkey rules exhibit higher levels of crosstalk. Because of the strong correlations present in MNIST patterns, the Storkey rule is not very effective at reducing crosstalk compared to the other methods.

Other discarded solutions

Beyond the methods presented earlier, many more solutions exist for storing correlated patterns in HN-like AMNs. Although these approaches perform well at storing correlated patterns, they do not satisfy our criteria for locality, simplicity, and biological interpretability. We mention them here to acknowledge their contributions to the field.

Modern Hopfield networks. Modern HNs, also known as dense associative memories, represent a significant advancement in AMNs. These networks achieve polynomial storage capacity, $M = \mathcal{O}(N^n)$ for some $n > 1$, compared to the linear scaling of classical HNs. They also show superior performance in pattern retrieval even with highly correlated data (Krotov & Hopfield, 2016).

However, we discard these models because of their reliance on nonlocal computations. The dynamics of these networks typically relies on operations such as

$$\mathbf{s}^{t+1} = \mathbf{X}\,\mathrm{softmax}(\beta\,\mathbf{X}^\top \mathbf{s}^t),$$

where $\mathbf{X}$ is the matrix containing all stored memory vectors as columns and $\mathbf{s}^t$ is the state of the network at iteration t. Here, softmax normalization requires computing a partition function over all stored patterns, a global operation incompatible with local synaptic mechanisms.

More recently, Krotov & Hopfield (2021) demonstrated that modern HNs admit an equivalent formulation as networks with hidden neurons connected through pairwise synaptic interactions, thereby eliminating the need for many-body synaptic junctions. This reformulation restores a degree of biological plausibility by relying solely on pairwise synaptic connections; however, it does so at the cost of introducing additional hidden neuronal populations. We do not consider such architectures here, as we restrict this work to the simplest network architecture : single-layer fully connected networks without hidden neuronal populations. Generalizing to architectures with hidden layers constitutes a natural direction for future work.

Threshold-linear Hebbian networks. Threshold-linear associative networks employ neurons with rectified-linear activation functions and have been shown to achieve near-optimal theoretical capacities (Schönsberg et al., 2021). These networks demonstrate that, by allowing neurons to operate with continuous-valued outputs above a threshold rather than binary states, the storage of correlated patterns can be significantly improved. However, achieving these capacities requires complex normalization procedures that substantially increase the implementation burden. The Hebbian learning rule for threshold-linear units requires careful control of the mean and variance of neural activity through appropriate rescaling of signal-to-noise ratios. To prevent un-

bounded weight growth and ensure stable pattern retrieval, these networks typically require multiple normalization steps : mean subtraction to center the activity distributions, variance normalization to standardize the spread of activations, and often spherical normalization to constrain weight vectors to lie on a unit sphere. While each of these operations can, in principle, be implemented using only local information at individual neurons, their combination creates a multi-stage normalization pipeline that runs counter to our preference for simple mechanisms.

Conclusion

The GDA emerges as the best option among the methods examined here : it maintains locality, employs a simple error-driven update rule, and achieves performance comparable to the theoretically optimal pseudo-inverse method in eliminating crosstalk.

However, like the pseudo-inverse rule, the GDA requires iterating over all patterns for convergence. When adding a new pattern $\mathbf{x}^{M+1}$ to a network that already stores M patterns, the algorithm must revisit all previously stored patterns to ensure that they remain stable fixed points.

In the next section, we discuss how this need to revisit previously stored patterns hinders incremental learning, a form of continuous learning (CL) in AMNs.

3.2 . The challenge of incremental learning in Hopfield Networks

We have discussed various methods to reduce crosstalk between stored memories ; we now ask whether these methods can support incremental learning.

3.2.1 . What is incremental learning ?

Incremental learning is the ability of a memory system to be updated sequentially while maintaining its functional integrity (Van De Ven et al., 2022). In the context of AMNs, it means expanding a network's memory set pattern-by-pattern using online updates that rely solely on information about the new pattern to be stored. This continual incorporation must occur without requiring external access to previously stored patterns.

We formalize incremental learning in AMNs as follows. The system receives data in the sequence $A^{(1)} \to B^{(2)} \to C^{(3)} \to D^{(4)}$, where $X^{(t)}$ denotes that only pattern X is available at time t. When the system has access to pattern B, it does not simultaneously have access to patterns A or C from any external source. When storing pattern C, patterns A and B must be safeguarded so that they remain retrievable. By contrast, in a non-incremental (offline/batch) paradigm, the system has simultaneous access to all patterns $\{A, B, C, D\}$ at all times.

This notion closely matches what the broader machine-learning and computational-neuroscience literature calls *sequential learning* (see Sec. 2.4.3). Both belong to the broader scope of continuous learning (CL) discussed in the Introduction. We retain the term *incremental learning* here, as it is established within the associative-memory literature, while noting this equivalence to help readers navigate between research communities.

As discussed in the Introduction, incremental learning is particularly desirable because it mirrors biological learning : mammals must continuously acquire new information throughout life without re-experiencing past inputs, suggesting that learning modalities are inherently incremental.

3.2.2 . Evaluation of candidate methods in the incremental regime

We now assess each previously introduced method with respect to its inherent support for incremental storage.

- **Pseudo-inverse :** In its standard form, the pseudo-inverse rule is not incremental, as it requires access to the full set of patterns for a batch matrix inversion.
- **Storkey update rule :** Fully incremental and local. When a new pattern is trained, the updated weights depend only on the previous weights and the new pattern. However, it only modestly reduces crosstalk and struggles with highly correlated patterns, leading to degraded perfor-

mance as pattern similarity and load increase.

- **GDA :** Each weight update is local, but convergence to stable fixed points typically requires iterative rehearsal over all stored patterns. Without access to previous patterns, the algorithm cannot guarantee that old memories remain stable after incorporating new ones ; therefore, strictly incremental learning without rehearsal is not possible.

A clear pattern emerges : methods that excel at eliminating crosstalk (pseudo-inverse, gradient descent) rely on the availability of information about the whole pattern set. Their strength lies in using this knowledge to minimize interferences and ensure that each memory forms a well-separated attractor.

It is important to note that, if used naively to store patterns incrementally by simply modifying the weights of a network that already contains memories, both the pseudo-inverse and GDA can destroy previously stored stable states to accommodate new ones. This outcome has been termed *catastrophic plasticity* (a form of catastrophic forgetting). Catastrophic plasticity is illustrated in Sec. 4.8.6.

3.2.3 . Conclusion : The need for autonomous rehearsal

The fundamental challenge is the apparent incompatibility between strong crosstalk reduction and strict incremental learning under locality constraints. GDA is particularly promising because it uses local updates and achieves strong performance, but its rehearsal requirement seems to preclude strict incrementality.

As the next chapter will show, this limitation can be overcome if the network can autonomously reinstate its stored patterns, rather than relying on an external agent to replay previously learned memories.

This insight motivates the model developed in this thesis, which couples autonomous rehearsal with local, gradient-based plasticity. By enabling the network to internally rehearse the patterns required for GDA convergence, we achieve efficient incremental learning with minimal crosstalk in a biologically grounded framework.

4 - Autonomous Rehearsal and Incremental Learning in Continuous Hopfield Networks

This chapter includes the article produced and published during this thesis (Saighi & Rozenberg, 2025). This work addresses the problem of incremental learning in associative memory networks (AMNs). We propose an autonomous rehearsal method that enables our model to autonomously retrieve previously stored memories. This ability can be used to prevent catastrophic forgetting (CF), allowing the integration of new information with existing knowledge without the need for an external memory lists. In this article, we employ the term continuous learning (CL) rather than incremental learning to make the link with the broader CL literature more explicit.

The chapter concludes with an additional section discussing in more detail our rationale for using continuous Hopfield networks in this work and presenting further results on the autonomous retrieval mechanism.

4.1 . Abstract

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 plastic self-inhibition 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 towards a more biologically plausible mechanism for rehearsal and continuous learning.

4.2 . Author summary

Catastrophic forgetting, that is, when acquiring new knowledge seriously degrades previously learned information, remains a fundamental challenge in machine learning, affecting systems from simple associative networks to complex language models. An approach widely studied to mitigate forgetting is rehearsal, in which prior stored information is periodically replayed, an idea supported by neurophysiological evidence of memory consolidation during sleep. In this work, we show that networks with adaptation can spontaneously recall stored memories without external guidance. This built-in retrieval mechanism allows for the incorporation of new memories while preserving older ones, opening a path toward a biologically inspired solution to the problem of catastrophic forgetting.

4.3 . 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.

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). In the mammalian nervous system, spontaneous memory replays occur during sleep (Robins, 1995; Louie & Wilson, 2001; Peyrache et al., 2009; Fauth & Van Rossum, 2019; Tononi & 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 bio-inspired neural networks demonstrates that short-term synaptic depression can facilitate spontaneous rehearsal of neural assemblies (Fauth & 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 & Opper, 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 ar-

tificial 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 bioplausibility 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 & Opper (1987) (Diederich & Opper, 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., 1985b).

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 reduce the attraction toward 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).

4.4 . Methods

4.4.1 . 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 :

$$c\frac{du_i}{dt} = \sum_j W_{ij}v_j - \frac{u_i}{r} \tag{4.1}$$

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 $|\frac{du}{dt}|_\infty < \varepsilon$, where $\varepsilon = 10^{-6}$.

4.4.2 . 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 $\mathbf{x}^\mu$ is defined as a binary vector such that $\mathbf{x}^\mu = (x_1^\mu, x_2^\mu, \ldots, x_N^\mu)$ 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 :

$$x_i^\mu = \begin{cases} 1 & \text{if } v_i(t) > 0.5 \\ 0 & \text{otherwise.} \end{cases} \tag{4.2}$$

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

$$\tilde{u}_i^\mu = \begin{cases} +u_{\text{target}}^\mu & \text{if } x_i^\mu = 1 \\ -u_{\text{target}}^\mu & \text{if } x_i^\mu = 0 \end{cases} \tag{4.3}$$

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

Inspired by previous work on DHN (Diederich & Opper, 1987), we introduce, in Alg. 4, 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. 6.1.

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 & Opper, 1987). Here, we rely on numerical evidence showing the network's ability to successfully query and revisit stored patterns.

Algorithm 4 Gradient descent for the storage of correlated patterns (GDA)

Initialize $W_{ij} = 0$ for all i, j
repeat
  for each pattern μ **do**
    Compute the target potential $\tilde{u}_j^\mu$ (Eq. 4.3) for each neuron j
    Compute the expected potential : $\hat{u}_i^\mu = r \sum_j W_{ij} \sigma(\tilde{u}_j^\mu)$
    Update weights : $W_{ij} \leftarrow W_{ij} - \alpha(\hat{u}_i^\mu - \tilde{u}_i^\mu) r \sigma(\tilde{u}_j^\mu)$
  end for
until $\|\Delta W\|_\infty < \varepsilon$

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 (Eq. 4.2) when the system reaches convergence. The querying procedure is detailed in Alg. 8 (Appendix 6.2) and illustrated in Fig. 4.1.

Similarly to the model proposed by Diederich and Opper (1987), the symmetry of the weight matrix W is not enforced in the GDA scheme. In classical Hopfield networks, this constraint is typically imposed to prevent unwanted oscillatory dynamics (Hopfield, 1982). A notable feature of the GDA is that it consistently stores patterns as attractor states despite the absence of enforced symmetry (Diederich & Opper, 1987). We observed the same phenome-

non in our simulations. This represents a strength of the algorithm, as enforcing symmetry without an explicit underlying mechanism would break the locality of the plasticity rules. In biological neural networks, the synapse from neuron A to neuron B does not necessarily have access to the state or sign of the reciprocal synapse from neuron B to A.

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 (Sec. 4.4.4), limiting the appearance of false memories as spurious states, which are often encountered in DHNs (Amit et al., 1985a). A variant of the DHN with stochastic units, the SHN (Amit et al., 1985a), 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.

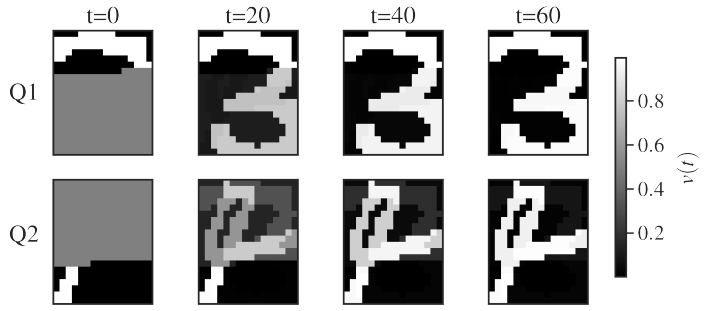

Figure 4.1 – Querying of two binary-patterns representation of the handwritten digit "3" and "4" from the MNIST dataset. The network is of size 20×16 as each neuron codes for a pixel. From black to white, the color represents the rate $v_i(t)$ of each unit. At $t = 0$ the network is initialized with the part of the units set to the target value as described in Appendix 6.2. 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".

4.4.3 . Continuous incorporation of correlated memories

While the GDA effectively enables the storage of correlated memories, its implementation in Alg. 4 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 & Cohen, 1989; Robins, 1995; Kirkpatrick et al., 2017; Shen et al., 2023). By repeatedly processing all patterns, the algorithm can find a weight matrix **W** that properly separates the patterns, despite their correlations (Diederich & Opper, 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 towards 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 Alg. 5. 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.

Algorithm 5 Continuous incorporation of correlated patterns through rehearsal of the whole memory set

1: **Input :** CHN trained on p patterns using the GDA
2: **Given :** New patterns $\{x^{p+1}, x^{p+2}, \dots\}$ to be stored
3: Retrieve set $\{\mathbf{x}\}$ of stored patterns through autonomous retrieval (AR) introduced in Sec. 4.4.4
4: Update pattern set : $\{\mathbf{x}\} \leftarrow \{\mathbf{x}\} \cup \{x^{p+1}, x^{p+2}, \dots\}$
5: Apply GDA to store updated pattern set

4.4.4 . 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 Eq. 4.1 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.

$$c\frac{du_i}{dt} = \sum_j W_{ij}v_j - A_i v_i - \frac{u_i}{r} \tag{4.4}$$

$$A_i \leftarrow A_i + \beta v_i(t_f) \tag{4.5}$$

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 & Cheong, 2017; Peron & Gabbiani, 2009). Repeated stimulation progressively reduces firing activity in the neuron, mirroring the dynamics produced by our adaptation 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 & Froemke, 2015).

After a given memory is retrieved (i.e., when the network falls into an attractor), the attraction toward that state is decreased through the adaptation term (Eq. 4.5). Consequently, once the 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.

Fig. 4.2 illustrates the sequential recovery of two stored patterns and the modification of the network dynamics induced by the procedure. The geometry of the vector field 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 corresponding to each patterns. 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 (Sec. 4.4.2). 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, Eq. 4.4, 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 (Eq. 4.1) until convergence. The entire procedure is detailed in Alg. 6.

It is important to note that correlated patterns and patterns with various sparsity levels induce similar vector fields. In all cases, we observe a separatrix crossing the neutral state and lying in the middle of the attraction basins. It appears to be an emergent property of the gradient descent algorithm, appropriately shaping the dynamical landscape to balance the attraction strength of memory states from the neutral state. The construction of correlated patterns is discussed in Sec. 4.5. The impact of sparsity on the network behavior is discussed in Sec. 4.8.3.

Fig. 4.3 provides a visualization of AR for binary-patterns 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 $(\mathbf{A})$. 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 $\mathbf{W}$. 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 Fig. 4.4, 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 Fig. 4.4(top)), the system converges sequentially to the attractors 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 Fig. 4.4(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 sixth iteration of 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 Fig. 4.4(bottom)), the network successfully retrieves some stored states, but mostly falls 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 & Wilson, 2001; Peyrache et al., 2009; Fauth & Van Rossum, 2019; Tononi & Cirelli, 2014).

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 App. 6.4. As implementing adaptation requires less computation and parameters, the following work focuses only on the adaptation model.

Algorithm 6 Autonomous retrieval (AR)

Input : Trained network weights W_{ij}, number of iterations k, plasticity rate β
Initialize : $A_i = 0$, $\{\mathbf{x}\} = \emptyset$
while $j < k$ **do**
    Set neutral initial conditions : $\mathbf{u}(t=0) = \mathbf{0}$
    Biased phase : Integrate Eq. 4.4 until convergence to the state $\mathbf{u}(t_b) = \mathbf{u}_b$
    Free phase : from the state $\mathbf{u}(t_b)$, integrate Eq. 4.1
until convergence ▷ optional
    Read pattern $\mathbf{x}^\mu$ from final state $\mathbf{v}(t_f)$ via thresholding (Eq. 4.2)
    Update retrieved pattern set : $\{\mathbf{x}\} \leftarrow \{\mathbf{x}\} \cup \{\mathbf{x}^\mu\}$
    Update inhibitory weights : $A_i \leftarrow A_i + \beta v_i(t_f)$
    $j \leftarrow j + 1$
end while
Return : $\{\mathbf{x}\}$

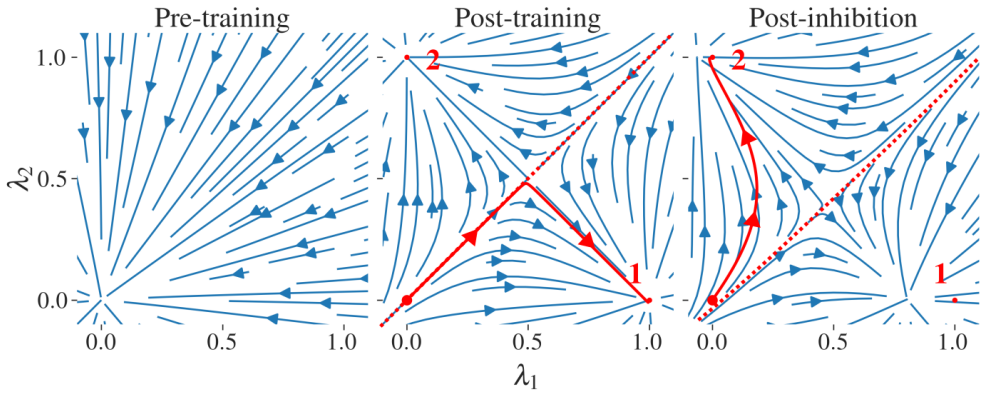

Figure 4.2 – **Stream plots of the CHN in a 2D subspace spanned by two pattern states.** We examine a network with two stored patterns, labeled "1" and "2", and their stable states $\tilde{\mathbf{u}}^1$ and $\tilde{\mathbf{u}}^2$; these define a two-dimensional subspace parametrized by $\tilde{\mathbf{u}}(\lambda_1, \lambda_2) = \lambda_1 \tilde{\mathbf{u}}^1 + \lambda_2 \tilde{\mathbf{u}}^2$. At each point $\tilde{\mathbf{u}}(\lambda_1, \lambda_2)$ in this subspace, we compute the full N-dimensional time derivative via Eq. 4.4 and project it back onto the $\{\tilde{\mathbf{u}}^1, \tilde{\mathbf{u}}^2\}$-plane to obtain the plotted flow. $\tilde{\mathbf{u}}^1$ and $\tilde{\mathbf{u}}^2$ 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 (1 and 2) becomes a stable attractor; the neutral state $\mathbf{u}_N = \mathbf{0}$ (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 (red dotted line), 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.

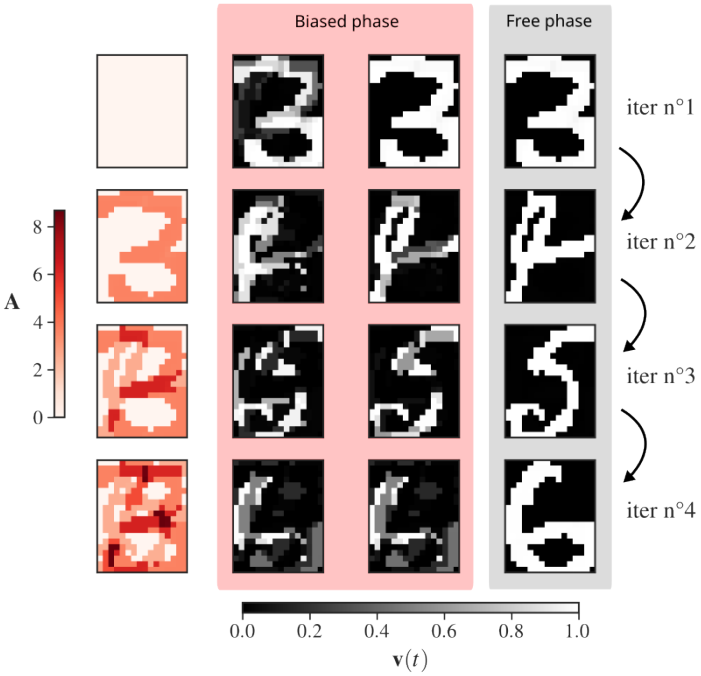

Figure 4.3 – Recovery of four binary-patterns representations of handwritten digits from the MNIST dataset, “3”, “4”, “5”, “6”. The network is of size 20×16 as each neuron codes for a pixel. Left panels (in red) : Evolution of adaptation of each neuron A_i. Right panels (pink and gray) : snapshots of 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 Alg. 6. 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 states deformation at the end of the biased phase, highlighting the need of the free phase.

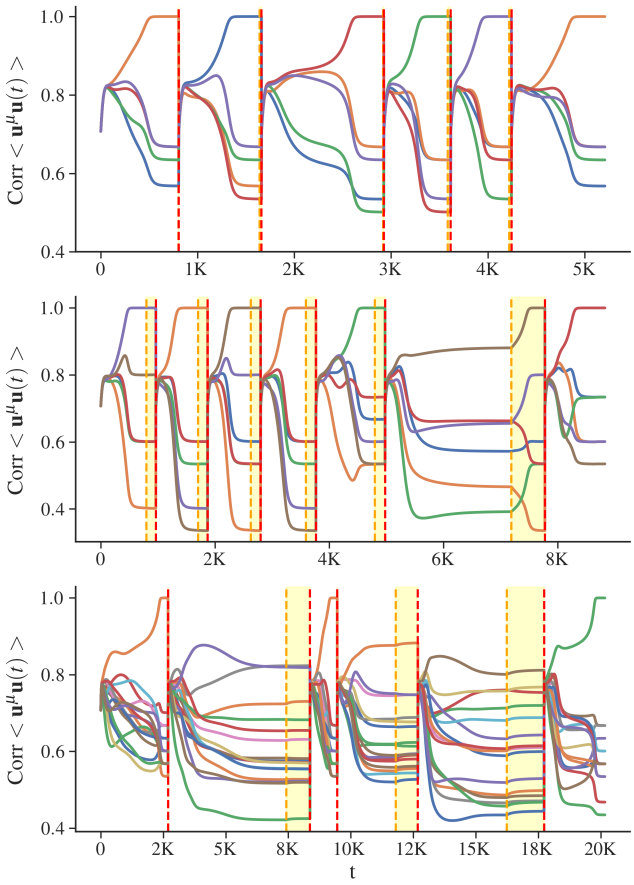

Figure 4.4 – Visualization of AR (Alg. 6). The x-axis shows the number of time steps for simulations of the CHN, with all query iterations of AR concatenated. The y-axis shows the correlation between the state of the network and any stored state, with each color corresponding to a specific memory. Vertical red dashed lines separate successive memory retrieval sequences. At the beginning of a memory recall adaptation $\mathbf{A}$ 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 in a low-load network : 5 patterns stored in a network of 60 units. Each memory is visited once before all are successfully recovered. **(Middle)** Recovery dynamics in a critically loaded network : 5 patterns stored in a network of 30 units. Some memories are revisited multiple times before all are recovered. **(Bottom)** Recovery dynamics in a high-load network : 16 patterns stored in a network of 60 units. Recovery fails as spurious patterns are visited, reflected by low correlation values.

4.5 . 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 is assigned a random binary parent pattern, where each bit is independently set to 0 or 1 with equal probability. Each pattern of a given set is generated by randomly choosing a fraction $(1-\rho)$ of bits from the parent pattern and randomizing them. The randomization process sets each selected bit to 0 or 1 with equal probability, independently of its original value, as described in Alg. 9 (App.6.3).

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.

Fig. 4.5 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 Alg. 5. The retrieval improves with network size as 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 & 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 6.5, Fig. 6.3, this push towards 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. Fig. 4.6 indicates the existence of a trade-off : Smaller β values require more iterations to retrieve all patterns but lower the chances of retrieving spurious patterns, while larger values accelerate pattern retrieval at the cost of increasing the probability of encountering a spurious 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 than presented here only lead to the need for more iterations to retrieve all patterns. For $\beta = 0$ the network converges toward one of the stored pattern states, and, so long we start near the same initial (neutral) state, it will always fall back into this attractor after each reset, we call this attractor the dominant attractor (or the dominant pattern) of the neutral state.

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 Alg. 4 increases with the memory load (Fig. 4.7). 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 & Opper, 1987). We can argue that this increase in computational cost may come from two factors. First, training a CHN through GDA (Alg. 4) inherently requires more iterations than training DHNs. Second, we observed that a smaller convergence parameter ε in Alg. 4, 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 in dynamics induced by adaptation during retrieval. These observations led to the adoption of a very small $\varepsilon = 10^{-6}$, which demanded more iterations when performing the GDA.

Finally, Fig. 4.8 gives a visual summary of the continuous incorporation of new correlated memories allowed by our AR mechanism (Alg. 5). The whole learning scheme can be divided into two phases : an awake phase during which new patterns are collected, and a sleep phase during which previous patterns are retrieved using AR. The storage of patterns collected during both phases requires the use of the GDA. During this storage, the adaptation terms are turned off for the GDA to converge.

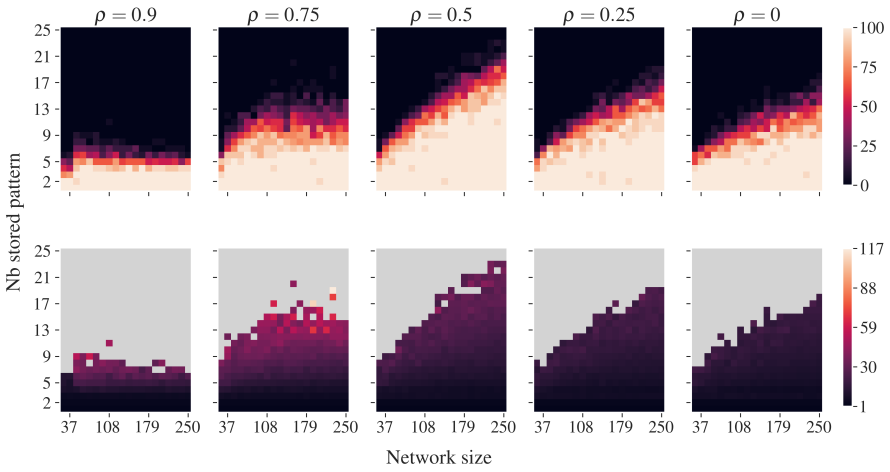

Figure 4.5 – Recovery capacity of AR for various correlations, modulated by ρ. Data are averaged over 20 simulations with different pattern sets. Correlated patterns are generated using Alg. 9 (Appendix 6.3). For networks of various sizes and numbers of stored patterns, we employ Alg. 6 until all patterns 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.

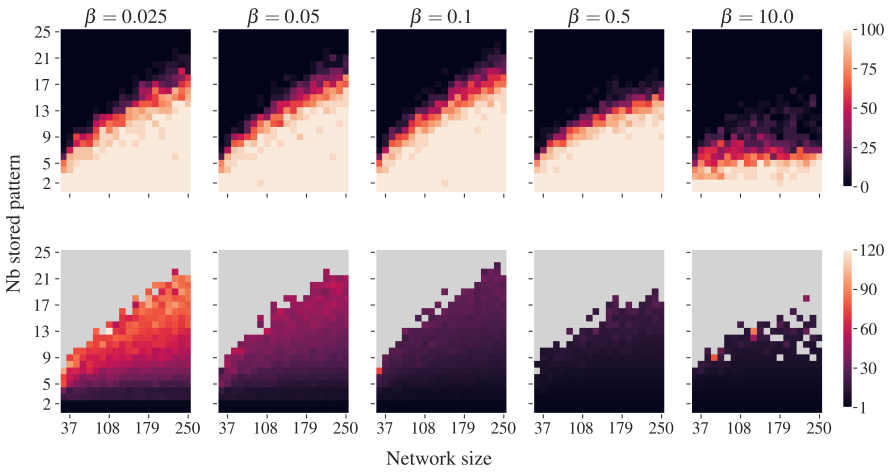

Figure 4.6 – Similar to Fig. 4.5 but for various β values. Correlated patterns are generated with a ρ of 0.5. Lower β values require more iterations to recover the complete set of stored patterns. Higher β values diminish the number of iterations needed but increase the probability of finding spurious states when the load is important.

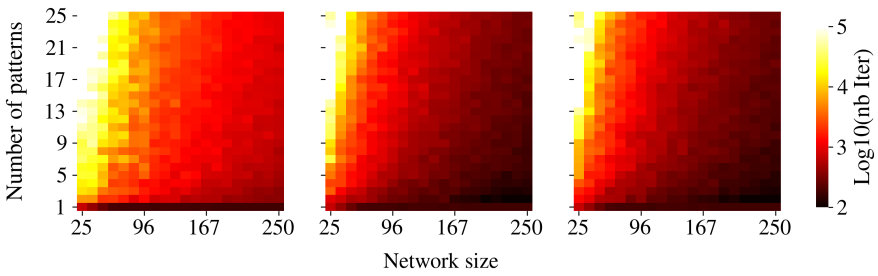

Figure 4.7 – Convergence time of the GDA. Color intensity shows the number of iterations required to store a given number of patterns using GDA (Alg. 4). Data are averaged over 20 simulations for various pattern sets. ρ is equal to 0.5 for all simulations.

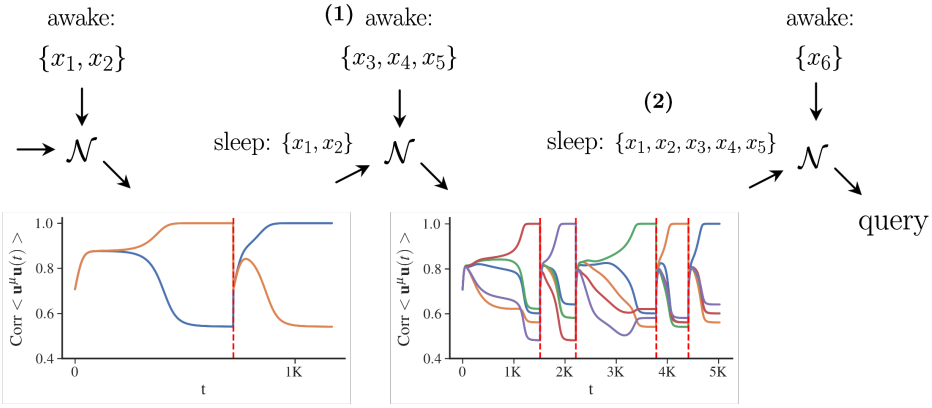

Figure 4.8 – **Continuous incorporation of correlated patterns.** Visualization of Alg. 5. $\mathcal{N}$ denotes the network. The awake phase corresponds to the incorporation of new memories, while the sleep phase corresponds to AR. **(1)** Memories added during the second awake phase. **(2)** Memories recovered during the second sleep phase. Memories retrieved during sleep, along with newly encountered ones, are stored in the network using the GDA. At the end of the simulation, querying the network enables the retrieval of the six incrementally stored memories $\{x_1, \ldots, x_6\}$.

4.6 . 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 need to recall patterns from an external list. 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., 1985a). However, our findings revealed the rather unexpected feature that moderate correlation levels ($\rho = 0.5$) 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 towards 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.

The mechanism presented here is robust to moderate amounts of noise, but it breaks down and induces spurious patterns when strong stochasticity is introduced. The impact of noise on the recovery dynamics has been tested by adding a stochastic term to the dynamics of each neuron :

$$c\frac{du_i}{dt} = \sum_j W_{ij}v_j - A_i v_i - \frac{u_i}{r} + D\eta_i(t), \tag{4.6}$$

where $\eta_i(t) \sim \mathscr{U}(-1,1)$ and $D < 1$ is the noise scaling factor. This added stochasticity does not alter the recovery dynamics for $D < 0.1$ but induces spurious patterns for higher values. Importantly, this stochasticity does not by itself induce the exploration of various memory states when $\beta = 0$. We have also tested slightly randomizing the neutral state, with $u_i(0) = D\eta_i$. This approach yields results identical to adding stochasticity to neural dynamics : convergence remains unchanged for $D < 0.1$, while spurious patterns emerge at higher values. This suggests that, even though the neutral state lies near the separatrix, in high dimensions, the dominance of an attractor cannot be easily disrupted through stochasticity alone, at least for the continuous model and learning scheme considered here. While this statement may depend on the specific values of N and D, a quantitative exploration of this feature is beyond the scope of the present work.

Previous work in computational neuroscience 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 allows the sequential reactivation of memories observed during sleep and resting states (Wilson & 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 for context-dependent activity of the network as observed in experimental settings (Kuchibhotla et al., 2017). Biological neural circuits might, therefore, employ similar mechanisms to navigate complex, correlated memory spaces. In our model, adaptation, or self-inhibition, is only potentiated at the end of each memory replay. This behavior contrasts with the plasticity dynamics in the brain, which are considered to occur continuously. This schematic implementation is intended to demonstrate the functional role of inhibition in guiding memory exploration, rather than to reproduce the precise temporal dynamics of neuronal plasticity. Another discontinuity in the plasticity dynamics comes from the fact that, during the training of the network on a new memory corpus, self-inhibition has to be removed to avoid disturbance of the storage process. Implementing a model that more closely matches the continuous dynamics observed in the brain would therefore represent a natural extension of this work.

The ability of our network to transition from one memory state to another critically depends on the reset of the dynamics to the neutral state. In the brain, this reset mechanism may likely be induced by observed physiological processes. During sleep, for instance, cortical neurons exhibit bistable dynamics, alternating between depolarized UP states characterized by sustained firing and hyperpolarized DOWN states with neuronal silence (Jercog et al., 2017; Luczak et al., 2007; Steriade et al., 1993). These UP states, lasting hundreds of milliseconds to seconds, provide windows for memory replay, while DOWN states could naturally separate distinct memory activations. Another natural extension of our work would be to incorporate specific neural mechanisms responsible for such network resets, such as post-inhibitory rebound, to implement a more biologically plausible version of our model.

4.7 . 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.

4.8 . Addition

This section provides supplementary analysis of the autonomous retrieval (AR) mechanism presented above and compares our approach with existing methods for incremental learning in Hopfield networks. We first justify our choice of continuous Hopfield networks for this study. We then characterize the dynamics at the neutral state and the factors that determine the order in which patterns are explored during AR, before examining the influence of pattern sparsity and leak parameters on AR performance. We next reproduce and extend McCallum's pseudorehearsal experiments and compare them with our continuous incorporation (CI) method. We then broaden the comparison to other incremental learning approaches, including Hebbian plasticity and the Storkey learning rule, applied to larger networks. Finally, we test the robustness of the mechanism when patterns are stored at different distances from the neutral state.

Throughout this section, we employ synthetic binary patterns with controlled statistical properties rather than real-world datasets such as MNIST or CIFAR. This choice is deliberate. Real datasets exhibit complex and heterogeneous statistics : patterns vary in sparsity, correlation structure, and higher-order dependencies in ways that are difficult to disentangle. In a single-layer network, these multiple sources of heterogeneity would confound our analysis, making it impossible to identify which specific property affects retrieval performance. Using synthetic patterns with controlled sparsity and correlation, we can isolate the influence of individual factors and thereby characterize the fundamental properties of our model. For the storage of real data, networks with hidden layers would be more appropriate candidates : hidden representations can learn to extract relevant features and normalize variability across inputs, thereby conferring robustness to the diverse correlations and sparsity levels present in real data. Extending the AR mechanism to such architectures represents a natural direction for future work.

4.8.1 . The use of CHNs

In this study, we introduce a model based on CHNs and present a gradient descent algorithm (GDA) that enables the storage of correlated memories, similar to the approach developed for DHNs (Sec. 3.1.2).

The use of CHNs is motivated by the observation that classical DHNs (Sec. 2.3.4) tend to produce spurious attractor states when employing the specific recall protocol used in this work (uncued convergence from a neutral state). In the context of DHNs, a neutral state corresponds to a completely random initialization, where each neuron's binary state is independently drawn with equal probability of being -1 or 1. These spurious attractor states emerge even when GDA is used for storage. Fig. 4.9 illustrates such spurious attractor states, showing network convergence to what appears to be a prototype spurious state, despite minimal crosstalk between memory states.

Initializing the network from a neutral state appears to create more opportunities for settling into spurious states compared to initialization from a partial cue, as is typically done in AMNs (Fig. 2.3). This observation is corroborated by numerous studies identifying the initial distance from any given stored state as a crucial factor for successful convergence (Zhang & Zhang, 2008). Intuitively, the closer the initial state is to a stored state, the better defined the dynamical landscape that enables convergence.

In contrast, CHNs consistently converge to stored states from any initial state when using our adapted GDA. Although the precise mechanism remains unclear, we hypothesize that CHNs maintain a smooth dynamical landscape even far from any stored memory state because of the continuous nature of their dynamics, which facilitates reliable convergence.

Interestingly, additional work not presented in this thesis demonstrates that stochastic Hopfield networks (SHNs) can also consistently converge from a neutral state to a stored pattern when the storage is performed with the GDA, but only in specific conditions. SHNs are variants of DHNs where units update stochastically according to a temperature parameter, following a probabilistic rule based on the Boltzmann distribution (Amit et al., 1985a). To achieve convergence, we employed a simulated annealing scheme in which the temperature is initially elevated (producing purely random activity) and then gradually decreased until the dynamics become deterministic.

It is crucial to emphasize that this approach differs from using a fixed temperature to eliminate certain spurious attractors when storing uncorrelated patterns with Hebbian plasticity, as demonstrated by Amit et al. (1985a). To our knowledge, successful retrieval of correlated patterns stored with GDA in SHN from a neutral pure random queries has not been previously demonstrated. Notably, this procedure bears striking similarity to approaches used in modern diffusion models for generating high-quality synthetic data (Song & Ermon, 2020; Ho et al., 2020). In these models, removing the simulated an-

nealing dynamics similarly induces retrieval of states that can be interpreted as prototype-like patterns.

These observations suggest that the MR procedure presented in this work may be applicable to SHNs, although this avenue remains unexplored.

Figure 4.9 – A discrete Hopfield network (DHN) converging to a spurious attractor state when initialized from a neutral state (random noise). Despite using GDA for storage of 5 MNIST patterns, the network fails to retrieve any stored pattern and instead settles into a spurious state that resembles a prototype-like pattern.

4.8.2 . Dynamics at the neutral state and impact of inhibition

As described above, when initialized at the neutral state $\mathbf{u}_0 = \mathbf{0}$, the network dynamics converge deterministically toward one of the stored patterns. We denote the corresponding baseline activity $\bar{\mathbf{v}} = \sigma(\mathbf{u}_0) = (1/2, \ldots, 1/2)^T$.

Following retrieval, the potentiation of self-inhibition weakens the attraction toward that pattern, thereby favoring convergence to a different stored pattern upon reset. To characterize this behavior, we quantify the tendency of the network to evolve toward each stored pattern from the neutral state.

For each stored pattern $\mathbf{x}^\mu$, we define the unit vector pointing from the neutral state toward the pattern :

$$\hat{d}_\mu = \frac{\mathbf{x}^\mu - \bar{\mathbf{v}}}{\|\mathbf{x}^\mu - \bar{\mathbf{v}}\|} \tag{4.7}$$

We define the *push* toward pattern μ, denoted P_μ, as the component of the instantaneous velocity in the direction of that pattern :

$$P_\mu = \left\langle \left. \frac{d\mathbf{v}}{dt} \right|_{\mathbf{u}_0}, \hat{d}_\mu \right\rangle \tag{4.8}$$

A positive value $P_\mu > 0$ indicates that the instantaneous velocity has a positive component in the direction of pattern μ.

Fig. 4.10 shows P_μ for each stored pattern for a network without self-inhibition (Eq. 4.1). The values are all positive and of comparable magnitude, suggesting that the GDA shapes the weight matrix such that no pattern is strongly favored over the others from the neutral state. However, small residual differences determine which memory is visited when no inhibition biases recovery.

We can now examine how self-inhibition, potentiated after each pattern retrieval, modifies the dynamics of the network at the neutral state (Eq. 4.4). Recall that after each retrieval, self-inhibition is potentiated according to $A_i \leftarrow A_i + \beta v_i(t_f)$ (Eq. 4.5), where $v_i(t_f)$ is the firing rate of neuron i at convergence. Fig. 4.11 tracks the push P_μ toward each stored pattern as inhibition accumulates during AR (Alg. 6). Each row corresponds to one iteration, with the retrieved pattern shown in red and previously retrieved patterns in orange. In this example, the network retrieves the pattern with the largest push at each iteration. We observe that after retrieval of a given pattern, self-inhibition reduces the push toward this pattern most strongly. This results in a sequential exploration of all the stored patterns.

Fig. 4.11 suggests that the pattern retrieved at each iteration corresponds to the one with the largest push value. However, this observation is based on a single simulation. To assess whether push reliably predicts retrieval across different network configurations, we systematically measure the probability that a pattern will be retrieved based on its push at each AR iteration. The results are shown in Fig. 4.12.

To obtain Fig. 4.12, we rank the stored patterns by decreasing push value at each AR iteration (rank 0 = highest push, rank $M-1$ = lowest push). Consider a network that stores $M=4$ patterns. At iteration 0, no inhibition has accumulated ($\mathbf{A}=\mathbf{0}$) : we compute the push toward each pattern, rank them in decreasing order (rank 0 for the highest push, rank 3 for the lowest), and observe which pattern is retrieved. Suppose pattern $\mu=3$ has the highest push (rank 0) and is indeed retrieved ; we record a retrieval event at rank 0. At iteration 1, inhibition has been potentiated predominantly on neurons active in pattern $\mu=3$, modifying all push values. We recompute the push toward each pattern and re-rank accordingly : suppose pattern $\mu=1$ now has the highest push (rank 0). If it is retrieved, we record another event at rank 0. This procedure continues until all four patterns are retrieved, yielding four retrieval events for this simulation.

Fig. 4.12 displays the resulting probability distribution over ranks. We aggregate retrieval events across all AR iterations and across 100 independent simulations for each memory load $M\in\{2,3,5,7,10\}$, with $N=200$ neurons and correlation parameter $\rho=0.5$. Iterations yielding spurious states are excluded. For small loads ($M=2,3,5$), the highest-push pattern (rank 0) is retrieved with probability close to 1, indicating that push is a near-perfect predictor of retrieval. As the memory load increases ($M=7,10$), the retrieval probability for rank 0 decreases progressively, while patterns with smaller push are retrieved with non-negligible probability.

These results indicate that, at low loads, the instantaneous velocity at the neutral state largely determines the retrieval outcome : the pattern with the largest push is reliably retrieved. At higher loads, push becomes less informative, and patterns with smaller push values are retrieved with non-negligible probability. This degradation of predictive power occurs approximately at memory loads where spurious states begin to appear (Figs. 4.5 and 4.6). Although further investigation would be necessary, this suggests that retrieval degrades when the instantaneous velocity at the neutral state does not point predominantly toward a single stored pattern.

The results are similar for correlation (ρ) varying from 0.2 to 0.8 and self-inhibitory potentiation (β) varying from 0.1 to 0.0001, although not illustrated here.

The analysis presented in this section reveals that the behavior of AR depends on the load. At low memory loads, the push metric accurately predicts retrieval : the network sequentially visits stored patterns in an order largely determined by the instantaneous velocity at the neutral state. As the load increases, this correspondence weakens. In the following section, we examine the influence of pattern sparsity on the push from the neutral state and on the performance of the AR mechanism.

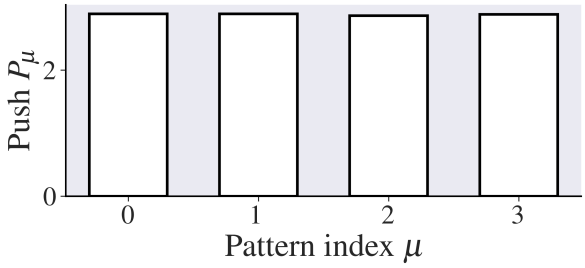

Figure 4.10 – Bar height correspond to P_μ the projection of the instantaneous velocity at the neutral state onto the direction of each stored pattern. The approximately equal values indicate balanced initial attraction toward all patterns. Network size $N = 200$, number of stored patterns $M = 4$, correlation parameter $\rho = 0.5$, patterns generated using Alg. 9.

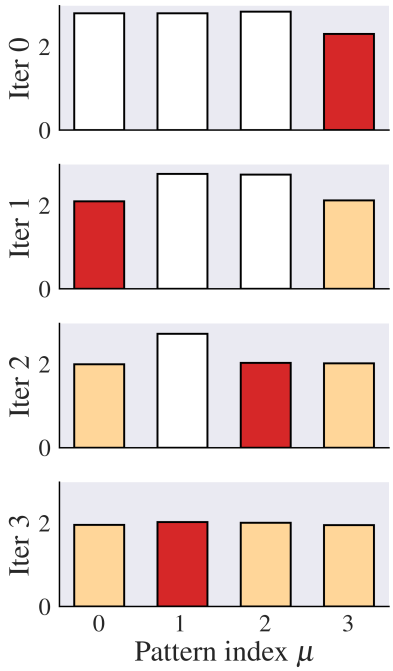

Figure 4.11 – Evolution of the push P_μ toward each stored pattern during autonomous retrieval. Each row corresponds to one AR iteration. Red : pattern retrieved at the current iteration; orange : patterns retrieved at previous iterations; white : patterns not yet retrieved. The push decreases more strongly for retrieved patterns than for non-retrieved patterns due to the selective potentiation of self-inhibition. Same simulation parameters as Fig. 4.10, with $\beta = 0.1$.

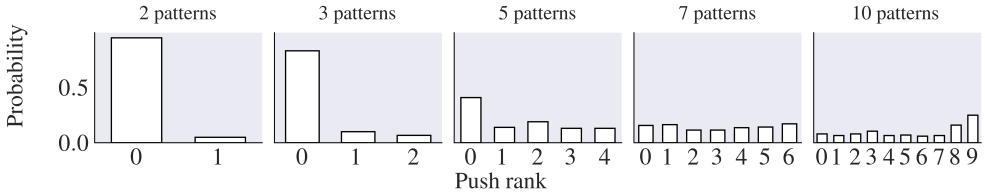

Figure 4.12 – Probability that a pattern of a given push rank is retrieved during AR, for memory loads $M \in \{2, 3, 5, 7, 10\}$. At each AR iteration k, push values are recomputed using the current inhibition state, and patterns are ranked in decreasing order (rank 0 = highest push). For low loads, rank 0 is retrieved with probability close to 1. As load increases, retrieval probability spreads across ranks. Network size $N = 200$, correlation parameter $\rho = 0.5$, inhibitory potentiation rate $\beta = 0.0001$, 100 simulations per configuration.

4.8.3 . Impact of pattern sparsity

Neural codes in the brain exhibit varying degrees of sparsity depending on the brain region and stimulus type (Willmore et al., 2011; Barth & Poulet, 2012; Vonderschen & Chacron, 2011). Similarly, representations in deep neural networks often display heterogeneous sparsity across layers (Wild & Anderson, 2024; Li et al., 2023). This motivates an investigation of how sparsity influences retrieval dynamics in our model.

Sparsity influence on the push from the neutral state

The results presented in the preceding section were obtained using patterns with sparsity $s = 0.5$. Patterns with a given sparsity s are generated by independently setting each bit x_i^μ to 1 with probability $(1-s)$ and to 0 with probability s. We now examine how varying s affects the push P_μ from the neutral state. In this section, patterns with heterogeneous sparsity are generated independently, without any imposed correlation (other than the correlation induced by sparsity itself).

To observe the effect of sparsity, we store five patterns in a network of size $N = 600$, each pattern having a distinct sparsity value : $s \in \{0.40, 0.45, 0.50, 0.55, 0.60\}$. Fig. 4.13 shows the push P_μ towards each stored pattern. The push increases monotonically as the sparsity decreases, indicating that patterns with more active units are more strongly favored by the dynamics at the neutral state.

Fig. 4.14 shows the evolution of P_μ during AR. Initially, patterns are retrieved in order of decreasing push (increasing sparsity). However, at iteration 4, a previously retrieved pattern is revisited despite having a lower push than at least one unretrieved pattern. This illustrates that the push provides a bias rather than a deterministic prediction of the retrieval order.

To quantify this bias induced by pattern sparsity, we run multiple independent simulations and measure two metrics : the number of times each pattern is visited during AR (visit count), and the iteration at which each pattern is first retrieved (first visit iteration). The results are shown in Figs. 4.15 and 4.16.

For each memory load $M \in \{4, 6, 8\}$, we generate M patterns with sparsities uniformly distributed in the interval $[0.3, 0.7]$. Networks of size $N = 200$ are trained using the GDA, and AR is run until all patterns are retrieved or a spurious state is encountered. We repeat this procedure for 30 independent simulations per configuration and group the results into several sparsity bins.

Fig. 4.15 shows the distribution of visit counts across sparsity bins. Patterns with lower sparsity are visited more frequently during AR. Fig. 4.16 shows the first visit iteration for each sparsity bin. Patterns with lower sparsity are retrieved earlier in the AR sequence. These results confirm that the bias observed in the push metric translates into a systematic effect on the retrieval dynamics.

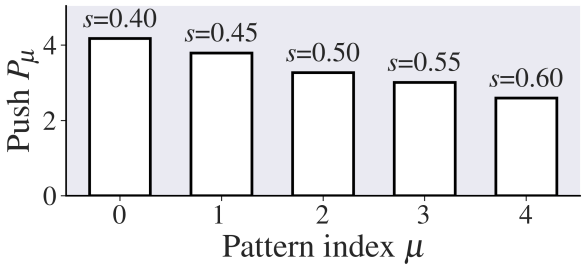

Figure 4.13 – Push P_μ toward each stored pattern for patterns with heterogeneous sparsities. Each pattern has a distinct sparsity $s \in \{0.40, 0.45, 0.50, 0.55, 0.60\}$. Network size $N = 600$, number of stored patterns $M = 5$.

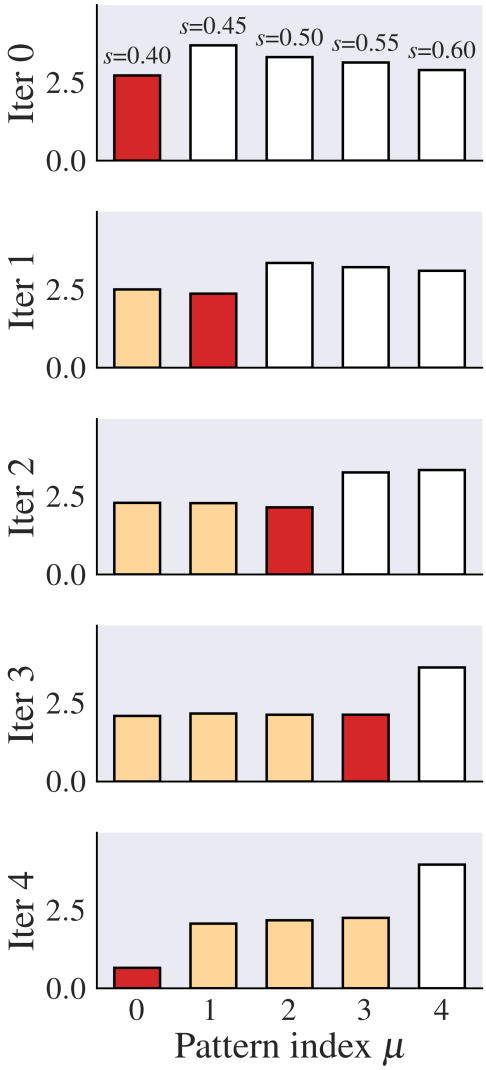

Figure 4.14 – Evolution of push P_μ during AR for patterns with heterogeneous sparsities. Red bars indicate the pattern retrieved at the current iteration; orange bars indicate previously retrieved patterns. The sparsest pattern ($s = 0.60$) is not retrieved within five iterations.

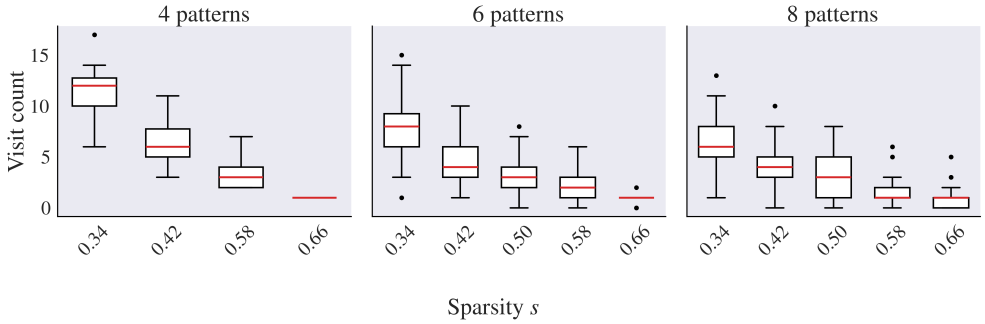

Figure 4.15 – Distribution of visit counts during AR as a function of pattern sparsity, for memory loads $M \in \{4, 6, 8\}$ with sparsities uniformly distributed in the interval $[0.3, 0.7]$. Patterns are grouped into several sparsity bins. Red lines indicate the median; boxes span the interquartile range; whiskers extend to the most extreme data points within 1.5 times the interquartile range; black dots indicate outliers. Network size $N = 200$, 30 simulations per configuration.

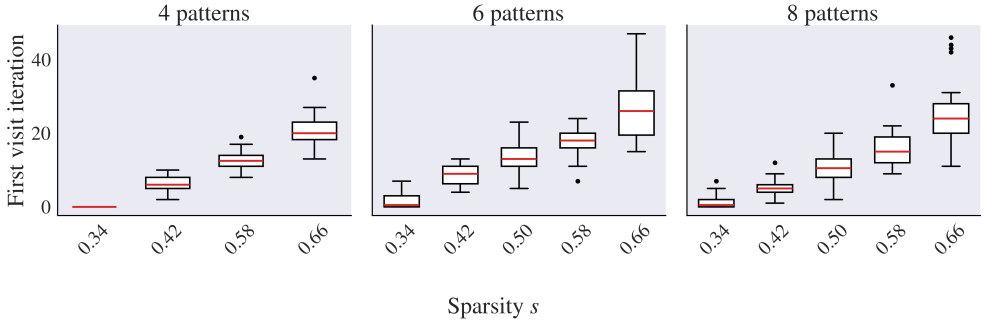

Figure 4.16 – Similar to Fig. 4.15 but visualizing distribution of first visit iteration during AR as a function of pattern sparsity. Network size $N = 200$, 30 simulations per configuration.

Sparsity influence on retrieval capacity

The preceding analysis demonstrates that sparsity influences the push toward individual patterns. We now examine how sparsity affects the overall retrieval capacity of the network. Specifically, we consider two scenarios : (i) corpora in which all patterns share the same sparsity, and (ii) corpora containing patterns with heterogeneous sparsities. (Recall that s denotes the fraction of inactive units).

We first consider corpora in which all stored patterns share the same sparsity s. Fig. 4.17 illustrates, for various sparsity values, the loads for which AR allows systematic recovery of all stored patterns. We observe that optimal retrieval occurs for moderately sparse codes ($s \approx 0.7$), with performance degrading for both higher and lower sparsity.

We next consider corpora containing patterns with heterogeneous sparsities. Fig. 4.18 illustrates the impact of heterogeneous sparsity on AR performance. For each pattern μ, the sparsity s^μ is drawn independently from a uniform distribution centered at 0.5 with range Δs, i.e., $s^\mu \sim \mathscr{U}[0.5 - \Delta s/2,\ 0.5 + \Delta s/2]$. For instance, $\Delta s = 0.4$ corresponds to sparsities uniformly distributed in the interval $[0.3, 0.7]$. Unlike the uniform sparsity case, no optimal intermediate value emerges : retrieval performance degrades monotonically as Δs increases. The AR mechanism therefore tolerates moderate heterogeneity in coding sparsity, but performs best when patterns share similar sparsity.

In summary, the AR mechanism accommodates patterns with varying sparsity, achieving optimal retrieval for moderately sparse codes ($s \approx 0.7$). Although the algorithm handles moderate heterogeneity in sparsity across stored patterns, retrieval performance benefits from relatively homogeneous sparsity of memory representations.

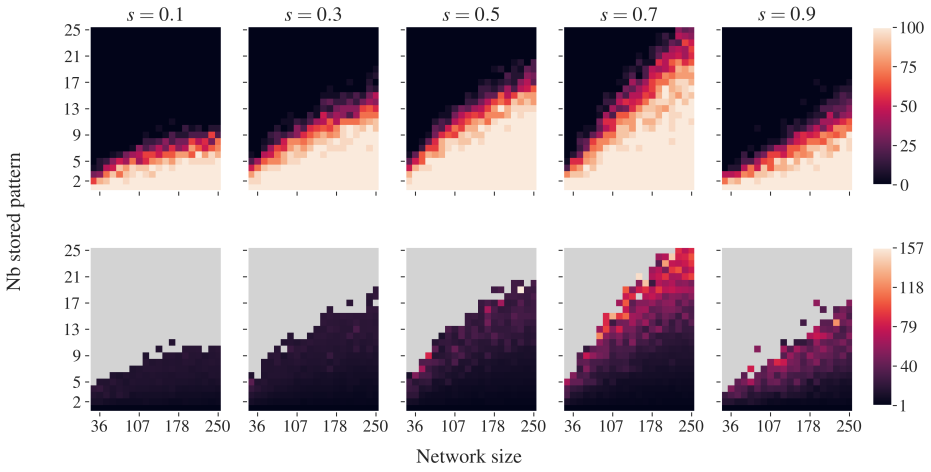

Figure 4.17 – Recovery capacity of AR for corpora with uniform sparsity $s \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. Patterns are generated independently with no imposed correlation other than that induced by sparsity. For networks of various sizes and numbers of stored patterns, we employ Alg. 6 with $\beta = 0.1$ until all patterns are recovered or until a spurious state is found. **(Top row)** Percentage of full retrieval, i.e., simulation runs where no spurious state occurs before recovering all stored patterns. Recovery is best for $s = 0.7$, with degraded performance for both lower and higher sparsity values. **(Bottom row)** Number of iterations required to recover all patterns. Data averaged over 30 independent simulations.

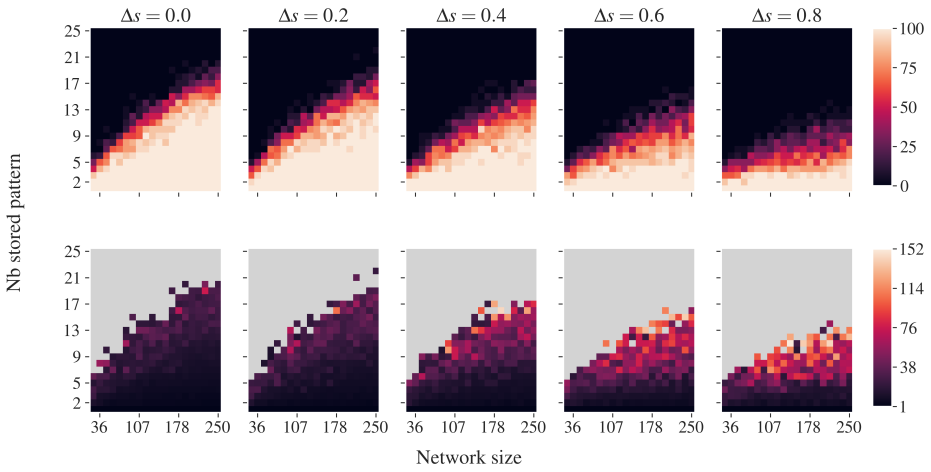

Figure 4.18 – Similar to Fig. 4.17 but for corpora with heterogeneous sparsities. For each pattern μ, sparsity is drawn from $\mathscr{U}[0.5 - \Delta s/2,\ 0.5 + \Delta s/2]$. The more heterogeneous is the sparsity, the lower the performances of AR.

4.8.4 . Impact of the leak parameter

In this section, we examine how the varying of the leak parameter affects the retrieval capacity of the AR mechanism.

In the CHN dynamics (Eq. 4.1), the term $-u_i/r$ is a leak term that pulls the membrane potential toward zero. The quantity $1/r$ determines the leak strength (larger $1/r$ yields a faster decay). In the rest of this thesis, the leak corresponds to $1/r = 1$.

Figs. 4.19 and 4.20 illustrate, for various leak strengths, the loads for which AR allows systematic recovery of all stored patterns. For weak leaks ($1/r < 1$), retrieval performance decreases compared to the reference case $1/r = 1$. For moderate-to-strong leaks ($1/r \geq 1$), the retrieval capacity remains comparable, but the number of iterations required to recover all patterns increases.

To understand the increased iteration count, we examine how the leak parameter affects synaptic weights. Figs. 4.21 and 4.22 show the distributions of storage weights W_{ij} and self-inhibition terms A_i, respectively. The self-inhibition values correspond to those accumulated at the end of AR, after all stored patterns have been successfully retrieved. Fig. 4.23 summarizes these observations by showing the average absolute weight $\langle |W_{ij}| \rangle$ and the average self-inhibition $\langle A_i \rangle$ as functions of $1/r$.

What appears from this study of the synaptic weights is that higher leaks induce larger-magnitude storage weights. As storage weights increase, the self-inhibition required to steer the network toward unvisited patterns also increases. Therefore, more iterations of the AR algorithm are necessary to explore the attractor landscape by sufficiently potentiating self-inhibition as shown in Figs. 4.19 and 4.20.

This raises the question : why does a stronger leak lead the GDA to produce larger-magnitude storage weights? The key point is that the GDA enforces a *target fixed-point potential* for each stored pattern. Ignoring self-inhibition, the CHN dynamics read

$$\dot{u}_i = -\frac{u_i}{r} + \sum_j W_{ij} v_j.$$

At a steady state associated with pattern μ, we have $\dot{u}_i = 0$ and $u_i = \tilde{u}_i^\mu$, hence

$$\tilde{u}_i^\mu \;=\; r \sum_j W_{ij} v_j^\mu \qquad \Longrightarrow \qquad \sum_j W_{ij} v_j^\mu \;=\; \frac{\tilde{u}_i^\mu}{r}.$$

Therefore, for fixed targets $\tilde{u}_i^\mu$, decreasing r (i.e., increasing the leak strength $1/r$) requires the synaptic drive $\sum_j W_{ij} v_j^\mu$ to grow proportionally like $1/r$. Since v_j^μ is bounded, achieving this larger drive forces the GDA to increase the magnitude of the weights W_{ij}, explaining the broader weight distributions observed at higher leak.

In summary, the AR mechanism is robust to variations in the leak parameter for $1/r \geq 1$, although a stronger leak requires more iterations due to the larger synaptic weights developed by the GDA.

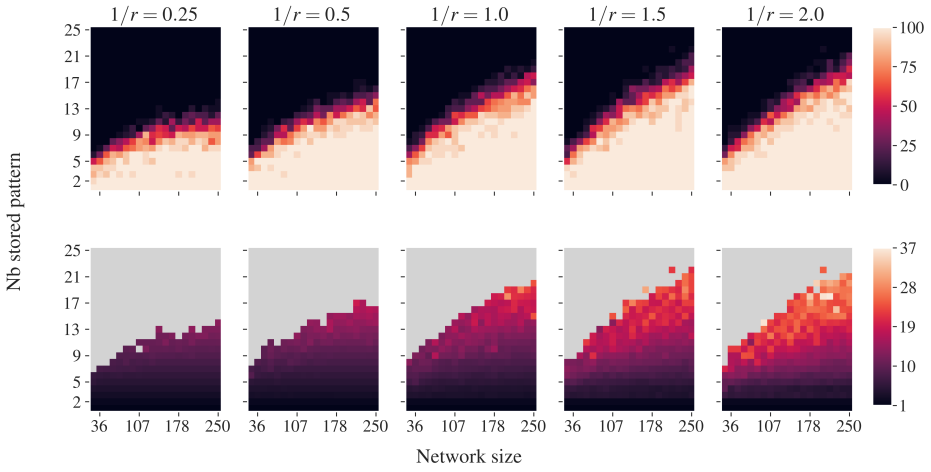

Figure 4.19 – Recovery capacity of AR for weak leak values ($1/r < 2$). Pattern sets are generated with Alg. 9 ($\rho = 0.5$) and AR is run with $\beta = 0.1$ until all patterns are recovered or until a spurious state is found. **(Top row)** Percentage of full retrieval (no spurious state before recovering all stored patterns). **(Bottom row)** Number of iterations required to recover all patterns. Data averaged over 30 independent simulations.

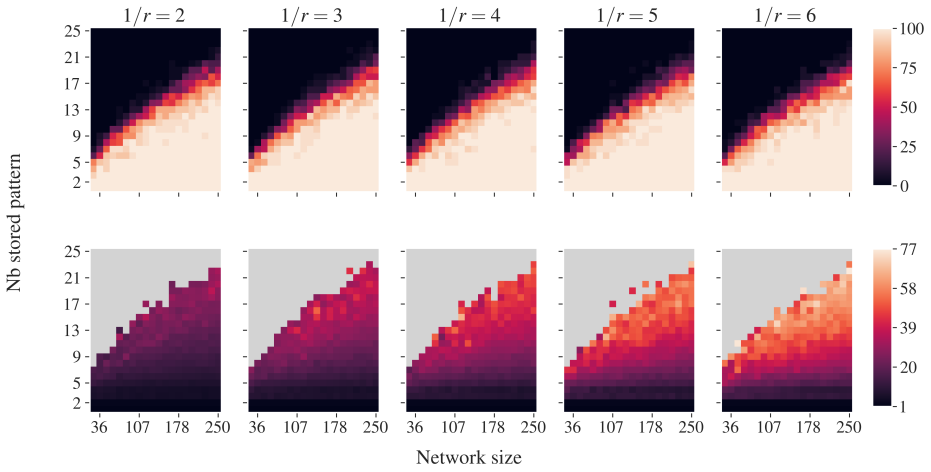

Figure 4.20 – Same as Fig. 4.19 but for stronger leak values ($1/r \geq 2$). Retrieval capacity remains comparable, while the number of iterations increases with leak strength.

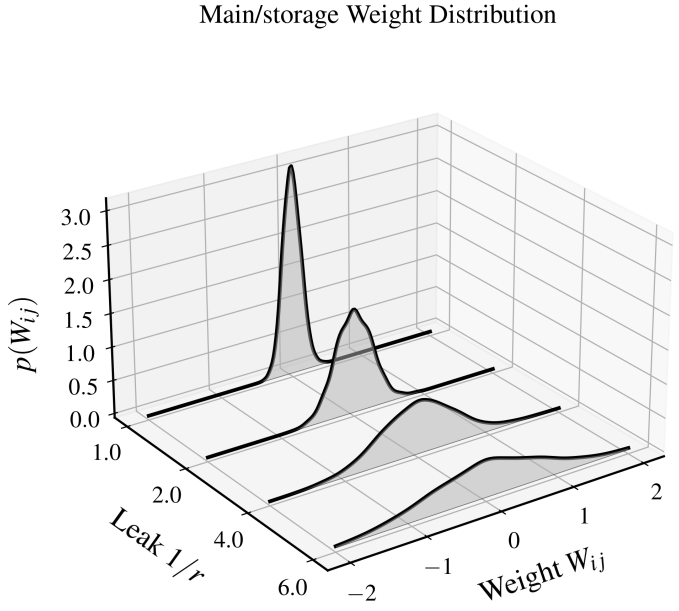

Figure 4.21 – Distribution of storage weights W_{ij} for various leak values. As the leak increases, the distribution broadens and shifts toward larger absolute values. Network size $N = 250$, number of stored patterns $M = 8$, correlation parameter $\rho = 0.5$, data averaged over 30 independent simulations.

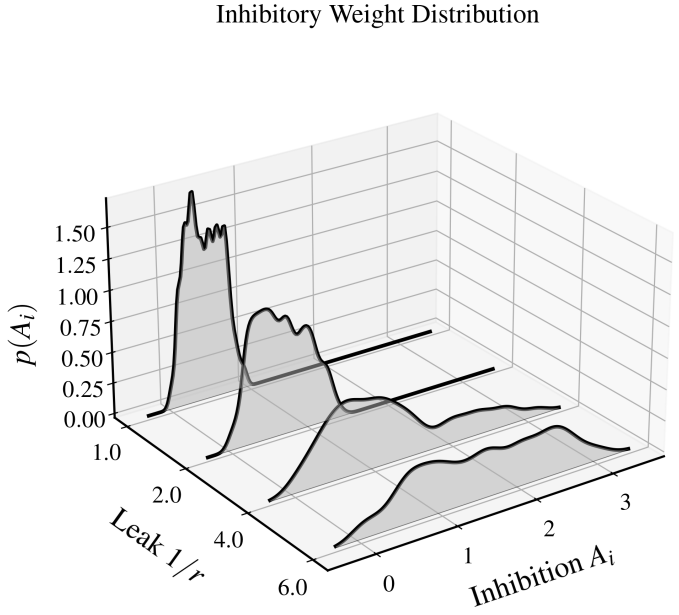

Figure 4.22 – Distribution of self-inhibition terms A_i at the end of AR for various leak values. Larger leak values require stronger self-inhibition to redirect the dynamics toward unvisited patterns. Same simulation parameters as Fig. 4.21.

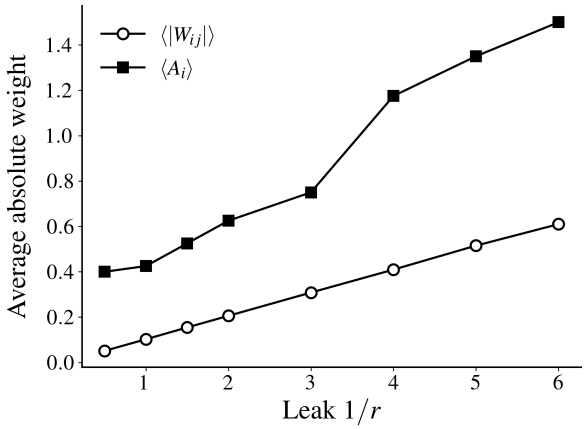

Figure 4.23 – Average absolute storage weight $\langle |W_{ij}| \rangle$ and average self-inhibition $\langle A_i \rangle$ as functions of the leak parameter $1/r$. Both quantities increase with leak strength. Same simulation parameters as Fig. 4.21.

4.8.5 . Reproduction and extension of McCallum's pseudorehearsal results

In Sec. 2.4.3, we presented the pseudorehearsal approach of McCallum (McCallum, 1998), which constitutes the most closely related prior work to our approach. Here, we summarize only the elements of McCallum's protocol that are necessary to understand our implementation choices and to reproduce the main qualitative findings; for a complete and exhaustive description of the original procedure (including all heuristic details and parameter choices), we refer the reader to McCallum's thesis. We then reproduce his central results and extend the analysis to partial-cue recovery, correlated patterns, and capacity scaling. Finally, we compare pseudorehearsal with our continuous incorporation (CI) method, which relies on AR for memory retrieval.

Results reproduction

McCallum stores patterns using the gradient descent algorithm, referred to as "delta learning" in the original work. He first stores a small base population of B patterns (in this section $B = 5$) in the network by applying GDA to this initial set. Then, for each new pattern to be incorporated, he performs pseudorehearsal as follows :

1. The network is probed with a large number of random initial states (2000 in the original experiments). Each probe state is relaxed to convergence under the network dynamics, until reaching a stable state.
2. The set of unique recovered states, capped at a maximum number P_{items}, forms the rehearsal population.
3. The new pattern to be incorporated is added to the rehearsal population, and the GDA is used to store this combined training set by continuing from the current weight matrix (i.e., without reinitializing $\mathbf{W}$ which potentially already contains previously stored patterns).

The full incremental protocol (bootstrap on B patterns followed by sequential incorporation) is summarized in Alg. 10 (Appendix. 6.7). The incorporation step performed at each iteration is detailed in Alg. 11 (Appendix. 6.7). Following McCallum's terminology, we refer to the recovered states as *items* in what follows.

During GDA training, two forms of noise are applied to the storage of the new pattern to broaden its attraction basin and improve learning; implementation details are given in McCallum's thesis McCallum (1998).

We first reproduce the central result of McCallum's pseudorehearsal experiments. The results are presented in Fig. 4.24, with our reproduction at the **top** and the original figure from McCallum's thesis at the **bottom**. Both figures show the number of *stable* patterns as a function of the total number

of sequentially incorporated patterns in a DHN of size $N = 100$ and for uncorrelated patterns ($\rho = 0$). A pattern is considered stable if, when the network is initialized with this pattern as its activity state, simulating the dynamics leaves the state unchanged (Sec. 2.3.4).

Our reproduction agrees with McCallum's original results : pseudorehearsal does partially protect previously stored patterns, and this protective effect strengthens as the number of rehearsed pseudo-items increases. In both Mccallum's and our results, the number of stable patterns follows the same qualitative evolution as the number of stored patterns M increases, and the plateau values at high loads are of the same order. The main discrepancy between our simulations and McCallum's results concerns the low-load regime. In McCallum's data, the pseudorehearsal conditions exhibit a pronounced bump where all incorporated patterns become stable at low loads. This bump is smaller in our reproduction. We tested several protocol variants (noise on/off, noise strength, early stopping criteria) but did not fully recover this feature.

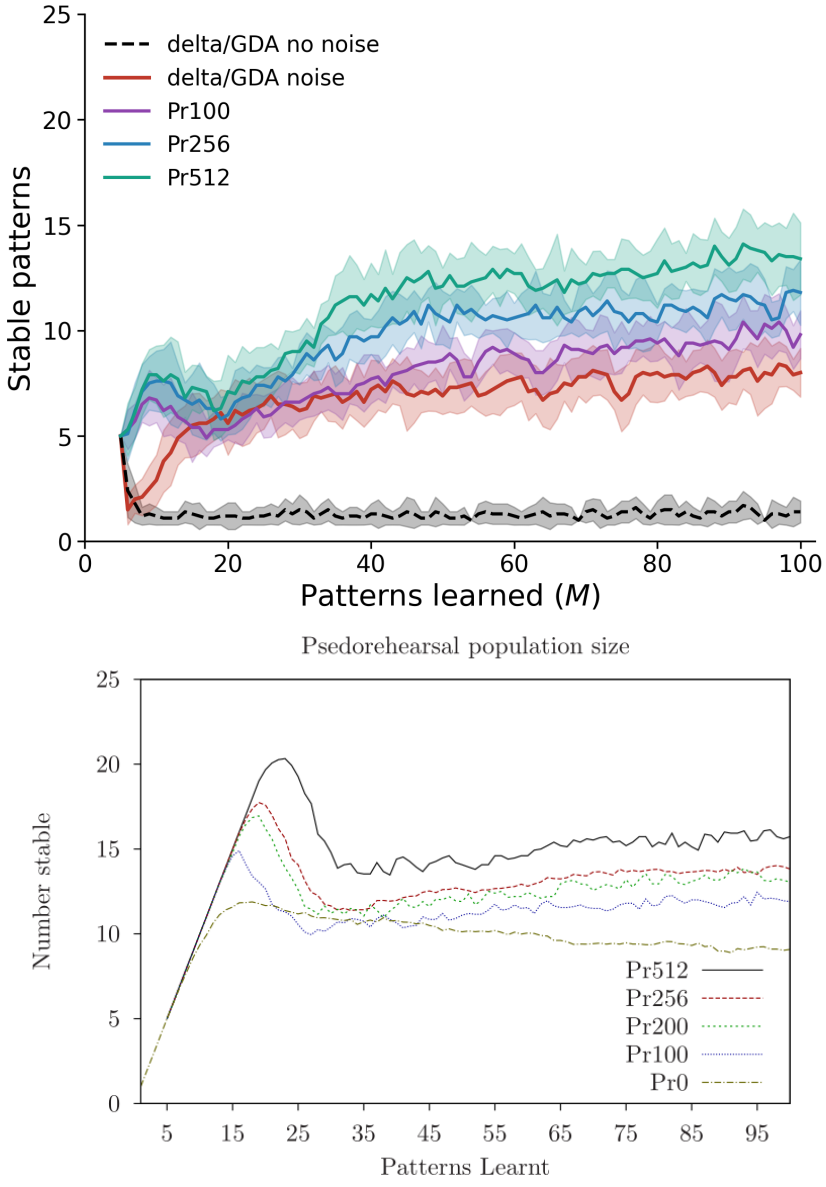

Figure 4.24 – **Pseudorehearsal in a DHN : our reproduction (Top) and McCallum's original results (Bottom, adapted from McCallum (1998)).** Number of stable patterns as a function of the total number of patterns learned M, for a network of size $N = 100$ with uncorrelated patterns ($\rho = 0$). A base population of 5 patterns is stored into the network before incremental learning. All Prk curves correspond to the incremental protocol in Alg. 10 (Appendix. 6.7). **Top :** Prk denotes pseudorehearsal with a pseudoitem cap of k. The "delta/GDA no noise" baseline (dashed black) applies the GDA to each new pattern alone without pseudoitems rehearsal or noise (Alg. 12 (Appendix. 6.7)). The "delta/GDA noise" baseline applies the GDA with heteroassociative and Gaussian noise without pseudoitems rehearsal (Alg. 12 (Appendix. 6.7)). **Bottom :** McCallum's original figure, with Pr0 corresponding to our "delta/GDA noise" condition.

Extension to partial-cue recovery

As emphasized above, stability does not ensure recoverability from degraded cues. To quantify retrieval robustness, we extend McCallum's analysis by measuring recovery from partial cues. For a given target pattern $\mathbf{x}^\mu$, we construct a partial cue by taking a fraction q of bits from $\mathbf{x}^\mu$ and randomizing the remaining $(1-q)$ fraction. We then run the DHN dynamics from this partial cue and declare successful recovery if the network converges back to the target pattern. Repeating this procedure over all stored patterns and multiple random cues yields the average number of recovered patterns at each load M.

Fig. 4.25 shows the number of recovered patterns as a function of M for different cue qualities q, under the Pr256 pseudorehearsal condition. As expected, recovery performance decreases as cues become more degraded. Importantly, even when stability remains relatively high, partial-cue recovery can drop substantially, confirming that stability is a permissive metric.

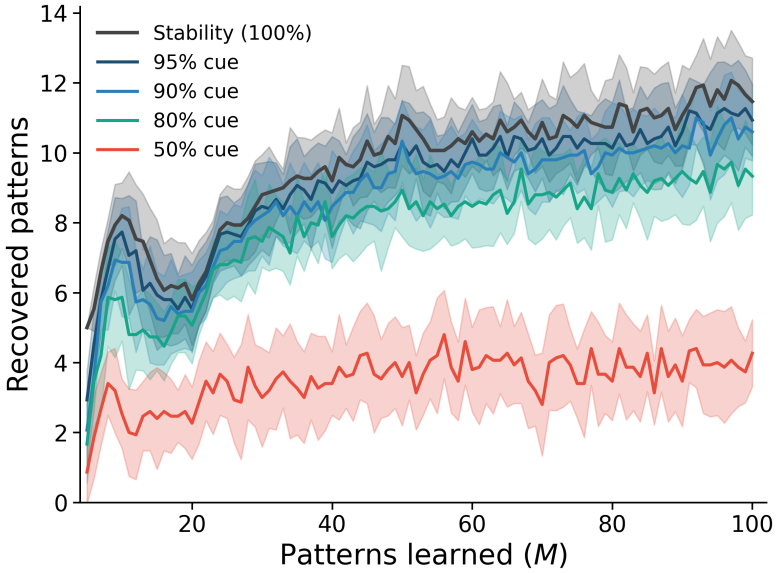

Figure 4.25 – **Pattern recovery from partial cues under pseudorehearsal (Pr256).** Number of recovered patterns as a function of the total number of patterns learned M, for a DHN of size $N = 100$ with uncorrelated patterns ($\rho = 0$). Each curve corresponds to a different fraction of informed bits q : the fraction indicates the proportion of bits taken from the target pattern, with the remaining bits randomized. Stability ($q = 1$) corresponds to the full-cue condition. Shaded regions indicate ± 1 standard deviation over 20 simulations. Incremental learning follows Alg. 10 (Appendix. 6.7); parameters as in Fig. 4.24 with $P_{\text{items}} = 256$. We can observe that, as the cues degrade, the number of recovered patterns decreases.

Extension to correlated patterns

McCallum's original experiments focus on uncorrelated patterns. To assess robustness when overlap increases, we also study correlated patterns. Patterns are generated using the parent-and-redraw procedure described in Alg. 9 (Appendix 6.3), controlled by a correlation parameter $\rho \in [0, 1]$.

Fig. 4.26 shows the number of stable patterns under Pr256 pseudorehearsal for different correlation levels. As ρ increases, the number of stable patterns decreases but does not drop to very low values. This is potentially due to the fact that the GDA/delta learning rule performs well for the storage of correlated patterns.

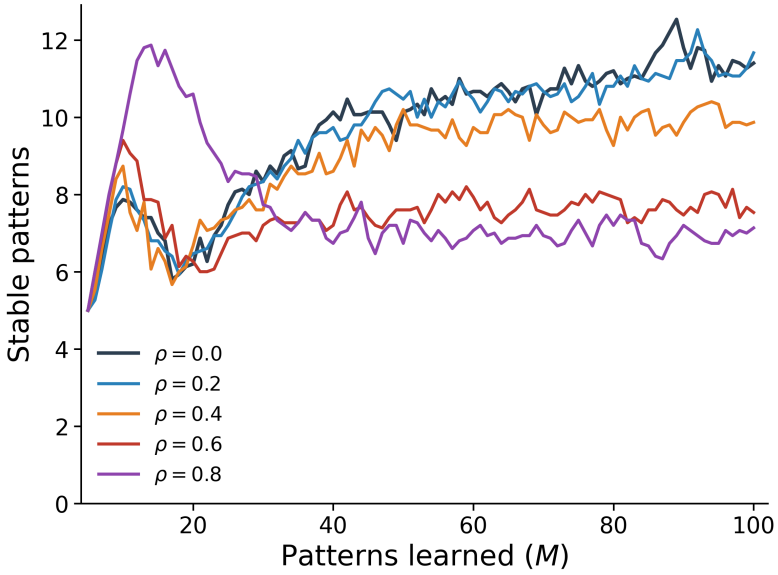

Figure 4.26 – **Effect of pattern correlation on pseudorehearsal performance (Pr256).** Number of stable patterns as a function of the total number of patterns learned M, for a DHN of size $N = 100$ and various correlation levels $\rho \in \{0.0, 0.2, 0.4, 0.6, 0.8\}$. Correlated patterns are generated using Alg. 9 (Appendix 6.3) and converted to bipolar coding via $\mathbf{x} \leftarrow 2\mathbf{x} - \mathbf{1}$. Incremental learning follows Alg. 10 (Appendix. 6.7); parameters as in Fig. 4.24 with $P_{\text{items}} = 256$.

Comparison with our method

To quantitatively compare our CI method with McCallum's, we introduce a scalar capacity metric : the *perfect retrieval capacity* M^*. M^* is measured from many CL scenarios and the procedure for a single scenario is as follows :

1. Patterns are incorporated one at a time into an initially empty network (no bootstrap population) via a chosen incorporation method. The measurement is independent of the incorporation method used; the method is entered only as a parameter of the evaluation.
2. After each incorporation of a new pattern, all currently stored patterns are queried using partial cues with a fraction of informed bits q.
3. The process continues until either (i) a query fails, (ii) the incorporation method itself signals a failure, or (iii) $M_{\max}$ patterns have been successfully stored and queried. The capacity for this scenario, denoted M_s^*, is the number of patterns successfully stored before the first failure.

The perfect retrieval capacity M^* is the maximum M such that at least a fraction θ of scenarios achieved $M_s^* \geq M$. In other words, M^* is the largest number of patterns that can be reliably stored across at least $\theta * S$ scenarios, with S the number of scenarios conducted. The complete procedure is detailed in Alg. 7.

We now describe how each incorporation method is instantiated within this framework :

- **McCallum's pseudorehearsal :**The incorporation step follows Alg. 11 (Appendix. 6.7). This method does not define any internal failure condition; the only way a simulation terminates is through a query failure after incorporation. Notice that the synaptic weight matrix **W** is never reinitialized during the process.
- **Continuous incorporation (CI) :** The incorporation step includes a sleep phase based on the AR mechanism (Alg. 6), adapted with two modifications. First, the free phase is omitted. Second, two internal stopping conditions are added. If a spurious pattern is encountered during retrieval, the simulation is terminated, as we consider that introducing false memories into the training set is not acceptable within our framework. Retrieval is considered successful when the number of distinct recovered patterns equals the number of currently stored patterns M, at which point the GDA is applied to the complete recovered set together with the new pattern. The adapted procedure is summarized in Alg. 14 (Appendix. 6.7). In our method, reinitialization of the synaptic weight matrix $\mathbf{W}$ before the GDA does not change the outcome.

We apply the M^* metric to both McCallum's pseudorehearsal (Pr512) and our CI method, sweeping over network sizes $N \in \{50, 100, 150, 200, 250\}$ and correlation levels $\rho \in \{0.0, 0.2, 0.4, 0.6, 0.8\}$. For McCallum's method, we use

$P_{\text{items}} = 512$, the largest pseudoitem count considered in the original thesis. Fig. 4.27 shows M^* for both methods, evaluated at four cue levels : stability ($q = 1$), $q = 0.95$, $q = 0.80$, and $q = 0.50$.

For McCallum's pseudorehearsal, M^* decreases as cue quality degrades. Furthermore, M^* does not increase with network size in the tested range. One potential avenue to improve scaling would be to increase the number of pseudoitems beyond 512, which could provide a more representative sample of the attractor landscape in larger networks.

By contrast, CI does not show such degradation as cue quality decreases, and capacity scales with network size. Regarding correlation, we observe the same non-monotonic behaviour reported earlier, whereby recovery is optimal for intermediate levels of correlation.

It should be noted that, as discussed in Sec. 4.8.5, our reproduction does not fully recover the very good performance at low loads observed in McCallum's original results. Since the stopping conditions of Alg. 7 cause simulations to terminate upon the first failure, the measured M^* values typically fall in this low-load regime. A reproduction that successfully recovers this low-load performance could therefore yield substantially higher M^* values for McCallum's method, potentially altering the comparison in its favor.

We now summarize the key differences and similarities between McCallum's pseudorehearsal and CI. Both methods share a common structure : they retrieve activity patterns from the network's own dynamics and retrain on them together with new patterns. Beyond the differences in stopping criteria and the role of spurious states discussed above, the two methods differ in several other respects.

- **Network type.** McCallum uses a DHN. Our method uses a CHN.
- **Retrieval mechanism.** McCallum probes the network with a large number of random initial states (2000 per incorporation step). Our method exploits self-inhibition to sequentially visit stored attractors from a neutral initial condition (Alg. 6). This typically necessitates less iterations, although this is not precisely quantified here.
- **Nature of recovered states.** In McCallum's method, the recovered population inevitably contains both true items and pseudoitems. The pseudoitems correspond to what we call spurious patterns. In CHNs, at low memory loads, convergence to a spurious state is typically rare.
- **Noise during storage.** McCallum applies heteroassociative noise (ν_h) and Gaussian input noise (σ_{G}) to the new pattern during GDA training. Our method applies no noise during storage or retrieval.
- **CF mitigation.** McCallum's pseudorehearsal partially mitigates CF, as the recovered population is an inexact and incomplete representation of the stored memory set. Our method, under favorable conditions, recovers the complete set of stored patterns with high probability. The

GDA then operates on the same dataset as in batch learning, which fully prevents CF. The limitation of our methods comes from the fact that we recover the stored patterns without outputting spurious patterns only for small loads.

In summary, the two methods embody different approaches to rehearsal. McCallum trains on a large population of recovered states of variable quality : the mixture of true items and pseudoitems collectively reconstructs part of the attractor landscape and partially mitigates CF. Our method trains on a small set of high-quality recovered states : the stored patterns themselves. When all stored patterns are successfully recovered, CF is fully prevented.

Because our goal is to avoid incorporating false memories into the training set, any spurious state during retrieval invalidates the entire sleep phase. McCallum's method, by contrast, tolerates spurious states in the recovered population.

Training on recovered states that include spurious patterns within our framework was not explored in this work and could constitute a direction for future investigation.

It is important to note that the stopping conditions used in CI imply two assumptions that constitute limitations of our approach. The network must have access to the number of stored patterns M to determine when retrieval is complete, and it must be able to distinguish stored patterns from spurious states. Note that the latter does not necessarily require exact knowledge of the stored patterns : as discussed by McCallum (1998), statistical properties of the retrieved population could in principle be used to identify spurious states. This direction is not explored in the present work. By contrast, McCallum's method does not require either assumption, as probing runs for a fixed number of random initializations and trains on all recovered states indiscriminately.

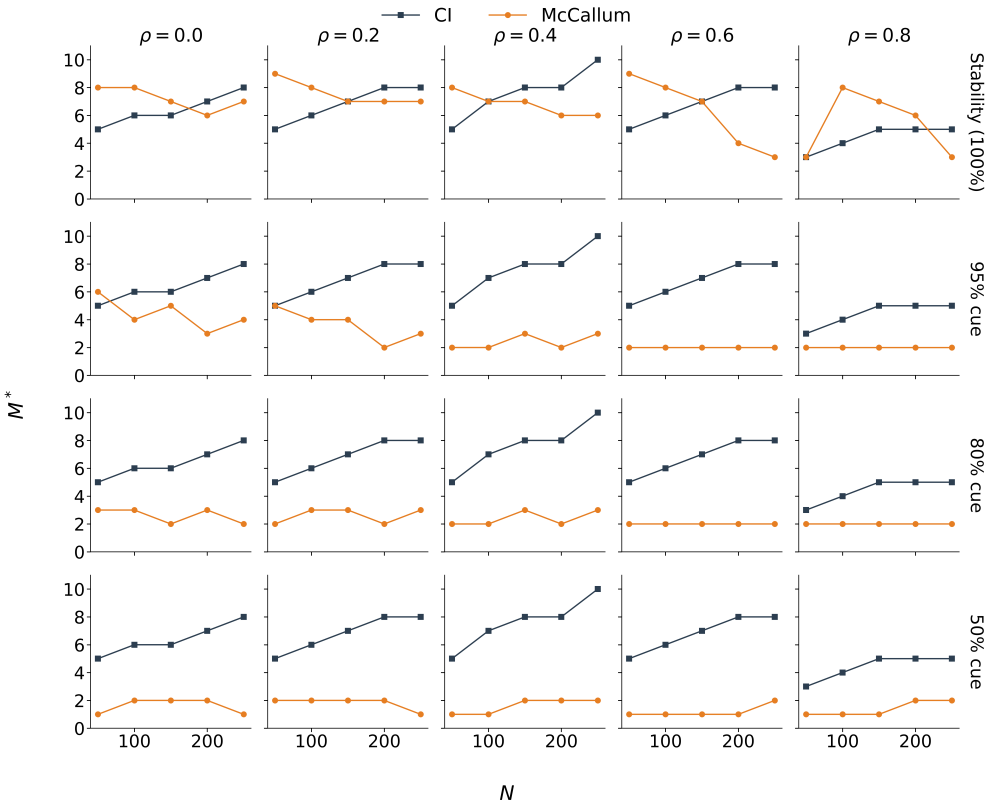

Figure 4.27 – **Comparison of perfect retrieval capacity between CI and McCallum's pseudorehearsal (Pr512) across network sizes and correlation levels.** M^* as a function of network size $N \in \{50, 100, 150, 200, 250\}$ for various fractions of informed bits (rows) and correlation levels ρ (columns), with $\theta = 0.9$. Correlated patterns are generated using Alg. 9 (Appendix 6.3) and converted to bipolar coding via $\mathbf{x} \leftarrow 2\mathbf{x} - \mathbf{1}$. M^* is estimated from $S = 20$ independent scenarios per (N, ρ) configuration. For McCallum's method, parameters as in Fig. 4.24 with $P_{\text{items}} = 512$.

Algorithm 7 Evaluation of perfect retrieval capacity

Input : Net size N, maximum load $M_{\max}$, number of scenarios S
Correlation parameter ρ, success threshold θ, fraction of informed bits q
Incorporation method $\mathscr{I} \in \{$pseudorehearsal, CI (our), Hebbian, Storkey, Iterative GDA$\}$
Initialize : Capacities $\{M_s^*\}_{s=1}^{S}$
for $s = 1$ to S **do**
    Generate pattern set $\{\mathbf{x}^\mu\}_{\mu=1}^{M_{\max}}$ with correlation parameter ρ
    Initialize empty network $\mathscr{N}$
    $M_s^* \leftarrow 0$
    for $M = 1$ to $M_{\max}$ **do**
        Incorporate pattern $\mathbf{x}^M$ in $\mathscr{N}$ using the specified method $\mathscr{I}$
        if $\mathscr{I} =$ CI **and** sleep-phase retrieval (Alg. 14) returns failure **then**
            break
        end if
        Query all patterns $\mathbf{x}^1, \ldots, \mathbf{x}^M$ with informed bits fraction q
        if any query fails **then**
            break (end of scenario s)
        end if
        $M_s^* \leftarrow M$
    end for
end for
Return : $M^* = \max\{M : |\{s : M_s^* \geq M\}|/S \geq \theta\}$

4.8.6 . Larger networks and comparison with other methods

Having compared our CI method with McCallum's pseudorehearsal in the previous section, we now broaden the comparison to other memory storage techniques applied to larger networks. McCallum's method is not included here : as shown in Sec. 4.8.5, its perfect retrieval capacity does not increase with network size, and the large number of random probes required at each incorporation step makes it computationally prohibitive for larger networks.

We compare CI applied to CHNs with two incremental DHN methods, Hebbian plasticity (Eq. 2.3) and the Storkey learning rule (Sec. 3.1.2), and with an iterative GDA baseline applied to CHNs (Alg. 12). The first two methods are chosen because, like CI, they support incremental learning. For Hebbian plasticity, this follows from the fact that the weight matrix is a sum of independent contributions $\Delta W_{ij} = \frac{1}{N} x_i^\mu x_j^\mu$ from each pattern μ. A new pattern can therefore be added by simply updating the weights with its contribution. The Storkey rule similarly allows for incremental updates (see Chap. 3).

Fig. 4.28 compares the perfect retrieval capacity M^* across methods for various network sizes and correlation levels, using the evaluation procedure introduced in Sec. 4.8.5 (Alg. 7). All methods are evaluated using partial cues with $q = 0.50$ informed bits.

As shown in Fig. 4.28, at low correlation ($\rho = 0.0$ and $\rho = 0.2$), Hebbian and Storkey methods substantially outperform CI. The Storkey rule achieves the highest capacity, saturating at the maximum number of patterns tested (indicated by triangle markers). As the correlation increases, this hierarchy changes. At $\rho = 0.4$, our method becomes competitive with the Storkey rule, while Hebbian plasticity shows marked degradation. At $\rho = 0.5$ and $\rho = 0.6$, CI outperforms both Hebbian and Storkey methods, which exhibit sharply reduced capacity in the presence of strong pattern correlations. Iterative GDA exhibits near-zero capacity across all correlation levels.

To verify if the choice of θ does not strongly perturb the observed dynamics, we examine how varying θ affects the perfect retrieval capacity of CI. Fig. 4.29 confirms that the qualitative behavior is robust to the choice of θ.

In summary, CI does not achieve the highest storage capacity for all regimes. For uncorrelated or weakly correlated patterns, classical DHN methods, particularly the Storkey rule, offer superior performance. However, our method shows improved performance in the regime of strongly correlated patterns.

It is important to note that more advanced methods for storing correlated patterns exist and, if made compatible with CL scenarios (which is realistic), would potentially outperform the approaches considered here. These include reformulated modern Hopfield networks, which can be understood as classical Hopfield networks with a hidden layer (Krotov & Hopfield, 2021), reservoir computing architectures (Kong et al., 2024; Maass et al., 2002; Sussillo & Ab-

bott, 2009), and threshold-linear networks with sophisticated normalization pipelines (Schönsberg et al., 2021), which are inherently compatible with CL. These methods fall outside our scope, as we deliberately restrict the discussion to networks without hidden layers or elaborate normalization schemes (see Chap. 3).

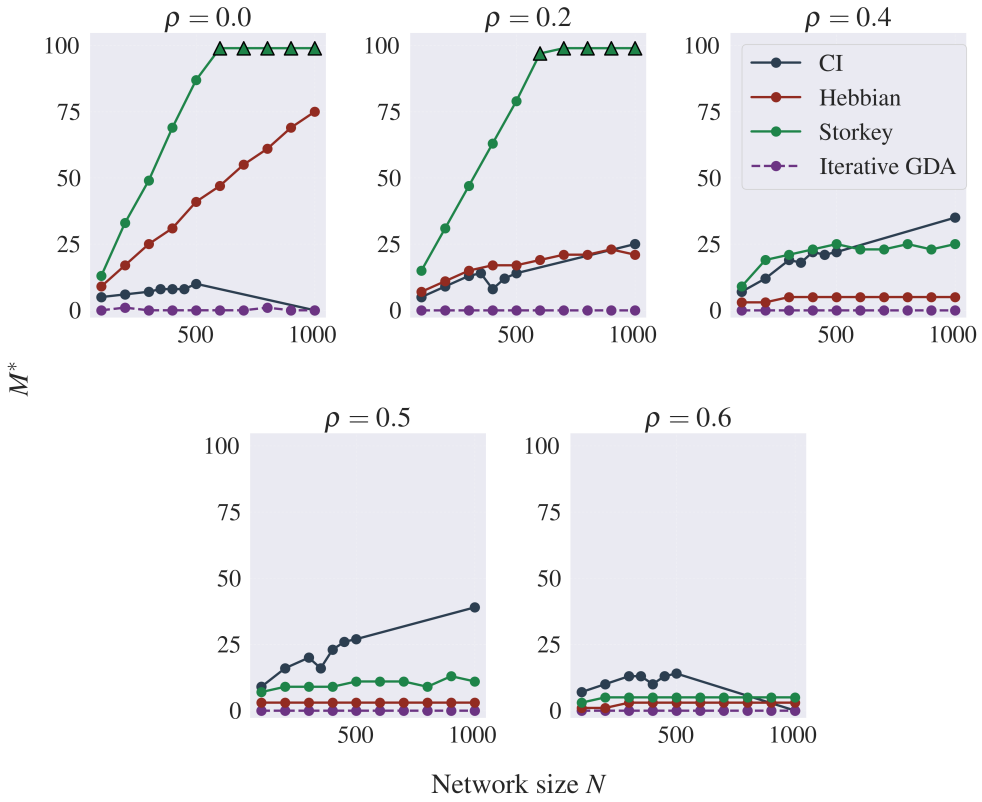

Figure 4.28 – **Comparison of perfect retrieval capacity across methods and correlation levels.** Perfect retrieval capacity M^* as a function of network size N for CI (CHN), Hebbian plasticity (DHN), Storkey learning rule (DHN), and iterative GDA (CHN). Each panel corresponds to a different correlation level ρ. Triangle markers indicate saturation at the maximum number of patterns tested ($M_{\max}$). For all methods, retrieval capacity is evaluated using partial cues with $q = 50\%$ informed bits. M^* is estimated from $S = 20$ independent scenarios per (N, ρ) configuration with $\theta = 0.9$. Correlated patterns are generated using Alg. 9 (Appendix 6.3) and converted to bipolar coding via $\mathbf{x} \leftarrow 2\mathbf{x} - \mathbf{1}$ for DHNs.

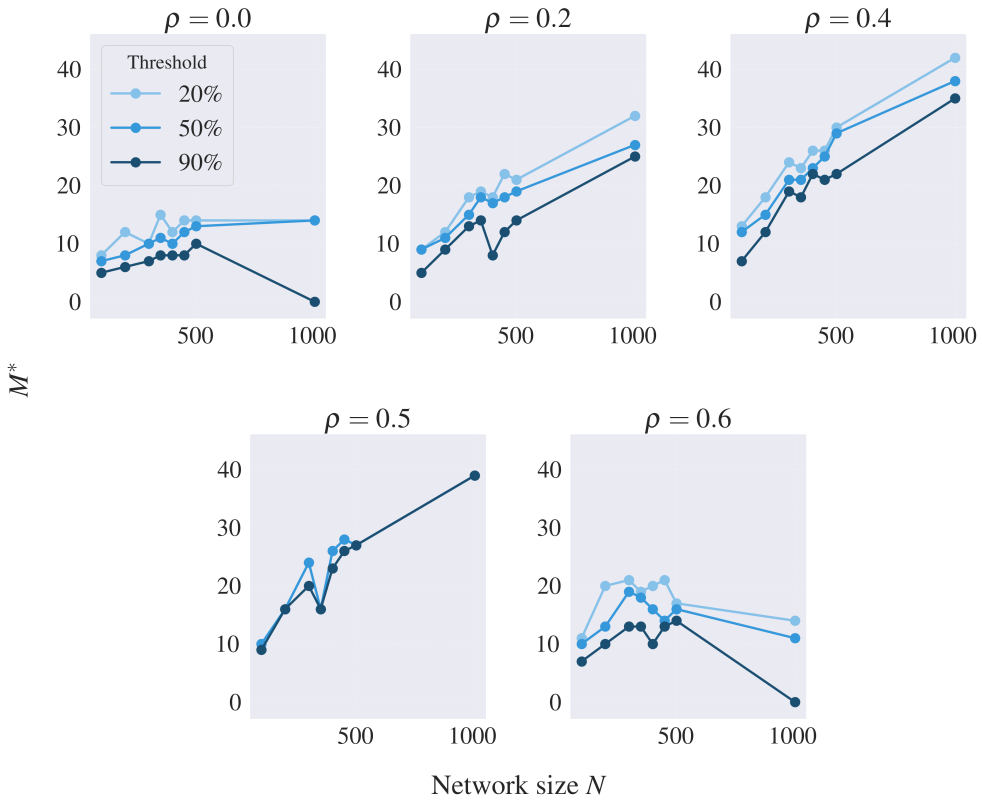

Figure 4.29 – **Influence of the success threshold θ on perfect retrieval capacity.** Perfect retrieval capacity M^* as a function of network size N for $\theta \in \{0.2, 0.5, 0.9\}$. Each panel corresponds to a different correlation level ρ. As expected, relaxing θ increases the measured capacity. The relative ordering across correlation levels is preserved regardless of the chosen θ. Retrieval is evaluated using partial cues with $q = 50\%$ informed bits. M^* is estimated from $S = 20$ independent scenarios per (N, ρ) configuration. Correlated patterns are generated using Alg. 9 (Appendix 6.3).

4.8.7 . Exploring memory states at various distances from the neutral state

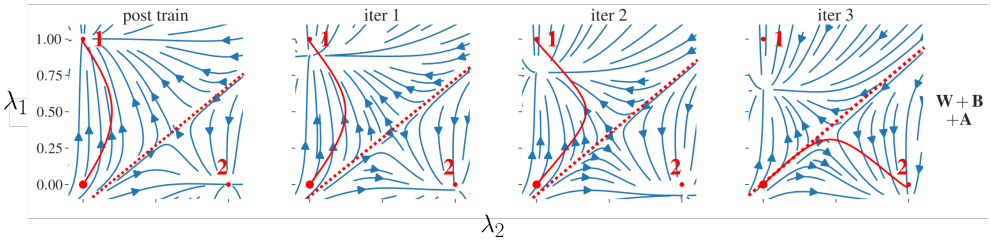

Figure 4.30 – **Vector fields showing the successive exploration of memory patterns stored at different distances from the neutral state.** Similar to Fig. 4.2 the 2D subspace spanned by the two pattern states is shown. In red the convergence trajectory of the network state. Pattern 1 (upper left) is stored at distance $d^1_{\text{mul}} = 1.25$, pattern 2 (lower right) at distance $d^2_{\text{mul}} = 1$. Post-training : network converges to pattern 1. Iterations 1-3 : successive potentiation of self-inhibition $\mathbf{A}$ (with $\beta = 0.1$) progressively shifts the separatrix (dashed red line), eventually enabling convergence to pattern 2 by iteration 3. Here the patterns are generated using Alg. 9 with $\rho = 0.5$.

The results presented in this work have shown that through the self-inhibition mechanism, the network is able to explore its attractor landscape to recover stored memories. Previously, memory states were constructed to lie at the same distance from the neutral state by choosing a binary pattern $\mathbf{x}^\mu$ and computing :

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

resulting in all patterns lying at distance $u_{\text{target}}\sqrt{N}$ from the neutral state $\mathbf{0}$, where N is the network size.

To assess the robustness of the rehearsal mechanism, we tested the case where memory patterns are stored at different distances from the neutral state. This was achieved by introducing a distance multiplier d^μ_{mul} for each pattern μ, modifying the memory state construction as :

$$\tilde{u}_i^\mu = \begin{cases} +u_{\text{target}} \cdot d^\mu_{\text{mul}} & \text{if } x_i^\mu = 1 \\ -u_{\text{target}} \cdot d^\mu_{\text{mul}} & \text{if } x_i^\mu = 0 \end{cases}$$

This places pattern μ at distance $u_{\text{target}} \cdot d^\mu_{\text{mul}}\sqrt{N}$ from the neutral state.

To enable the storage of memory patterns at arbitrary distances, we introduced plastic biases $\mathbf{b} = (b_1, b_2, \ldots, b_N)$ for each unit. The network dynamics become :

$$c\frac{du_i}{dt} = \sum_j W_{ij} v_j + b_i - \frac{u_i}{r} \tag{4.9}$$

These biases are learned alongside the synaptic weights using the gradient descent algorithm (GDA). During training, both the weight matrix $\mathbf{W}$ and the bias vector $\mathbf{b}$ are updated to minimize the error between the network's steady state and the target memory state. Notably, removing the symmetry constraint on the weights (i.e., allowing $W_{ij} \neq W_{ji}$) becomes essential; the learning process fails entirely when weights are constrained to be symmetric.

Fig. 4.30 illustrates the dynamical landscape induced by GDA when storing patterns at different distances from the neutral state. Pattern 1 (λ_1) is stored at distance $d^1_{\text{mul}} = 1.25$ and pattern 2 (λ_2) at distance $d^2_{\text{mul}} = 1$. An emergent property of the GDA learning scheme is that patterns stored at greater distances develop larger attraction basins, biasing convergence from the neutral state toward these more distant patterns. By updating the adaptation terms $\mathbf{A}$ using a small inhibitory potentiation parameter $\beta = 0.1$ after each convergence, the network progressively shifts the separatrix between attraction basins, eventually enabling access to pattern 2.

This demonstrates that even with asymmetric distances between patterns and the neutral state, our mechanism can compensate for the resulting uneven dynamical landscape and successfully explore both stored memories. In this case, our approach exhibits robustness to asymmetric dynamical landscapes.

These results suggest that our approach may be applicable to networks with inherently uneven dynamical landscapes, as is likely the case for ANNs addressing credit assignment problems in classification or generation tasks. In such models, if a dynamical landscape can be defined, there is no *a priori* reason for memories to lie at uniform distances from a given neutral state.

4.8.8 . Conclusion

This section gives additional characterization of our model.

First, we show that, from the neutral state, the gradient descent algorithm produces a balanced attraction toward all stored patterns ; self-inhibition then selectively suppresses retrieved patterns, enabling sequential exploration. The push metric reliably predicts the retrieval order at low loads, but this predictive power diminishes as the load increases.

Then we show that sparsity biases the exploration order toward less sparse patterns and modulates overall capacity, which peaks at intermediate sparsity values.

We also show that the leak parameter affects the retrieval performance. The retrieval mechanism fails for small leaks but remains robust even for very strong leak values. The leak also affects the magnitude of synaptic weights, with a stronger leak inducing larger storage synaptic weights (the ones trained by the GDA) and therefore requiring correspondingly stronger self-inhibition to allow pattern retrieval.

A reproduction and extension of McCallum's pseudorehearsal experiments shows that his method partially mitigates catastrophic forgetting but does not scale with network size over the range of pseudoitem limits explored in this work. It is possible that larger pseudoitem populations would improve scaling, but we limited our analysis to the maximum value considered in McCallum's original study. When evaluated on stored pattern stability alone, McCallum's method and our CI method achieve comparable capacity. However, when retrieval is evaluated from partial cues, CI achieves higher perfect retrieval capacity across all cue levels and network sizes tested. However, this comparison should be interpreted with caution : our reproduction did not fully recover the favorable low-load behavior reported by McCallum, and since Alg. 7 tends to yield M^* values precisely in this regime, the outcome could be more favorable to pseudorehearsal under a more faithful reproduction. In any case, the purpose of this comparison is not to establish a ranking between the two methods but to provide a qualitative characterization of their respective behaviors and trade-offs.

A broader comparison with the Hebbian and Storkey learning rules shows that CI does not achieve the highest capacity for uncorrelated patterns but offers advantages for strongly correlated memories.

And finally, we show that our AR mechanism has the potential to tolerate patterns stored at heterogeneous distances from the neutral state.

5 - Conclusion

In this work, we develop a method that allows continuous learning (CL) in associative memory networks (AMNs). To achieve this, we introduced a mechanism that allows the network to autonomously explore its memory space without being trapped in dominant attractor states. The mechanism enabling this exploration is based on inhibitory adaptation inspired by biological observations. We show that the memory replays (MR) resulting from the exploration dynamic can be used to orchestrate rehearsal and, therefore, mitigate catastrophic forgetting (CF). By mitigating CF, our approach enables the incremental addition of new memories to the network, thus demonstrating its capacity for CL.

It has been previously stated that an important criterion for CL is the autonomy of the learning system/agent (Sec. 2.4.2). In this regard, we can highlight an important limitation of our learning scheme. While the replay dynamic is fully autonomous, it is less obvious to interpret the storage mechanism as autonomous. In fact, we rely on the gradient descent algorithm (GDA) to store the memories that have been extracted during spontaneous activity. The GDA relies on a somewhat implicit memory buffer $\{\mathbf{x}\}$ that can still be interpreted as an external memory for which no implementation is explicitly defined. We can also consider this through the lens of locality of the dynamics as defined in Sec. 2.2.3. Querying the memory buffer can be considered a non-local dynamic since we have not identified any physical substrate to do so. Here we can see that the notions of autonomy and locality are deeply intertwined.

An interesting development of this work would therefore be to implement a model for which both the MR dynamic and the storage dynamic of correlated memory states are fully autonomous and local. This model would be an improvement in terms of bioplausibility and therefore give a better account of what the brain may do during spontaneous activity. This model would also offer more potential in terms of neuromorphic implementations, especially for technologies that rely only on local dynamics.

Another possible development is to adapt the memory replay mechanism proposed in this work to other architectures such as generative models. Similarly to associative memory networks (AMNs), generative models (LLMs, diffusion models, autoencoders) show attractor-like dynamics whereby the activity of the network is characterized by a trajectory converging toward stored memories (Im et al., 2016; Biroli et al., 2024; Wang et al., 2025). We therefore hypothesize that iterative inhibition of spontaneously generated activity could enable generative models to explore and output a broad range of their stored information. In these models, the ability to extract stored information could also help mitigate CF, among other things.

This work points toward a potential shift in the way artificial neural networks are trained. The current paradigm relies primarily on conditioning the behavior of the network through external data. Our approach demonstrates that networks can also leverage their own structured spontaneous activity for learning. Because this activity exposes previously stored information, it enables learning processes, such as CL, that typically rely not only on newly acquired knowledge.

6 - Appendices

6.1 . Derivation of the gradient descent learning rule

At steady state, when $\frac{du_i}{dt} = 0$, from Eq. 4.1 we have :

$$u_i = r\sum_j W_{ij} v_j$$

where $v_j = \sigma(u_j)$. We can then define

$$\hat{u}_i = r\sum_j W_{ij}\sigma(\tilde{u}_j)$$

the expected potential of the unit i when the rest of the units are at their target potential $\tilde{u}_j$.

We can then define the error function E as the sum of squared differences between the expected potentials $\hat{u}_i$ and target potentials $\tilde{u}_i$:

$$E = \frac{1}{2}\sum_i (\tilde{u}_i - \hat{u}_i)^2$$

Taking the derivative with respect to the expected potential :

$$\frac{\partial E}{\partial \hat{u}_i} = (\tilde{u}_i - \hat{u}_i)$$

To compute how the error changes with respect to the weights W_{ij}, we use the chain rule :

$$\frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial \hat{u}_i}\frac{\partial \hat{u}_i}{\partial W_{ij}}$$

The partial derivative with respect to W_{ij} is :

$$\frac{\partial \hat{u}_i}{\partial W_{ij}} = r v_j$$

This is an approximation that captures the immediate direct effect of W_{ij} on $\hat{u}_i$, assuming that v_j remains constant and ignoring feedback effects in the recurrent network. This approximation is valid when the optimization steps are small relative to the nonlinearities in the sigmoid function.

Therefore, the error gradient with respect to W_{ij} becomes :

$$\frac{\partial E}{\partial W_{ij}} = (\tilde{u}_i - \hat{u}_i) r v_j$$

Finally, we update the weights using gradient descent :

$$\Delta W_{ij} = \alpha \frac{\partial E}{\partial W_{ij}} = \alpha r (\tilde{u}_i - \hat{u}_i) v_j$$

where α is the learning rate, typically set to 0.0001 to ensure convergence.

6.2 . Querying procedure

The network's ability to retrieve stored patterns can be tested through partial cues. Given a subset of informed units from a pattern μ, we initialize their potentials according to the partial pattern while leaving other units at rest :

$$u_i^\mu(t=0) = \begin{cases} +6 & \text{if unit } i \text{ is informed and } x_i^\mu = 1 \\ -6 & \text{if unit } i \text{ is informed and } x_i^\mu = 0 \\ 0 & \text{otherwise} \end{cases}$$

The network then evolves according to Eq. 4.1 until it reaches a stable state ($|\frac{du}{dt}|_\infty < \varepsilon$). The thresholding procedure then allows for the reading of a stored binary pattern $\mathbf{x}^\mu$:

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

with $v_i(t_f)$ the rate of neuron i after convergence of the dynamic. Alg. 8 summarizes the procedure.

When operating below capacity, the dynamics typically converges to the stored pattern most similar to the initial cue. The final state is interpreted as a binary pattern using the thresholding procedure (Eq. 4.2). As with traditional DHNs, retrieval success depends both on the network load and on the number of units informed in the initial cue. Although a theoretical analysis of the storage capacity under Alg. 4 would be of interest, our focus here is on the autonomous retrieval mechanism, which operates in a regime well below this limit for which pattern retrieval is highly reliable.

Algorithm 8 Querying with convergence of the network using Euler method

1: Given a pattern μ and a subset of informed units
2: **set** $u_i^\mu(t=0) = \begin{cases} +u_{\text{target}} & \text{if } x_i^\mu = 1 \\ -u_{\text{target}} & \text{if } x_i^\mu = 0 \\ 0 & \text{if } x_i^\mu \text{ not informed} \end{cases}$
3: **repeat**
4: **for** each unit i **do**
5: $u_i(t+=1) = u_i(t) + \delta(\sum_j W_{ij} v_j - \frac{u_i(t)}{r})$
6: **end for**
7: **until** $\|\dot{\mathbf{u}}\|_\infty < \varepsilon$
8: **define** $t_f = t$ ▷ Final time at convergence
9: **read** : $x_i^\mu = \begin{cases} 1 & \text{if } v_i(t_f) > 0.5 \\ 0 & \text{otherwise.} \end{cases}$

6.3 . Construction of correlated patterns

Algorithm 9 Generation of p correlated random patterns

Generate a random binary pattern $\mathbf{x}^{parent} \in \{0,1\}^N$
with $p(x_i^{parent} = 0) = 0.5 = 1 - p(x_i^{parent} = 1)$
Set $k = \lfloor (1-\rho)N \rfloor$, where ρ is the ratio of bits not randomized
for $i = 1$ to p **do**
    $\mathbf{x}^i \leftarrow \mathbf{x}^{\text{parent}}$
    Randomly select k distinct indices $\{j_1, \ldots, j_k\}$ from $\{1, \ldots, N\}$
    for $m = 1$ to k **do**
        $x^i_{j_m} \leftarrow$ random choice from $\{0,1\}$ with equal probability
    end for
end for

6.4 . Model with inhibitory matrix

To allow the sequential exploration of stored patterns, a second approach has been tested. A set of plastic inhibitory synapses W'_{ij} has been added to the network dynamics :

$$c\frac{du_i}{dt} = \sum_j W_{ij}v_j - \sum_j W'_{ij}v_j - \frac{u_i}{r} \tag{6.1}$$

$$W'_{ij} \leftarrow W'_{ij} + \beta v_i(t_f)v_j(t_f) \tag{6.2}$$

with $v_i(t_f)$ the rate of neuron i after convergence of the dynamics. Self-inhibition $W'_{i,i}$ is set to 0 and undergoes no plasticity. A model combining self-inhibition and plastic inhibitory synapses has been tested and yields the same qualitative results. β represents the potentiation strength of the inhibitory synapses and is selected as a small value, typically below 0.05. The plasticity rule in Eq.6.2 is Hebbian.

We can observe that, similarly to adaptation, plastic synapses allow the retrieval of correlated memories Fig. 6.1 and Fig. 6.2. The potentiation strength β has to be weaker than that for adaptation, as a neuron is contacted by many inhibitory synapses. The retrieval dynamics are similar to the ones observed with adaptation in every aspect.

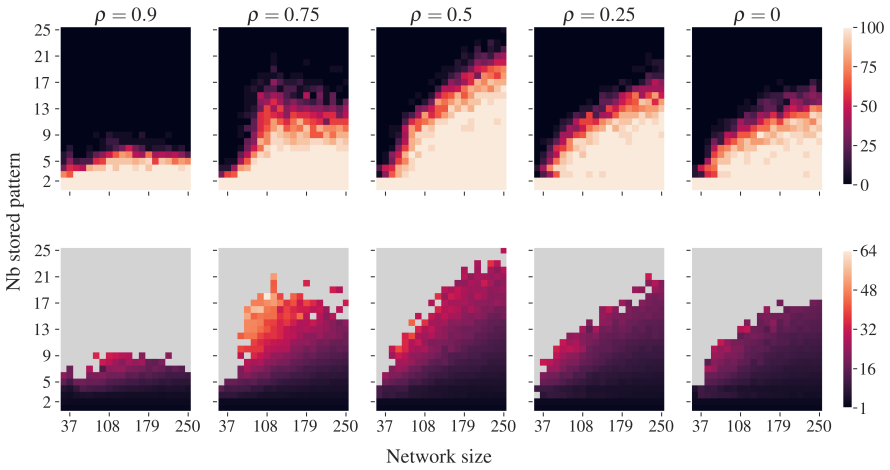

Figure 6.1 – Similar to Fig. 4.5, recovery capacity of AR for various correlations but for a model using an inhibitory matrix $\mathbf{W}'$.

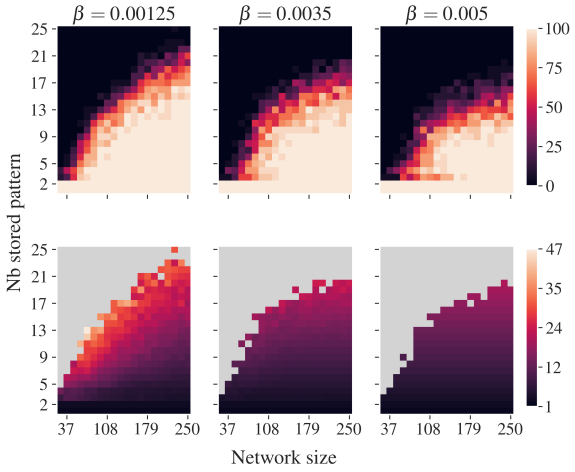

Figure 6.2 – Similar to Fig. 4.6, recovery capacity of AR for various values of β but for a model using an inhibitory matrix $\mathbf{W}'$.

6.5 . Drive-pattern correlation and spurious states

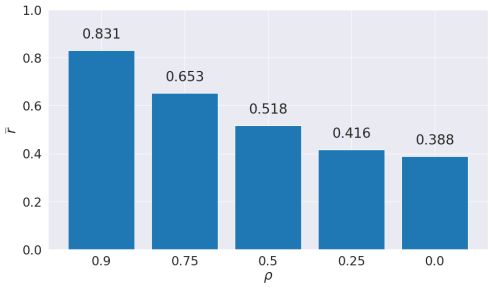

Figure 6.3 – For networks trained with sets of varying correlation ρ, we compute the synaptic drive of each unit $I_i^{syn} = \sum_j W_{ij}$. This synaptic drive is the main component influencing the dynamic of the network at the neutral state, when the leak is null. Then, we compute the average correlation between each stored pattern and the synaptic drive : $\bar{r} = \frac{1}{p}\sum_{\mu=1}^{p} r^\mu$ with $r^\mu = \text{Corr}(\mathbf{x}^\mu, \mathbf{I}^{syn})$. We can see that, the higher the ρ, the stronger the average correlation between the drive $\mathbf{I}^{syn}$ and stored patterns. At the neutral state, from which the network is initialized, stronger correlations induce a stronger push of the network state toward stored patterns. This could explain why our algorithm performs better with moderately correlated patterns.

6.6 . Parameters for simulations

Table 6.1 – CHN parameters

Name	Value	Description
ε_{sim}	10^{-6}	Convergence constant
r	1	Leak time constant
c	1	Integration time constant
δ	0.001	Euler step size

Table 6.2 – Gradient descent

Name	Value	Description
α	0.0001	Learning rate
ε_{learn}	10^{-6}	Convergence constant
u^{target}	6	Target potential winning units

Algorithm 10 McCallum's incremental pseudorehearsal protocol (bootstrap + sequential incorporation)

Input : Pattern sequence $\mathbf{x}^1, \dots, \mathbf{x}^{M_{\max}}$, base size B
Number of probes n_{probe}, pseudoitem cap P_{items}, noise parameters $(\nu_h, \sigma_{\text{G}})$
GDA hyperparameters (Alg. 13)
Initialize : Weight matrix $\mathbf{W} \leftarrow 0$
Bootstrap : Store $\{\mathbf{x}^1, \dots, \mathbf{x}^B\}$ in current $\mathbf{W}$ using the GDA (Alg. 13)
for $m = B+1$ to $M_{\max}$ **do**
    Incorporate $\mathbf{x}^m$ in current $\mathbf{W}$ using pseudorehearsal (Alg. 11)
end for

Algorithm 11 McCallum's pseudorehearsal incorporation. Check McCallum (1998) for more details.

Input : Current weight matrix $\mathbf{W}$, new pattern $\mathbf{x}^{\text{new}}$, number of probes n_{probe}, pseudoitem cap P_{items}
Heteroassociative noise fraction ν_h, Gaussian noise standard deviation σ_{G}
GDA hyperparameters (Alg. 13)
Initialize : Recovered set $\mathscr{R} \leftarrow \emptyset$
for $j = 1$ to n_{probe} **do**
    Sample random initial state $\mathbf{s}_0$
    Set $\mathbf{s}(0) \leftarrow \mathbf{s}_0$
    Simulate DHN dynamics (Sec. 2.3.4) until convergence to $\mathbf{s}^*$
    if $\mathbf{s}^* \notin \mathscr{R}$ and $-\mathbf{s}^* \notin \mathscr{R}$ **then**
        Add $\mathbf{s}^*$ to $\mathscr{R}$
    end if
end for
Prune $\mathscr{R}$ to at most P_{items} items
Define training set $\mathscr{T} \leftarrow \mathscr{R} \cup \{\mathbf{x}^{\text{new}}\}$
Apply noise to $\mathbf{x}^{\text{new}}$ (parameters ν_h, σ_{G}) during GDA; pseudoitems are left unchanged
Store $\mathscr{T}$ in current $\mathbf{W}$ using the GDA (Alg. 13)

Algorithm 12 Iterative GDA baseline (single-pattern updates, no rehearsal)

Input : Initial weight matrix $\mathbf{W}$; pattern sequence $\mathbf{x}^1, \dots, \mathbf{x}^{M_{\max}}$; network type $\in \{\text{DHN}, \text{CHN}\}$
GDA hyperparameters;
Optional (DHN only) : storage-noise parameters $(\nu_h, \sigma_{\text{G}})$
for $m = 1$ to $M_{\max}$ **do**
Set the training set $\mathscr{T} \leftarrow \{\mathbf{x}^m\}$ ▷ important : only the newly incorporated pattern
if network type is DHN **then**
Run Alg. 13 on $(\mathbf{W}, \mathscr{T})$ (optionally enabling storage noise)
else
Run Alg. 4 on $(\mathbf{W}, \mathscr{T})$ without reinitializing $\mathbf{W}$
▷ in practice : omit the initialization line $W_{ij} = 0$ in Alg. 4
end if
end for
Return : Updated weight matrix $\mathbf{W}$

Algorithm 13 McCallum's DHN GDA update step

Input : Current weight matrix $\mathbf{W}$; training set $\mathscr{T} = \{\mathbf{x}^\mu\}$; GDA hyperparameters;
Optional : storage-noise parameters $(\nu_h, \sigma_{\text{G}})$

Apply the DHN delta-learning / GDA procedure to store $\mathscr{T}$ into $\mathbf{W}$ starting from the current $\mathbf{W}$
(similar to Alg. 3; implementation details in McCallum (1998))
If noise is enabled, apply storage noise only to the newly stored pattern
Return : Updated weight matrix $\mathbf{W}$

Algorithm 14 Sleep-phase retrieval for continuous incorporation (CI), adapted from the autonomous retrieval procedure (Alg. 6) with the free phase omitted and a maximum number of retrieval attempts added.

Input : Trained network weights W_{ij}, number of stored patterns M, inhibitory potentiation rate β, maximum number of retrieval attempts $k_{\max}$
Output : Recovered pattern set $\mathscr{R}$, or failure
Initialize : $\mathbf{A} \leftarrow \mathbf{0}$, $\mathscr{R} \leftarrow \emptyset$, $j \leftarrow 0$
while $j < k_{\max}$ **do**
    Set neutral initial conditions : $\mathbf{u}(t=0) \leftarrow \mathbf{0}$
    Biased phase : integrate Eq. 4.4 until convergence to $\mathbf{u}(t_f) = \mathbf{u}^*$
    Let $\mathbf{v}^* \leftarrow \sigma(\mathbf{u}^*)$
    Read pattern $\mathbf{x}^*$ from $\mathbf{v}^*$ via thresholding (Eq. 4.2)
    if $\mathbf{x}^*$ is a spurious state **then**
        return failure
    end if
    if $\mathbf{x}^* \notin \mathscr{R}$ **then**
        $\mathscr{R} \leftarrow \mathscr{R} \cup \{\mathbf{x}^*\}$
        if $|\mathscr{R}| = M$ **then**
            return $\mathscr{R}$
        end if
    end if
    Update inhibitory weights : $\mathbf{A} \leftarrow \mathbf{A} + \beta\, \mathbf{v}^*$
    $j \leftarrow j + 1$
end while
return failure

Bibliography

Laurence F. Abbott and Peter Dayan. *Theoretical Neuroscience : Computational and Mathematical Modeling of Neural Systems*. Computational Neuroscience Series. MIT Press, Cambridge, MA, USA, August 2005. ISBN 978-0-262-54185-5.

Wickliffe C. Abraham. Metaplasticity : tuning synapses and networks for plasticity. *Nature Reviews Neuroscience*, 9(5) :387–387, May 2008. ISSN 1471-0048. doi : 10.1038/nrn2356. URL https://www.nature.com/articles/nrn2356. Publisher : Nature Publishing Group.

Daniel J. Amit, Hanoch Gutfreund, and H. Sompolinsky. Spin-glass models of neural networks. *Physical Review A*, 32(2) :1007–1018, August 1985a. ISSN 0556-2791. doi : 10.1103/PhysRevA.32.1007. URL https://link.aps.org/doi/10.1103/PhysRevA.32.1007.

Daniel J. Amit, Hanoch Gutfreund, and H. Sompolinsky. Storing Infinite Numbers of Patterns in a Spin-Glass Model of Neural Networks. *Physical Review Letters*, 55(14) :1530–1533, September 1985b. ISSN 0031-9007. doi : 10.1103/PhysRevLett.55.1530. URL https://link.aps.org/doi/10.1103/PhysRevLett.55.1530.

Igor Antonov, Irina Antonova, Eric R. Kandel, and Robert D. Hawkins. The Contribution of Activity-Dependent Synaptic Plasticity to Classical Conditioning in Aplysia. *Journal of Neuroscience*, 21(16) :6413–6422, August 2001. ISSN 0270-6474, 1529-2401. doi : 10.1523/JNEUROSCI.21-16-06413.2001. URL https://www.jneurosci.org/content/21/16/6413.

H.C. Barron, T.P. Vogels, U.E. Emir, T.R. Makin, J. O'Shea, S. Clare, S. Jbabdi, R.J. Dolan, and T.E.J. Behrens. Unmasking Latent Inhibitory Connections in Human Cortex to Reveal Dormant Cortical Memories. *Neuron*, 90(1) : 191–203, April 2016. ISSN 08966273. doi : 10.1016/j.neuron.2016.02.031. URL https://linkinghub.elsevier.com/retrieve/pii/S0896627316001689.

Helen C. Barron, Tim P. Vogels, Timothy E. Behrens, and Mani Ramaswami. Inhibitory engrams in perception and memory. *Proceedings of the National Academy of Sciences*, 114(26) :6666–6674, June 2017. doi : 10.1073/pnas.1701812114. URL https://www.pnas.org/doi/10.1073/pnas.1701812114. Publisher : Proceedings of the National Academy of Sciences.

Alison L. Barth and James F. A. Poulet. Experimental evidence for sparse firing in the neocortex. *Trends in Neurosciences*, 35(6) :345–355, June 2012. ISSN

0166-2236, 1878-108X. doi : 10.1016/j.tins.2012.03.008. URL https://www.cell.com/trends/neurosciences/abstract/S0166-2236(12)00051-3.

Jan Benda and Andreas V. M. Herz. A universal model for spike-frequency adaptation. *Neural Computation*, 15(11) :2523–2564, November 2003. ISSN 0899-7667. doi : 10.1162/089976603322385063.

Ben Varkey Benjamin, Peiran Gao, Emmett McQuinn, Swadesh Choudhary, Anand R. Chandrasekaran, Jean-Marie Bussat, Rodrigo Alvarez-Icaza, John V. Arthur, Paul A. Merolla, and Kwabena Boahen. Neurogrid : A Mixed-Analog-Digital Multichip System for Large-Scale Neural Simulations. *Proceedings of the IEEE*, 102(5) :699–716, May 2014. ISSN 0018-9219, 1558-2256. doi : 10.1109/JPROC.2014.2313565. URL http://ieeexplore.ieee.org/document/6805187/.

Marcus K. Benna and Stefano Fusi. Computational principles of synaptic memory consolidation. *Nature Neuroscience*, 19(12) :1697–1706, December 2016. ISSN 1546-1726. doi : 10.1038/nn.4401. URL https://www.nature.com/articles/nn.4401. Publisher : Nature Publishing Group.

Jeffrey R. Binder, Rutvik H. Desai, William W. Graves, and Lisa L. Conant. Where Is the Semantic System ? A Critical Review and Meta-Analysis of 120 Functional Neuroimaging Studies. *Cerebral Cortex (New York, NY)*, 19(12) :2767–2796, December 2009. ISSN 1047-3211. doi : 10.1093/cercor/bhp055. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2774390/.

Giulio Biroli, Tony Bonnaire, Valentin de Bortoli, and Marc Mézard. Dynamical regimes of diffusion models. *Nature Communications*, 15(1) :9957, November 2024. ISSN 2041-1723. doi : 10.1038/s41467-024-54281-3. URL https://www.nature.com/articles/s41467-024-54281-3.

T. V. P. Bliss and G. L. Collingridge. A synaptic model of memory : long-term potentiation in the hippocampus. *Nature*, 361(6407) :31–39, January 1993. ISSN 0028-0836, 1476-4687. doi : 10.1038/361031a0. URL https://www.nature.com/articles/361031a0.

T. V. P. Bliss and T. Lømo. Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path. *The Journal of Physiology*, 232(2) :331–356, July 1973a. ISSN 0022-3751. doi : 10.1113/jphysiol.1973.sp010273. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1350458/.

T. V. P. Bliss and T. Lømo. Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path. *The Journal of Physiology*, 232(2) :331–356, July 1973b. ISSN 0022-3751, 1469-7793. doi : 10.1113/jphysiol.1973.

sp010273. URL https://physoc.onlinelibrary.wiley.com/doi/10.1113/jphysiol.1973.sp010273.

K.A. Boahen. Point-to-point connectivity between neuromorphic chips using address events. *IEEE Transactions on Circuits and Systems II : Analog and Digital Signal Processing*, 47(5) :416–434, May 2000. ISSN 10577130. doi : 10.1109/82.842110. URL http://ieeexplore.ieee.org/document/842110/.

Léon Bottou. Large-Scale Machine Learning with Stochastic Gradient Descent. In Yves Lechevallier and Gilbert Saporta (eds.), *Proceedings of COMPSTAT'2010*, pp. 177–186. Physica-Verlag (Springer), Heidelberg, 2010. doi : 10.1007/978-3-7908-2604-3_16. URL https://leon.bottou.org/publications/pdf/compstat-2010.pdf.

Leonardo Enzo Brito da Silva, Islam Elnabarawy, and Donald C. Wunsch. A survey of adaptive resonance theory neural network models for engineering applications. *Neural Networks*, 120 :167–203, December 2019. ISSN 0893-6080. doi : 10.1016/j.neunet.2019.09.012. URL https://www.sciencedirect.com/science/article/pii/S0893608019302734.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark Experience for General Continual Learning : a Strong, Simple Baseline, October 2020. URL http://arxiv.org/abs/2004.07211. arXiv :2004.07211 [stat].

Denise J. Cai, Daniel Aharoni, Tristan Shuman, Justin Shobe, Jeremy Biane, Weilin Song, Brandon Wei, Michael Veshkini, Mimi La-Vu, Jerry Lou, Sergio E. Flores, Isaac Kim, Yoshitake Sano, Miou Zhou, Karsten Baumgaertel, Ayal Lavi, Masakazu Kamata, Mark Tuszynski, Mark Mayford, Peyman Golshani, and Alcino J. Silva. A shared neural ensemble links distinct contextual memories encoded close in time. *Nature*, 534(7605) :115–118, June 2016. ISSN 0028-0836, 1476-4687. doi : 10.1038/nature17955. URL https://www.nature.com/articles/nature17955.

Antonio Caputi, Sarah Melzer, Magdalena Michael, and Hannah Monyer. The long and short of GABAergic neurons. *Current Opinion in Neurobiology*, 23(2) :179–186, April 2013. ISSN 0959-4388. doi : 10.1016/j.conb.2013.01.021. URL https://www.sciencedirect.com/science/article/pii/S0959438813000378.

Gail A. Carpenter and Stephen Grossberg. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, 37(1) :54–115, January 1987. ISSN 0734-189X. doi : 10.1016/S0734-189X(87)80014-2. URL https://www.sciencedirect.com/science/article/pii/S0734189X87800142.

Gail A. Carpenter, Stephen Grossberg, and David B. Rosen. Fuzzy ART : Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4(6) :759–771, January 1991. ISSN 0893-6080. doi : 10.1016/0893-6080(91)90056-B. URL https://www.sciencedirect.com/science/article/pii/089360809190056B.

Le Chang and Doris Y. Tsao. The Code for Facial Identity in the Primate Brain. *Cell*, 169(6) :1013–1028.e14, June 2017. ISSN 0092-8674. doi : 10.1016/j.cell.2017.05.011. URL https://www.sciencedirect.com/science/article/pii/S009286741730538X.

Joseph Cichon and Wen-Biao Gan. Branch-specific dendritic Ca2+ spikes cause persistent synaptic plasticity. *Nature*, 520(7546) :180–185, April 2015. ISSN 1476-4687. doi : 10.1038/nature14251. URL https://www.nature.com/articles/nature14251. Publisher : Nature Publishing Group.

Ami Citri and Robert C. Malenka. Synaptic Plasticity : Multiple Forms, Functions, and Mechanisms. *Neuropsychopharmacology*, 33(1) :18–41, January 2008. ISSN 1740-634X. doi : 10.1038/sj.npp.1301559. URL https://www.nature.com/articles/1301559.

Brittany C. Clawson, Jaclyn Durkin, and Sara J. Aton. Form and Function of Sleep Spindles across the Lifespan. *Neural Plasticity*, 2016 :1–16, 2016. ISSN 2090-5904, 1687-5443. doi : 10.1155/2016/6936381. URL http://www.hindawi.com/journals/np/2016/6936381/.

Francis Crick. The recent excitement about neural networks. *Nature*, 337 (6203) :129–132, January 1989. ISSN 0028-0836, 1476-4687. doi : 10.1038/337129a0. URL https://www.nature.com/articles/337129a0.

Neil Davey, Ray Frank, Steve Hunt, Rod Adams, and Lee Calcraft. High capacity associative memory models - Binary and bipolar representation. *Applied Soft Computing - ASC*, January 2004.

Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, Yuyun Liao, Chit-Kwan Lin, Andrew Lines, Ruokun Liu, Deepak Mathaikutty, Steven McCoy, Arnab Paul, Jonathan Tse, Guruguhanathan Venkataramanan, Yi-Hsin Weng, Andreas Wild, Yoonseok Yang, and Hong Wang. Loihi : A Neuromorphic Manycore Processor with On-Chip Learning. *IEEE Micro*, 38(1) :82–99, January 2018. ISSN 0272-1732, 1937-4143. doi : 10.1109/MM.2018.112130359. URL https://ieeexplore.ieee.org/document/8259423/.

Sigurd Diederich and Manfred Opper. Learning of correlated patterns in spin-glass networks by local learning rules. *Physical Review Letters*, 58(9) :949–

952, March 1987. ISSN 0031-9007. doi : 10.1103/PhysRevLett.58.949. URL https://link.aps.org/doi/10.1103/PhysRevLett.58.949.

James A. D'amour and Robert C. Froemke. Inhibitory and Excitatory Spike-Timing-Dependent Plasticity in the Auditory Cortex. *Neuron*, 86(2) :514–528, April 2015. ISSN 08966273. doi : 10.1016/j.neuron.2015.03.014. URL https://linkinghub.elsevier.com/retrieve/pii/S089662731500210X.

Florian Engert and Tobias Bonhoeffer. Dendritic spine changes associated with hippocampal long-term synaptic plasticity. *Nature*, 399(6731) :66–70, May 1999. ISSN 0028-0836, 1476-4687. doi : 10.1038/19978. URL https://www.nature.com/articles/19978.

Michael Jan Fauth and Mark Cw Van Rossum. Self-organized reactivation maintains and reinforces memories despite synaptic turnover. *eLife*, 8 : e43717, May 2019. ISSN 2050-084X. doi : 10.7554/eLife.43717. URL https://elifesciences.org/articles/43717.

D. J. Felleman and D. C. Van Essen. Distributed Hierarchical Processing in the Primate Cerebral Cortex. *Cerebral Cortex*, 1(1) :1–47, January 1991. ISSN 1047-3211, 1460-2199. doi : 10.1093/cercor/1.1.1. URL https://academic.oup.com/cercor/article-lookup/doi/10.1093/cercor/1.1.1.

L. M. J. Fernandez and A. Lüthi. Sleep spindles : Mechanisms and functions. *Physiological Reviews*, 2020. URL https://www.grc.com/dev/Historical/TheHalo/SMR/SLEEP%20SPINDLES--MECHANISMS-AND-FUNCTIONS.pdf.

J.F. Fontanari and W.K. Theumann. On the storage of correlated patterns in Hopfield's model. *Journal de Physique*, 51(5) :375–386, 1990. ISSN 0302-0738. doi : 10.1051/jphys:01990005105037500. URL http://www.edpsciences.org/10.1051/jphys:01990005105037500.

D. J. Foster and M. A. Wilson. Reverse replay of behavioural sequences in hippocampal place cells during the awake state. *Nature*, 2006. URL https://cogsci.ucsd.edu/~chiba/Awake1.pdf.

Winrich A Freiwald, Doris Y Tsao, and Margaret S Livingstone. A face feature space in the macaque temporal lobe. *Nature neuroscience*, 12(9) :1187–1196, September 2009. ISSN 1097-6256. doi : 10.1038/nn.2363. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2819705/.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4), 1999.

Min Fu and Yi Zuo. Experience-dependent structural plasticity in the cortex. *Trends in Neurosciences*, 34(4) :177–187, April 2011. ISSN 1878-108X. doi : 10.1016/j.tins.2011.02.001.

Steve Furber. Large-scale neuromorphic computing systems. *Journal of Neural Engineering*, 13(5) :051001, October 2016. ISSN 1741-2560, 1741-2552. doi : 10.1088/1741-2560/13/5/051001. URL https://iopscience.iop.org/article/10.1088/1741-2560/13/5/051001.

Stefano Fusi and L F Abbott. Limits on the memory storage capacity of bounded synapses. *Nature Neuroscience*, 10(4) :485–493, April 2007. ISSN 1097-6256, 1546-1726. doi : 10.1038/nn1859. URL https://www.nature.com/articles/nn1859.

Stefano Fusi, Patrick J. Drew, and L. F. Abbott. Cascade Models of Synaptically Stored Memories. *Neuron*, 45(4) :599–611, February 2005. ISSN 0896-6273. doi : 10.1016/j.neuron.2005.02.001. URL https://www.cell.com/neuron/abstract/S0896-6273(05)00117-0. Publisher : Elsevier.

H. Gelbard-Sagiv, R. Mukamel, M. Harel, R. Malach, and I. Fried. Internally generated reactivation of single neurons in human hippocampus during free recall. *Science*, 322(5898) :96–101, 2008. doi : 10.1126/science.1164685. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC2650423/pdf/nihms87358.pdf.

Anna K. Gillespie, Daniela A. Astudillo Maya, Eric L. Denovellis, Daniel F. Liu, David B. Kastner, Michael E. Coulter, Demetris K. Roumis, Uri T. Eden, and Loren M. Frank. Hippocampal replay reflects specific past experiences rather than a plan for subsequent choice. *Neuron*, 109(19) :3149–3163.e6, 2021. doi : 10.1016/j.neuron.2021.07.029. URL https://pubmed.ncbi.nlm.nih.gov/34450026/.

Corinna Giorgi and Silvia Marinelli. Roles and Transcriptional Responses of Inhibitory Neurons in Learning and Memory. *Frontiers in Molecular Neuroscience*, 14, June 2021. ISSN 1662-5099. doi : 10.3389/fnmol.2021.689952. URL https://www.frontiersin.org/journals/molecular-neuroscience/articles/10.3389/fnmol.2021.689952/full. Publisher : Frontiers.

Ryan Golden, Jean Erik Delanois, Pavel Sanda, and Maxim Bazhenov. Sleep prevents catastrophic forgetting in spiking neural networks by forming a joint synaptic weight representation. *PLOS Computational Biology*, 18 (11) :e1010628, November 2022. ISSN 1553-7358. doi : 10.1371/journal.pcbi.1010628. URL https://dx.plos.org/10.1371/journal.pcbi.1010628.

Ian J. Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An Empirical Investigation of Catastrophic Forgetting in Gradient-Based Neural Networks. *arXiv*, 2013. URL https://arxiv.org/pdf/1312.6211.

Chris Gorman, Anthony Robins, and Alistair Knott. Hopfield networks as a model of prototype-based category learning : A method to distinguish trained, spurious, and prototypical attractors. *Neural Networks*, 91 :76–84, July 2017. ISSN 08936080. doi : 10.1016/j.neunet.2017.04.007. URL https://linkinghub.elsevier.com/retrieve/pii/S0893608017300874.

Daniel L. Greenberg and Mieke Verfaeilli. Interdependence of episodic and semantic memory : Evidence from neuropsychology. *Journal of the International Neuropsychological Society : JINS*, 16(5) :748–753, September 2010. ISSN 1355-6177. doi : 10.1017/S1355617710000676. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2952732/.

Stephen Grossberg. How does a brain build a cognitive code? *Psychological Review*, 87(1) :1–51, 1980. ISSN 1939-1471. doi : 10.1037/0033-295X.87.1.1. Place : US Publisher : American Psychological Association.

Stephen Grossberg. Adaptive Resonance Theory : how a brain learns to consciously attend, learn, and recognize a changing world. *Neural Networks : The Official Journal of the International Neural Network Society*, 37 :1–47, January 2013. ISSN 1879-2782. doi : 10.1016/j.neunet.2012.09.017.

Segundo Jose Guzman, Alois Schlögl, Michael Frotscher, and Peter Jonas. Synaptic mechanisms of pattern completion in the hippocampal CA3 network. *Science*, 353(6304) :1117–1123, September 2016. ISSN 0036-8075, 1095-9203. doi : 10.1126/science.aaf1836. URL https://www.science.org/doi/10.1126/science.aaf1836.

Go Eun Ha and Eunji Cheong. Spike Frequency Adaptation in Neurons of the Central Nervous System. *Experimental Neurobiology*, 26(4) :179–185, August 2017. ISSN 1226-2560. doi : 10.5607/en.2017.26.4.179. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC5597548/.

Christopher D. Harvey and Karel Svoboda. Locally dynamic synaptic learning rules in pyramidal neuron dendrites. *Nature*, 450(7173) :1195–1200, December 2007. ISSN 0028-0836, 1476-4687. doi : 10.1038/nature06416. URL https://www.nature.com/articles/nature06416.

James V. Haxby, M. Ida Gobbini, Maura L. Furey, Alumit Ishai, Jennifer L. Schouten, and Pietro Pietrini. Distributed and Overlapping Representations of Faces and Objects in Ventral Temporal Cortex. *Science*, 293(5539) :2425–2430, September 2001. ISSN 0036-8075, 1095-9203. doi : 10.1126/science.1063736. URL https://www.science.org/doi/10.1126/science.1063736.

D.O. Hebb. *The Organization of Behavior*. 1949.

Melissa Hernández-Frausto, Olesia M. Bilash, Arjun V. Masurkar, and Jayeeta Basu. Local and long-range GABAergic circuits in hippocampal area CA1 and their link to Alzheimer's disease. *Frontiers in Neural Circuits*, 17 :1223891, September 2023. ISSN 1662-5110. doi : 10.3389/fncir.2023.1223891. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC10570439/.

John Hertz, Anders Krogh, and Richard G. Palmer. *Introduction to the theory of neural computation*. Number v. 1 in Santa Fe Institute studies in the sciences of complexity. Addison-Wesley Pub. Co, Redwood City, Calif, 1991. ISBN 978-0-201-50395-1 978-0-201-51560-2.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models, December 2020. URL http://arxiv.org/abs/2006.11239. arXiv :2006.11239 [cs].

J J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79 (8) :2554–2558, April 1982. ISSN 0027-8424, 1091-6490. doi : 10.1073/pnas.79.8.2554. URL https://pnas.org/doi/full/10.1073/pnas.79.8.2554.

J J Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences*, 81(10) :3088–3092, May 1984. ISSN 0027-8424, 1091-6490. doi : 10.1073/pnas.81.10.3088. URL https://pnas.org/doi/full/10.1073/pnas.81.10.3088.

John Hopfield and D Tank. Neural Computation of Decisions in Optimization Problems. *Biological cybernetics*, 52 :141–52, February 1985. doi : 10.1007/BF00339943.

John J. Hopfield. Neurodynamics of mental exploration. *Proceedings of the National Academy of Sciences*, 107(4) :1648–1653, January 2010. doi : 10.1073/pnas.0913991107. URL https://www.pnas.org/doi/10.1073/pnas.0913991107. Publisher : Proceedings of the National Academy of Sciences.

Wen-Hsien Hou, Meet Jariwala, Kai-Yi Wang, Anna Seewald, Yu-Ling Lin, Yi-Chen Liou, Alessia Ricci, Francesco Ferraguti, Cheng-Chang Lien, and Marco Capogna. Inhibitory fear memory engram in the mouse central lateral amygdala. *Cell Reports*, 43(8), August 2024. ISSN 2211-1247. doi : 10.1016/j.celrep.2024.114468. URL https://www.cell.com/cell-reports/abstract/S2211-1247(24)00797-6. Publisher : Elsevier.

Eriola Hoxha, Filippo Tempia, Pellegrino Lippiello, and Maria Concetta Miniaci. Modulation, Plasticity and Pathophysiology of the Parallel Fiber-Purkinje Cell Synapse. *Frontiers in Synaptic Neuroscience*, 8, November 2016. ISSN 1663-3563. doi : 10.3389/fnsyn.2016.00035. URL https://www.frontiersin.org/journals/synaptic-neuroscience/articles/10.3389/fnsyn.2016.00035/full.

Ferenc Huszár. Note on the quadratic penalties in elastic weight consolidation. *Proceedings of the National Academy of Sciences*, 115(11) :E2496–E2497, March 2018. doi : 10.1073/pnas.1717042115. URL https://www.pnas.org/doi/full/10.1073/pnas.1717042115. Publisher : Proceedings of the National Academy of Sciences.

Daniel Im, Mohamed Belghazi, and Roland Memisevic. Conservativeness of Untied Auto-Encoders. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1), February 2016. ISSN 2374-3468, 2159-5399. doi : 10.1609/aaai.v30i1.10268. URL https://ojs.aaai.org/index.php/AAAI/article/view/10268.

Giacomo Indiveri and Shih-Chii Liu. Memory and information processing in neuromorphic systems. *Proceedings of the IEEE*, 103(8) :1379–1397, August 2015. ISSN 0018-9219, 1558-2256. doi : 10.1109/JPROC.2015.2444094. URL http://arxiv.org/abs/1506.03264. arXiv :1506.03264 [cs].

Jeffry S. Isaacson and Massimo Scanziani. How Inhibition Shapes Cortical Activity. *Neuron*, 72(2) :231–243, October 2011. ISSN 0896-6273. doi : 10.1016/j.neuron.2011.09.027. URL https://www.cell.com/neuron/abstract/S0896-6273(11)00879-8. Publisher : Elsevier.

Haruka Isawa, Haruna Matsushita, and Yoshifumi Nishio. Fuzzy Adaptive Resonance Theory Combining Overlapped Category in consideration of connections. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pp. 3595–3600, June 2008. doi : 10.1109/IJCNN.2008.4634312. URL https://ieeexplore.ieee.org/document/4634312. ISSN : 2161-4407.

S. P. Jadhav, C. Kemere, P. W. German, and L. M. Frank. Awake hippocampal sharp-wave ripples support spatial memory. *Science*, 2012. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC4441285/.

Daniel Jercog, Alex Roxin, Peter Barthó, Artur Luczak, Albert Compte, and Jaime De La Rocha. UP-DOWN cortical dynamics reflect state transitions in a bistable network. *eLife*, 6, August 2017. ISSN 2050-084X. doi : 10.7554/elife.22425. URL https://elifesciences.org/articles/22425.

D. Ji and M. A. Wilson. Coordinated memory replay in the visual cortex and hippocampus during sleep. *Nature Neuroscience*, 2007. URL https://www.cs.utexas.edu/~dana/Wilson.pdf.

Sheena A. Josselyn and Susumu Tonegawa. Memory engrams : Recalling the past and imagining the future. *Science*, 367(6473) :eaaw4325, January 2020. ISSN 0036-8075, 1095-9203. doi : 10.1126/science.aaw4325. URL https://www.science.org/doi/10.1126/science.aaw4325.

I. Kanter and H. Sompolinsky. Associative recall of memory without errors. *Physical Review A*, 35(1) :380–392, January 1987. ISSN 0556-2791. doi : 10.1103/PhysRevA.35.380. URL https://link.aps.org/doi/10.1103/PhysRevA.35.380.

Mattias P. Karlsson and Loren M. Frank. Awake replay of remote experiences in the hippocampus. *Nature Neuroscience*, 12(7) :913–918, 2009. doi : 10.1038/nn.2344. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2775135/.

Adam Kepecs and Gordon Fishell. Interneuron cell types are fit to function. *Nature*, 505(7483) :318–326, January 2014. ISSN 1476-4687. doi : 10.1038/nature12983. URL https://www.nature.com/articles/nature12983. Publisher : Nature Publishing Group.

R. P. Kesner and E. T. Rolls. A computational theory of hippocampal function, and tests of the theory : New developments. *Neuroscience & Biobehavioral Reviews*, 48 :92–147, 2015. doi : 10.1016/j.neubiorev.2014.11.009. URL https://cenl.ucsd.edu/Jclub/Kesner-Rolls%2BComputational-theory-hippocampal-function%2BNeuroBiobehavRev%2B2015.pdf.

Kyuree Kim, Min Suk Song, Hwiho Hwang, Sungmin Hwang, and Hyungjin Kim. A comprehensive review of advanced trends : from artificial synapses to neuromorphic systems with consideration of non-ideal effects. *Frontiers in Neuroscience*, 18, April 2024. ISSN 1662-453X. doi : 10.3389/fnins.2024.1279708. URL https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2024.1279708/full.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13) :3521–3526, March 2017. ISSN 0027-8424, 1091-6490. doi : 10.1073/pnas.1611835114. URL https://pnas.org/doi/full/10.1073/pnas.1611835114.

Jayaraj N. Kodangattil, Matthieu Dacher, Michael E. Authement, and Fereshteh S. Nugent. Spike timing-dependent plasticity at GABAergic synapses in the ventral tegmental area. *The Journal of Physiology*, 591(19) :4699–4710, October 2013. ISSN 0022-3751, 1469-7793. doi : 10.1113/jphysiol.2013.257873. URL https://physoc.onlinelibrary.wiley.com/doi/10.1113/jphysiol.2013.257873.

Ling-Wei Kong, Gene A. Brewer, and Ying-Cheng Lai. Reservoir-computing based associative memory and itinerancy for complex dynamical attractors. *Nature Communications*, 15(1) :4840, June 2024. ISSN 2041-1723. doi : 10.1038/s41467-024-49190-4. URL https://www.nature.com/articles/s41467-024-49190-4.

Vignesh Kothapalli. Neural Collapse : A Review on Modelling Principles and Generalization, April 2023. URL http://arxiv.org/abs/2206.04041. arXiv :2206.04041 [cs].

Nikolaus Kriegeskorte. Representational similarity analysis – connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2008. ISSN 16625137. doi : 10.3389/neuro.06.004.2008. URL http://journal.frontiersin.org/article/10.3389/neuro.06.004.2008/abstract.

Nikolaus Kriegeskorte, Elia Formisano, Bettina Sorger, and Rainer Goebel. Individual faces elicit distinct response patterns in human anterior temporal cortex. *Proceedings of the National Academy of Sciences*, 104(51) : 20600–20605, December 2007. ISSN 0027-8424, 1091-6490. doi : 10.1073/pnas.0705654104. URL https://pnas.org/doi/full/10.1073/pnas.0705654104.

Dmitry Krotov and John Hopfield. Large Associative Memory Problem in Neurobiology and Machine Learning, April 2021. URL http://arxiv.org/abs/2008.06996. arXiv :2008.06996 [q-bio].

Dmitry Krotov and John J. Hopfield. Dense Associative Memory for Pattern Recognition, September 2016. URL http://arxiv.org/abs/1606.01164. arXiv :1606.01164 [cs].

Kishore V Kuchibhotla, Jonathan V Gill, Grace W Lindsay, Eleni S Papadoyannis, Rachel E Field, Tom A Hindmarsh Sten, Kenneth D Miller, and Robert C Froemke. Parallel processing by cortical inhibition enables context-dependent behavior. *Nature Neuroscience*, 20(1) :62–71, January 2017. ISSN 1097-6256, 1546-1726. doi : 10.1038/nn.4436. URL https://www.nature.com/articles/nn.4436.

Szabolcs Káli and Peter Dayan. Off-line replay maintains declarative memories in a model of hippocampal-neocortical interactions. *Nature Neuroscience*, 7 (3) :286–294, March 2004. ISSN 1097-6256, 1546-1726. doi : 10.1038/nn1202. URL https://www.nature.com/articles/nn1202.

Stephan Lammel, Daniela I. Ion, Jochen Roeper, and Robert C. Malenka. Projection-specific modulation of dopamine neuron synapses by aversive and rewarding stimuli. *Neuron*, 70(5) :855–862, June 2011. ISSN 0896-6273. doi : 10.1016/j.neuron.2011.03.025. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC3112473/.

Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. The MNIST database of handwritten digits. Technical report, 1998. URL http://yann.lecun.com/exdb/mnist/.

A. K. Lee and M. A. Wilson. Memory of sequential experience in the hippocampus during slow-wave sleep. *Neuron*, 2002. URL https://www.cell.com/neuron/pdf/S0896-6273(02)01096-6.pdf.

X.-G. Li, P. Somogyi, A. Ylinen, and G. Buzsáki. The hippocampal CA3 network : An in vivo intracellular labeling study. *Journal of Comparative Neurology*, 339(2) :181–208, January 1994. ISSN 0021-9967, 1096-9861. doi : 10.1002/cne.903390204. URL https://onlinelibrary.wiley.com/doi/10.1002/cne.903390204.

Zonglin Li, Chong You, Srinadh Bhojanapalli, Daliang Li, Ankit Singh Rawat, Sashank J. Reddi, Ke Ye, Felix Chern, Felix Yu, Ruiqi Guo, and Sanjiv Kumar. The Lazy Neuron Phenomenon : On Emergence of Activation Sparsity in Transformers, June 2023. URL http://arxiv.org/abs/2210.06313. arXiv :2210.06313 [cs].

Qun Liu and Supratik Mukhopadhyay. Unsupervised Learning using Pretrained CNN and Associative Memory Bank, May 2018. URL http://arxiv.org/abs/1805.01033. arXiv :1805.01033 [cs].

Xu Liu, Steve Ramirez, Petti T. Pang, Corey B. Puryear, Arvind Govindarajan, Karl Deisseroth, and Susumu Tonegawa. Optogenetic stimulation of a hippocampal engram activates fear memory recall. *Nature*, 484(7394) :381–385, April 2012. ISSN 0028-0836, 1476-4687. doi : 10.1038/nature11028. URL https://www.nature.com/articles/nature11028.

Simona Lodato and Paola Arlotta. Generating Neuronal Diversity in the Mammalian Cerebral Cortex. *Annual Review of Cell and Developmental Biology*, 31 (Volume 31, 2015) :699–720, November 2015. ISSN 1081-0706, 1530-8995. doi : 10.1146/annurev-cellbio-100814-125353. URL https://www.annualreviews.

org/content/journals/10.1146/annurev-cellbio-100814-125353. Publisher : Annual Reviews.

Michael London and Michael Häusser. Dendritic computation. *Annual Review of Neuroscience*, 28 :503–532, 2005. ISSN 0147-006X. doi : 10.1146/annurev.neuro.28.061604.135703.

K. Louie and M. A. Wilson. Temporally structured replay of awake hippocampal ensemble activity during REM sleep. *Neuron*, 2001. URL https://www.its.caltech.edu/~jkenny/nb250c/papers/Louie%20%26%20Wilson%202001.pdf.

Artur Luczak, Peter Barthó, Stephan L. Marguet, György Buzsáki, and Kenneth D. Harris. Sequential structure of neocortical spontaneous activity *in vivo*. *Proceedings of the National Academy of Sciences*, 104(1) :347–352, January 2007. ISSN 0027-8424, 1091-6490. doi : 10.1073/pnas.0605643104. URL https://pnas.org/doi/full/10.1073/pnas.0605643104.

Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-time computing without stable states : a new framework for neural computation based on perturbations. *Neural Computation*, 14(11) :2531–2560, November 2002. ISSN 0899-7667. doi : 10.1162/089976602760407955.

Robert C. Malenka, Barrie Lancaster, and Robert S. Zucker. Temporal limits on the rise in postsynaptic calcium required for the induction of long-term potentiation. *Neuron*, 9(1) :121–128, July 1992. ISSN 0896-6273. doi : 10.1016/0896-6273(92)90227-5. URL https://www.cell.com/neuron/abstract/0896-6273(92)90227-5.

Roberto Malinow and Robert C. Malenka. AMPA Receptor Trafficking and Synaptic Plasticity. *Annual Review of Neuroscience*, 25(Volume 25, 2002) : 103–126, March 2002. ISSN 0147-006X, 1545-4126. doi : 10.1146/annurev.neuro.25.112701.142758. URL https://www.annualreviews.org/content/journals/10.1146/annurev.neuro.25.112701.142758.

Eve Marder. Neuromodulation Of Neuronal Circuits : Back To The Future. *Neuron*, 76(1) :1–11, October 2012. ISSN 0896-6273. doi : 10.1016/j.neuron.2012.09.010. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3482119/.

Henry Markram, Joachim Lübke, Michael Frotscher, and Bert Sakmann. Regulation of Synaptic Efficacy by Coincidence of Postsynaptic APs and EPSPs. *Science*, 275(5297) :213–215, January 1997. ISSN 0036-8075, 1095-9203. doi : 10.1126/science.275.5297.213. URL https://www.science.org/doi/10.1126/science.275.5297.213.

D. Marr. Simple memory : a theory for archicortex. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 262(841) :23–81, 1971. doi : 10.1098/rstb.1971.0078. URL https://www.cs.cmu.edu/afs/cs/academic/class/15883-f17/readings/marr-1971.pdf.

Giordano De Marzo and Giulio Iannelli. Effect of spatial correlations on Hopfield Neural Network and Dense Associative Memories. *Physica A : Statistical Mechanics and its Applications*, 612 :128487, February 2023. ISSN 03784371. doi : 10.1016/j.physa.2023.128487. URL http://arxiv.org/abs/2207.05218. arXiv :2207.05218 [cond-mat].

Hayden McAlister, Anthony Robins, and Lech Szymanski. Prototype Analysis in Hopfield Networks with Hebbian Learning, May 2024. URL http://arxiv.org/abs/2407.03342. arXiv :2407.03342 [cs].

Simon McCallum. Catastrophic Forgetting and the Pseudorehearsal Solution in Hopfield-type Networks. *Connection Science*, 10(2) :121–135, June 1998. ISSN 0954-0091, 1360-0494. doi : 10.1080/095400998116530. URL http://www.tandfonline.com/doi/abs/10.1080/095400998116530.

James L. McClelland, Bruce L. McNaughton, and Randall C. O'Reilly. 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(3) :419–457, 1995. ISSN 1939-1471. doi : 10.1037/0033-295X.102.3.419. Place : US.

Michael McCloskey and Neal J. Cohen. Catastrophic Interference in Connectionist Networks : The Sequential Learning Problem. In Gordon H. Bower (ed.), *Psychology of Learning and Motivation*, volume 24, pp. 109–165. Academic Press, San Diego, 1989. doi : 10.1016/S0079-7421(08)60536-8. URL https://www.andywills.info/hbab/mccloskeycohen.pdf.

Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4) : 115–133, December 1943. ISSN 1522-9602. doi : 10.1007/BF02478259. URL https://doi.org/10.1007/BF02478259.

C. Mead. Neuromorphic electronic systems. *Proceedings of the IEEE*, 78(10) : 1629–1636, October 1990. ISSN 1558-2256. doi : 10.1109/5.58356. URL https://ieeexplore.ieee.org/document/58356.

Niklas M. Melton, Leonardo Enzo Brito da Silva, Sasha Petrenko, and Donald C. Wunsch. Deep ARTMAP : Generalized Hierarchical Learning with Adaptive Resonance Theory, March 2025. URL http://arxiv.org/abs/2503.07641. arXiv :2503.07641 [cs].

Paul A. Merolla, John V. Arthur, Rodrigo Alvarez-Icaza, Andrew S. Cassidy, Jun Sawada, Filipp Akopyan, Bryan L. Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, Bernard Brezzo, Ivan Vo, Steven K. Esser, Rathinakumar Appuswamy, Brian Taba, Arnon Amir, Myron D. Flickner, William P. Risk, Rajit Manohar, and Dharmendra S. Modha. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197) :668–673, August 2014. doi : 10.1126/science.1254642. URL https://www.science.org/doi/10.1126/science.1254642.

Yuanyuan Mi, C. C. Alan Fung, K. Y. Michael Wong, and Si Wu. Spike Frequency Adaptation Implements Anticipative Tracking in Continuous Attractor Neural Networks. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://papers.nips.cc/paper_files/paper/2014/hash/57c76ace5d28ad312d72e9d5923fb097-Abstract.html.

R Miles and R K Wong. Excitatory synaptic interactions between CA3 neurones in the guinea-pig hippocampus. *The Journal of Physiology*, 373(1) : 397–418, April 1986. ISSN 0022-3751, 1469-7793. doi : 10.1113/jphysiol.1986.sp016055. URL https://physoc.onlinelibrary.wiley.com/doi/10.1113/jphysiol.1986.sp016055.

Guo-li Ming and Hongjun Song. Adult Neurogenesis in the Mammalian Brain : Significant Answers and Significant Questions. *Neuron*, 70(4) :687, May 2011. doi : 10.1016/j.neuron.2011.05.001. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC3106107/.

R. G. M. Morris, E. Anderson, G. S. Lynch, and M. Baudry. Selective impairment of learning and blockade of long-term potentiation by an N-methyl-D-aspartate receptor antagonist, AP5. *Nature*, 319(6056) :774–776, February 1986. ISSN 0028-0836, 1476-4687. doi : 10.1038/319774a0. URL https://www.nature.com/articles/319774a0.

Dano J. Morrison, Asim J. Rashid, Adelaide P. Yiu, Chen Yan, Paul W. Frankland, and Sheena A. Josselyn. Parvalbumin interneurons constrain the size of the lateral amygdala engram. *Neurobiology of Learning and Memory*, 135 :91–99, November 2016. ISSN 1074-7427. doi : 10.1016/j.nlm.2016.07.007. URL https://www.sciencedirect.com/science/article/pii/S1074742716301071.

Sadegh Nabavi, Rocky Fox, Christophe D. Proulx, John Y. Lin, Roger Y. Tsien, and Roberto Malinow. Engineering a memory with LTD and LTP. *Nature*, 511 (7509) :348–352, July 2014. ISSN 1476-4687. doi : 10.1038/nature13294. URL https://www.nature.com/articles/nature13294.

Erwin Neher and Takeshi Sakaba. Multiple Roles of Calcium Ions in the Regulation of Neurotransmitter Release. *Neuron*, 59(6) :861–872, September 2008. ISSN 08966273. doi : 10.1016/j.neuron.2008.08.019. URL https://linkinghub.elsevier.com/retrieve/pii/S0896627308007423.

J. P. Neunuebel and J. J. Knierim. CA3 retrieves coherent representations from degraded input : direct evidence for CA3 pattern completion and dentate gyrus pattern separation. *Neuron*, 81(2) :416–427, 2014. doi : 10.1016/j.neuron.2013.11.017. URL https://www.cell.com/neuron/pdf/S0896-6273(13)01166-6.pdf.

H.-V. V. Ngo. Sleep spindles mediate hippocampal–neocortical coupling during long-duration ripples. *eLife*, 2020. URL https://elifesciences.org/articles/57011.pdf.

Yuval Nir and Giulio Tononi. Dreaming and the brain : from phenomenology to neurophysiology. *Trends in Cognitive Sciences*, 14(2) :88–100, February 2010. ISSN 13646613. doi : 10.1016/j.tics.2009.12.001. URL https://linkinghub.elsevier.com/retrieve/pii/S1364661309002678.

Konstantinos Pagiamtzis and Ali Sheikholeslami. Content-Addressable Memory (CAM) Circuits and Architectures : A Tutorial and Survey. *IEEE Journal of Solid-State Circuits*, 41(3) :712–727, 2006. doi : 10.1109/JSSC.2005.864128. URL https://www.pagiamtzis.com/pubs/pagiamtzis-jssc2006.pdf.

Christopher Parisien, Charles H. Anderson, and Chris Eliasmith. Solving the Problem of Negative Synaptic Weights in Cortical Models. *Neural Computation*, 20(6) :1473–1494, June 2008. ISSN 0899-7667. doi : 10.1162/neco.2008.07-06-295. URL https://doi.org/10.1162/neco.2008.07-06-295.

Virginia B. Penhune and Christopher J. Steele. Parallel contributions of cerebellar, striatal and M1 mechanisms to motor sequence learning. *Behavioural Brain Research*, 226(2) :579–591, January 2012. ISSN 1872-7549. doi : 10.1016/j.bbr.2011.09.044.

Simon Peron and Fabrizio Gabbiani. Spike frequency adaptation mediates looming stimulus selectivity in a collision-detecting neuron. *Nature Neuroscience*, 12(3) :318–326, March 2009. ISSN 1097-6256, 1546-1726. doi : 10.1038/nn.2259. URL https://www.nature.com/articles/nn.2259.

L. Personnaz, I. Guyon, and G. Dreyfus. Information storage and retrieval in spin-glass like neural networks. *Journal de Physique Lettres*, 46(8) :359–365, 1985. ISSN 0302-072X. doi : 10.1051/jphyslet:01985004608035900. URL http://www.edpsciences.org/10.1051/jphyslet:01985004608035900.

Luiz Pessoa. Understanding brain networks and brain organization. *Physics of life reviews*, 11(3) :400–435, September 2014. ISSN 1571-0645. doi : 10.1016/j.plrev.2014.03.005. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4157099/.

Huttenlocher Peter R. Synaptic density in human frontal cortex — Developmental changes and effects of aging. *Brain Research*, 163(2) :195–205, March 1979. ISSN 00068993. doi : 10.1016/0006-8993(79)90349-4. URL https://linkinghub.elsevier.com/retrieve/pii/0006899379903494.

Sasha Petrenko, Leonardo Enzo Brito da Silva, and Donald C. Wunsch. DeepART : Deep gradient-free local learning with adaptive resonance. *Neural Networks*, 190 :107580, October 2025. ISSN 0893-6080. doi : 10.1016/j.neunet.2025.107580. URL https://www.sciencedirect.com/science/article/pii/S0893608025004605.

Adrien Peyrache, Mehdi Khamassi, Karim Benchenane, Sidney I Wiener, and Francesco P Battaglia. Replay of rule-learning related neural patterns in the prefrontal cortex during sleep. *Nature Neuroscience*, 12(7) :919–926, July 2009. ISSN 1097-6256, 1546-1726. doi : 10.1038/nn.2337. URL https://www.nature.com/articles/nn.2337.

Panayiota Poirazi, Terrence Brannon, and Bartlett W. Mel. Pyramidal Neuron as Two-Layer Neural Network. *Neuron*, 37(6) :989–999, March 2003. ISSN 08966273. doi : 10.1016/S0896-6273(03)00149-1. URL https://linkinghub.elsevier.com/retrieve/pii/S0896627303001491.

Swann Proust. *Du côté de chez Swann*. 1913.

R. Quian Quiroga, L. Reddy, G. Kreiman, C. Koch, and I. Fried. Invariant visual representation by single neurons in the human brain. *Nature*, 435(7045) : 1102–1107, June 2005. ISSN 0028-0836, 1476-4687. doi : 10.1038/nature03687. URL https://www.nature.com/articles/nature03687.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, Victor Greiff, David Kreil, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. Hopfield Networks is All You Need, April 2021. URL http://arxiv.org/abs/2008.02217. arXiv :2008.02217 [cs].

Catharine H. Rankin, Thomas Abrams, Robert J. Barry, Seema Bhatnagar, David Clayton, John Colombo, Gianluca Coppola, Mark A. Geyer, David L. Glanzman, Stephen Marsland, Frances McSweeney, Donald A. Wilson,

Chun-Fang Wu, and Richard F. Thompson. Habituation Revisited : An Updated and Revised Description of the Behavioral Characteristics of Habituation. *Neurobiology of learning and memory*, 92(2) :135–138, September 2009. ISSN 1074-7427. doi : 10.1016/j.nlm.2008.09.012. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2754195/.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL : Incremental Classifier and Representation Learning. pp. 2001–2010, Honolulu, HI, 2017. IEEE. URL https://openaccess.thecvf.com/content_cvpr_2017/papers/Rebuffi_iCaRL_Incremental_Classifier_CVPR_2017_paper.pdf.

Mattia Rigotti, Omri Barak, Melissa R. Warden, Xiao-Jing Wang, Nathaniel D. Daw, Earl K. Miller, and Stefano Fusi. The importance of mixed selectivity in complex cognitive tasks. *Nature*, 497(7451) :585–590, May 2013. ISSN 0028-0836. doi : 10.1038/nature12160. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4412347/.

Anthony Robins. Catastrophic Forgetting, Rehearsal and Pseudorehearsal. *Connection Science*, 7(2) :123–146, June 1995. ISSN 0954-0091, 1360-0494. doi : 10.1080/09540099550039318. URL https://www.tandfonline.com/doi/full/10.1080/09540099550039318.

Edmund T. Rolls and Raymond P. Kesner. A computational theory of hippocampal function, and empirical tests of the theory. *Progress in Neurobiology*, 79(1) :1–48, May 2006. ISSN 03010082. doi : 10.1016/j.pneurobio.2006.04.005. URL https://linkinghub.elsevier.com/retrieve/pii/S0301008206000360.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P. Lillicrap, and Greg Wayne. Experience replay for continual learning. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, number 32, pp. 350–360. Curran Associates Inc., Red Hook, NY, USA, 2019.

Lisa Roux and György Buzsáki. Tasks for inhibitory interneurons in intact brain circuits. *Neuropharmacology*, 88 :10–23, January 2015. ISSN 0028-3908. doi : 10.1016/j.neuropharm.2014.09.011. URL https://www.sciencedirect.com/science/article/pii/S0028390814003189.

Tomás J. Ryan, Dheeraj S. Roy, Michele Pignatelli, Autumn Arons, and Susumu Tonegawa. Engram Cells Retain Memory Under Retrograde Amnesia. *Science (New York, N.Y.)*, 348(6238) :1007–1013, May 2015. ISSN 0036-8075. doi : 10.1126/science.aaa5542. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC5583719/.

Paul Saighi and Marcelo Rozenberg. Autonomous retrieval for continuous learning in associative memory networks. *Frontiers in Computational Neuroscience*, 19, August 2025. ISSN 1662-5188. doi : 10.3389/fncom.2025.1655701. URL https://www.frontiersin.org/journals/computational-neuroscience/articles/10.3389/fncom.2025.1655701/full.

Catherine D. Schuman, Thomas E. Potok, Robert M. Patton, J. Douglas Birdwell, Mark E. Dean, Garrett S. Rose, and James S. Plank. A Survey of Neuromorphic Computing and Neural Networks in Hardware, May 2017. URL http://arxiv.org/abs/1705.06963. arXiv :1705.06963 [cs].

Francesca Schönsberg, Yasser Roudi, and Alessandro Treves. Efficiency of Local Learning Rules in Threshold-Linear Associative Networks. *Physical Review Letters*, 126(1) :018301, January 2021. ISSN 0031-9007, 1079-7114. doi : 10.1103/PhysRevLett.126.018301. URL https://link.aps.org/doi/10.1103/PhysRevLett.126.018301.

W. B. Scoville and B. Milner. Loss of recent memory after bilateral hippocampal lesions. *Journal of Neurology, Neurosurgery, and Psychiatry*, 1957. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC497229/.

Jeremy K. Seamans and Charles R. Yang. The principal features and mechanisms of dopamine modulation in the prefrontal cortex. *Progress in Neurobiology*, 74(1) :1–58, September 2004. ISSN 0301-0082. doi : 10.1016/j.pneurobio.2004.05.006.

Abu Sebastian, Manuel Le Gallo, R. Khaddam-Aljameh, and Evangelos Eleftheriou. Memory devices and applications for in-memory computing. *Nature Nanotechnology*, 15 :529–544, 2020. doi : 10.1038/s41565-020-0655-z. URL https://knowen-production.s3.amazonaws.com/uploads/attachment/file/5270/10.1038_s41565-020-0655-z.pdf.

Richard Semon. *The Mneme*. 1921.

Yang Shen, Sanjoy Dasgupta, and Saket Navlakha. Reducing Catastrophic Forgetting With Associative Learning : A Lesson From Fruit Flies. *Neural Computation*, 35(11) :1797–1819, October 2023. ISSN 0899-7667. doi : 10.1162/neco_a_01615. URL https://doi.org/10.1162/neco_a_01615.

M. Sheng and E. Kim. The Postsynaptic Organization of Synapses. *Cold Spring Harbor Perspectives in Biology*, 3(12) :a005678–a005678, December 2011. ISSN 1943-0264. doi : 10.1101/cshperspect.a005678. URL http://cshperspectives.cshlp.org/lookup/doi/10.1101/cshperspect.a005678.

Hyun Geun Shim, Yong-Seok Lee, and Sang Jeong Kim. The Emerging Concept of Intrinsic Plasticity : Activity-dependent Modulation of Intrinsic Excitability in Cerebellar Purkinje Cells and Motor Learning. *Experimental Neurobiology*, 27(3) :139–154, June 2018. ISSN 1226-2560, 2093-8144. doi : 10.5607/en.2018.27.3.139. URL http://www.en-journal.org/journal/view.html?doi=10.5607/en.2018.27.3.139.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual Learning with Deep Generative Replay. *arXiv*, 2017. URL https://arxiv.org/pdf/1705.08690.

H. Francis Song, Guangyu R. Yang, and Xiao-Jing Wang. Training Excitatory-Inhibitory Recurrent Neural Networks for Cognitive Tasks : A Simple and Flexible Framework. *PLOS Computational Biology*, 12(2) : e1004792, February 2016. ISSN 1553-7358. doi : 10.1371/journal.pcbi.1004792. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1004792. Publisher : Public Library of Science.

Yang Song and Stefano Ermon. Generative Modeling by Estimating Gradients of the Data Distribution, October 2020. URL http://arxiv.org/abs/1907.05600. arXiv :1907.05600 [cs].

Larry R. Squire and Adam J.O. Dede. Conscious and Unconscious Memory Systems. *Cold Spring Harbor Perspectives in Biology*, 7(3) : a021667, March 2015. ISSN 1943-0264. doi : 10.1101/cshperspect.a021667. URL http://cshperspectives.cshlp.org/lookup/doi/10.1101/cshperspect.a021667.

M. Steriade, A. Nuñez, and F. Amzica. A novel slow (< 1 Hz) oscillation of neocortical neurons in vivo : depolarizing and hyperpolarizing components. *The Journal of Neuroscience : The Official Journal of the Society for Neuroscience*, 13(8) :3252–3265, August 1993. ISSN 0270-6474. doi : 10.1523/JNEUROSCI.13-08-03252.1993.

A.J. Storkey and R. Valabregue. The basins of attraction of a new Hopfield learning rule. *Neural Networks*, 12(6) :869–876, July 1999. ISSN 08936080. doi : 10.1016/S0893-6080(99)00038-6. URL https://linkinghub.elsevier.com/retrieve/pii/S0893608099000386.

Amos Storkey. Increasing the capacity of a hopfield network without sacrificing functionality. In Gerhard Goos, Juris Hartmanis, Jan Van Leeuwen, Wulfram Gerstner, Alain Germond, Martin Hasler, and Jean-Daniel Nicoud (eds.), *Artificial Neural Networks — ICANN'97*, volume 1327, pp. 451–456. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997. ISBN 978-3-540-63631-1 978-

3-540-69620-9. doi : 10.1007/BFb0020196. URL http://link.springer.com/10.1007/BFb0020196. Series Title : Lecture Notes in Computer Science.

David Sussillo and L. F. Abbott. Generating Coherent Patterns of Activity from Chaotic Neural Networks. *Neuron*, 63(4) :544–557, August 2009. ISSN 0896-6273. doi : 10.1016/j.neuron.2009.07.018. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC2756108/.

Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, and Joel S. Emer. Efficient Processing of Deep Neural Networks : A Tutorial and Survey. *Proceedings of the IEEE*, 105(12) :2295–2329, 2017. doi : 10.1109/JPROC.2017.2761740. URL https://dspace.mit.edu/bitstream/handle/1721.1/134932/1703.09039.pdf?isAllowed=y&sequence=2.

Timothy Tadros, Giri P. Krishnan, Ramyaa Ramyaa, and Maxim Bazhenov. Sleep-like unsupervised replay reduces catastrophic forgetting in artificial neural networks. *Nature Communications*, 13(1) :7742, December 2022. ISSN 2041-1723. doi : 10.1038/s41467-022-34938-7. URL https://www.nature.com/articles/s41467-022-34938-7.

A. Takashima, K. M. Petersson, F. Rutters, I. Tendolkar, O. Jensen, M. J. Zwarts, B. L. McNaughton, and G. Fernández. Declarative memory consolidation in humans : A prospective functional magnetic resonance imaging study. *Proceedings of the National Academy of Sciences*, 103(3) :756–761, January 2006. ISSN 0027-8424, 1091-6490. doi : 10.1073/pnas.0507774103. URL https://pnas.org/doi/full/10.1073/pnas.0507774103.

Tomonori Takeuchi, Adrian J. Duszkiewicz, and Richard G. M. Morris. The synaptic plasticity and memory hypothesis : encoding, storage and persistence. *Philosophical Transactions of the Royal Society B : Biological Sciences*, 369(1633) :20130288, January 2014. ISSN 0962-8436, 1471-2970. doi : 10.1098/rstb.2013.0288. URL https://royalsocietypublishing.org/doi/10.1098/rstb.2013.0288.

Nobuaki Tamamaki and Ryohei Tomioka. Long-Range GABAergic Connections Distributed throughout the Neocortex and their Possible Function. *Frontiers in Neuroscience*, 4, December 2010. ISSN 1662-453X. doi : 10.3389/fnins.2010.00202. URL https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2010.00202/full. Publisher : Frontiers.

Christian Tetzlaff, Christoph Kolodziejski, Marc Timme, and Florentin Wörgötter. Synaptic Scaling in Combination with Many Generic Plasticity Mechanisms Stabilizes Circuit Connectivity. *Frontiers in Computational Neuroscience*, 5, November 2011. ISSN 1662-5188. doi :

10.3389/fncom.2011.00047. URL https://www.frontiersin.org/journals/computational-neuroscience/articles/10.3389/fncom.2011.00047/full.

B. T. Thomas Yeo, Fenna M. Krienen, Jorge Sepulcre, Mert R. Sabuncu, Danial Lashkari, Marisa Hollinshead, Joshua L. Roffman, Jordan W. Smoller, Lilla Zöllei, Jonathan R. Polimeni, Bruce Fischl, Hesheng Liu, and Randy L. Buckner. The organization of the human cerebral cortex estimated by intrinsic functional connectivity. *Journal of Neurophysiology*, 106(3) :1125–1165, September 2011. ISSN 0022-3077, 1522-1598. doi : 10.1152/jn.00338.2011. URL https://www.physiology.org/doi/10.1152/jn.00338.2011.

Douglas Feitosa Tomé, Ying Zhang, Tomomi Aida, Olivia Mosto, Yifeng Lu, Mandy Chen, Sadra Sadeh, Dheeraj S. Roy, and Claudia Clopath. Dynamic and selective engrams emerge with memory consolidation. *Nature Neuroscience*, 27(3) :561–572, March 2024. ISSN 1546-1726. doi : 10.1038/s41593-023-01551-w. URL https://www.nature.com/articles/s41593-023-01551-w. Publisher : Nature Publishing Group.

Susumu Tonegawa, Xu Liu, Steve Ramirez, and Roger Redondo. Memory Engram Cells Have Come of Age. *Neuron*, 87(5) :918–931, September 2015. ISSN 08966273. doi : 10.1016/j.neuron.2015.08.002. URL https://linkinghub.elsevier.com/retrieve/pii/S0896627315006777.

Giulio Tononi and Chiara Cirelli. Sleep and the Price of Plasticity : From Synaptic and Cellular Homeostasis to Memory Consolidation and Integration. *Neuron*, 81(1) :12–34, January 2014. ISSN 08966273. doi : 10.1016/j.neuron.2013.12.025. URL https://linkinghub.elsevier.com/retrieve/pii/S0896627313011860.

Robin Tremblay, Soohyun Lee, and Bernardo Rudy. GABAergic interneurons in the neocortex : From cellular properties to circuits. *Neuron*, 91(2) :260–292, July 2016. ISSN 0896-6273. doi : 10.1016/j.neuron.2016.06.033. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC4980915/.

A. Treves and E. T. Rolls. Computational analysis of the role of the hippocampus in memory. *Hippocampus*, 4(3) :374–391, 1994. doi : 10.1002/hipo.450040319. URL https://www.oxcns.org/papers/186_Treves%2BRolls94.pdf.

G. Turrigiano. Homeostatic Synaptic Plasticity : Local and Global Mechanisms for Stabilizing Neuronal Function. *Cold Spring Harbor Perspectives in Biology*, 4(1) :a005736–a005736, January 2012. ISSN 1943-0264. doi : 10.1101/cshperspect.a005736. URL http://cshperspectives.cshlp.org/lookup/doi/10.1101/cshperspect.a005736.

Gina G. Turrigiano, Kenneth R. Leslie, Niraj S. Desai, Lana C. Rutherford, and Sacha B. Nelson. Activity-dependent scaling of quantal amplitude in neocortical neurons. *Nature*, 391(6670) :892–896, February 1998. ISSN 0028-0836, 1476-4687. doi : 10.1038/36103. URL https://www.nature.com/articles/36103.

L Ungerleider. 'What' and 'where' in the human brain. *Current Opinion in Neurobiology*, 4(2) :157–165, 1994. ISSN 09594388. doi : 10.1016/0959-4388(94)90066-3. URL https://linkinghub.elsevier.com/retrieve/pii/0959438894900663.

Gido M. Van De Ven, Tinne Tuytelaars, and Andreas S. Tolias. Three types of incremental learning. *Nature Machine Intelligence*, 4(12) :1185–1197, December 2022. ISSN 2522-5839. doi : 10.1038/s42256-022-00568-3. URL https://www.nature.com/articles/s42256-022-00568-3.

T. P. Vogels, H. Sprekeler, F. Zenke, C. Clopath, and W. Gerstner. Inhibitory Plasticity Balances Excitation and Inhibition in Sensory Pathways and Memory Networks. *Science*, 334(6062) :1569–1573, December 2011. ISSN 0036-8075, 1095-9203. doi : 10.1126/science.1211095. URL https://www.science.org/doi/10.1126/science.1211095.

Katrin Vonderschen and Maurice J. Chacron. Sparse and dense coding of natural stimuli by distinct midbrain neuron subpopulations in weakly electric fish. *Journal of Neurophysiology*, 106(6) :3102–3118, December 2011. ISSN 0022-3077. doi : 10.1152/jn.00588.2011. URL https://journals.physiology.org/doi/full/10.1152/jn.00588.2011.

Zhilin Wang, Yafu Li, Jianhao Yan, Yu Cheng, and Yue Zhang. Unveiling Attractor Cycles in Large Language Models : A Dynamical Systems View of Successive Paraphrasing, May 2025. URL http://arxiv.org/abs/2502.15208. arXiv :2502.15208 [cs].

Cody Wild and Jesper Anderson. Uncovering Layer-Dependent Activation Sparsity Patterns in ReLU Transformers, July 2024. URL http://arxiv.org/abs/2407.07848. arXiv :2407.07848 [cs].

Ben D. B. Willmore, James A. Mazer, and Jack L. Gallant. Sparse coding in striate and extrastriate visual cortex. *Journal of Neurophysiology*, 105(6) :2907–2919, June 2011. ISSN 0022-3077. doi : 10.1152/jn.00594.2010. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC3118756/.

M. A. Wilson and B. L. McNaughton. Reactivation of hippocampal ensemble memories during sleep. *Science*, 1994. URL https://www.weizmann.ac.il/brain-sciences/labs/ulanovsky/sites/

neurobiology.labs.ulanovsky/files/uploads/wilson_mcnaughton_reactivationofhippocampalensembleactivity_science_1994.pdf.

Jiaming Wu and Marcelo Rozenberg. A Trivial Implementation of an Analog Spiking Neuron Using a Memristor, for Less than $1. June 2024. ISBN 978-0-85466-167-1. doi : 10.5772/intechopen.1004909.

Gang Yang and Fei Ding. Associative memory optimized method on deep neural networks for image classification. *Information Sciences*, 533 :108–119, September 2020. ISSN 00200255. doi : 10.1016/j.ins.2020.05.038. URL https://linkinghub.elsevier.com/retrieve/pii/S0020025520304175.

Semir Zeki. A massively asynchronous, parallel brain. *Philosophical Transactions of the Royal Society B : Biological Sciences*, 370(1668) :20140174, May 2015. ISSN 0962-8436. doi : 10.1098/rstb.2014.0174. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4387515/.

Fan Zhang and Xinhong Zhang. The Average Radius of Attraction Basin of Hopfield Neural Networks. In Fuchun Sun, Jianwei Zhang, Ying Tan, Jinde Cao, and Wen Yu (eds.), *Advances in Neural Networks - ISNN 2008*, pp. 253–258, Berlin, Heidelberg, 2008. Springer. ISBN 978-3-540-87734-9. doi : 10.1007/978-3-540-87734-9_29.

X. Zou, S. Xu, and X.-M. Chen. Breaking the von Neumann bottleneck : architecture-level processing-in-memory technology. *Science China Information Sciences*, 64 :160404, 2021. doi : 10.1007/s11432-020-3227-1. URL https://scis.scichina.com/en/2021/160404.pdf.

Robert S. Zucker and Wade G. Regehr. Short-Term Synaptic Plasticity. *Annual Review of Physiology*, 64(1) :355–405, March 2002. ISSN 0066-4278, 1545-1585. doi : 10.1146/annurev.physiol.64.092501.114547. URL https://www.annualreviews.org/doi/10.1146/annurev.physiol.64.092501.114547.

H. Freyja Ólafsdóttir, Daniel Bush, and Caswell Barry. The Role of Hippocampal Replay in Memory and Planning. *Current Biology*, 28(1) :R37–R50, January 2018. ISSN 09609822. doi : 10.1016/j.cub.2017.10.073. URL https://linkinghub.elsevier.com/retrieve/pii/S0960982217314410.

Özge D. Özçete, Aditi Banerjee, and Pascal S. Kaeser. Mechanisms of neuromodulatory volume transmission. *Molecular Psychiatry*, 29(11) :3680–3693, November 2024. ISSN 1476-5578. doi : 10.1038/s41380-024-02608-3. URL https://www.nature.com/articles/s41380-024-02608-3.