frontiers | Frontiers in Neuroscience

TYPE Original Research
PUBLISHED 19 September 2024
DOI 10.3389/fnins.2024.1439414

OPEN ACCESS

EDITED BY
Timothée Masquelier,
Centre National de la Recherche Scientifique (CNRS), France

REVIEWED BY
Andreas Knoblauch,
Hochschule Albstadt-Sigmaringen, Germany
Saverio Ricci,
Polytechnic University of Milan, Italy

*CORRESPONDENCE
Pawel Herman
✉ paherman@kth.se

RECEIVED 27 May 2024
ACCEPTED 29 August 2024
PUBLISHED 19 September 2024

CITATION
Ravichandran N, Lansner A and Herman P (2024) Spiking representation learning for associative memories. *Front. Neurosci.* 18:1439414. doi: 10.3389/fnins.2024.1439414

Spiking representation learning for associative memories

Naresh Ravichandran[1], Anders Lansner[1,2] and Pawel Herman[1,3,4]*

[1]Computational Cognitive Brain Science Group, Department of Computational Science and Technology, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden, [2]Department of Mathematics, Stockholm University, Stockholm, Sweden, [3]Digital Futures, KTH Royal Institute of Technology, Stockholm, Sweden, [4]Swedish e-Science Research Centre (SeRC), Stockholm, Sweden

Networks of interconnected neurons communicating through spiking signals offer the bedrock of neural computations. Our brain's spiking neural networks have the computational capacity to achieve complex pattern recognition and cognitive functions effortlessly. However, solving real-world problems with artificial spiking neural networks (SNNs) has proved to be difficult for a variety of reasons. Crucially, scaling SNNs to large networks and processing large-scale real-world datasets have been challenging, especially when compared to their non-spiking deep learning counterparts. The critical operation that is needed of SNNs is the ability to learn distributed representations from data and use these representations for perceptual, cognitive and memory operations. In this work, we introduce a novel SNN that performs unsupervised representation learning and associative memory operations leveraging Hebbian synaptic and activity-dependent structural plasticity coupled with neuron-units modelled as Poisson spike generators with sparse firing (~1 Hz mean and ~100 Hz maximum firing rate). Crucially, the architecture of our model derives from the neocortical columnar organization and combines feedforward projections for learning hidden representations and recurrent projections for forming associative memories. We evaluated the model on properties relevant for attractor-based associative memories such as pattern completion, perceptual rivalry, distortion resistance, and prototype extraction.

KEYWORDS

spiking neural networks, associative memory, attractor dynamics, Hebbian learning, structural plasticity, BCPNN, representation learning, unsupervised learning

1 Introduction

The human brain has long captivated scientists and engineers across disciplines, serving as a wellspring of inspiration for advancements in artificial intelligence, robotics, computing paradigms, and algorithmic designs. The brain's remarkable efficiency, robustness, and parallel processing capabilities continues to act as a blueprint for developing sophisticated computing systems. Conventional computing paradigms, characterized by their sequential execution of instructions and rigid separation of memory and processing units, are increasingly being challenged by the growing demands of emerging applications such as real-time data analytics, autonomous systems, and cognitive computing. The brain seamlessly integrates memory and computation, operates in a massively parallel fashion, and exhibits remarkable fault tolerance and energy efficiency – all features that modern computing systems strive to achieve.

The potential of brain-like computing is evident in the realm of SNNs with the aim to reduce the energy cost. Notably, this energy efficiency of the brain is not attributed to a small

network size; rather, the human brain packs in billions of neurons and trillions of synapses. SNNs have been shown to efficiently process real-time data streams through sparse and asynchronous event-based communication paradigms (Marković et al., 2020; Roy et al., 2019; Schuman et al., 2022; Zenke and Neftci, 2021). However, despite their potential, SNNs currently face several limitations. Notably, they lack robust mechanisms for learning sparse distributed internal representations from real-world data, a capability essential for real-world pattern recognition tasks, as deep learning has demonstrated. Moreover, in the spirit of human-like perceptual functionality, SNNs should be able to learn these representations in an unsupervised manner and utilize it for associative memory function, a hallmark feature of neural computations in the brain. Addressing these challenges is crucial for unlocking the full potential of SNNs and their neuromorphic implementations.

In this work, with the ambition to systematically tackle the aforementioned challenges, we introduce and evaluate a novel SNN model grounded in our previous work on non-spiking brain-like computing architectures (Ravichandran et al., 2023a; Ravichandran et al., 2024, 2021, 2020; Ravichandran et al., 2023b). Our earlier work derived from the Bayesian Confidence Propagation Neural Network (BCPNN) framework (Lansner and Ekeberg, 1989) and showed capacity to learn sparse distributed representations and employ these representations for associative memory function. Here, our spiking neuron model is a stochastic Poisson spike generation process and operates at low firing rates recapitulating the characteristics of *in vivo* cortical pyramidal neurons. We have incorporated several brain-like features into our model to enhance its biological plausibility (O'Reilly, 1998; Pulvermüller et al., 2021; Ravichandran et al., 2024): (1) Hebbian plasticity: online synaptic learning leveraging only localized correlational information from pre- and post-synaptic spikes, (2) structural plasticity: an activity-dependent rewiring algorithm that learns a sparse (<10%) patchy connectivity matrix, (3) sparsely spiking activities: neuronal firing with Poisson statistics and around 1 Hz mean and 100 Hz maximum firing rate, (4) neocortical columnar architecture: functional hypercolumn modules with minicolumns competing in a soft-winner-takes-all manner, and (5) cortex-like network architecture: feedforward, recurrent, and feedback projections. Crucially, our feedforward projections are responsible for extracting sparse distributed hidden representations from data and the recurrent projections facilitate robust and reconstructive associative memory functions through attractor dynamics.

Based on the results from our previous research (Ravichandran et al., 2024, 2021, 2020; Ravichandran et al., 2023b), here in this work we have tested the following hypothesis: *the sparsely spiking Poissonian neurons integrated within our brain-like network architecture achieve the same performance as non-spiking rate-based networks in terms of learning representations and associative memory functionality*. To this effect, we designed models with feedforward-only and full architectures (*Ff* and *Full*; Figure 1A), each with three different activations (Figure 1B): rate-based (*Rate*), spiking (*Spk*; 1,000 Hz maximum firing rate), and sparsely spiking (*Spspk*; 100 Hz maximum firing rate). In effect, we have compared the following six models:

1 *RateFf*: rate-based activation in a feedforward network (without recurrent projections).
2 *RateFull*: rate-based activation in a full network (with recurrent projections).
3 *SpkFf*: spiking activation in a feedforward network.
4 *SpkFull*: spiking activation in a full network.
5 *SpspkFf*: sparsely spiking activation in a feedforward network, and
6 *SpspkFull*: sparsely spiking activation in a full network.

We have evaluated our models on the widely used MNIST handwritten digits dataset and made the following key observations: (1) the sparsely spiking model closely approximates the spiking (densely spiking) and rate-based models in terms of representation learning and associative memory function; (2) the previous published rate-based BCPNN model can be recast entirely as a sparsely spiking model with minimal modifications, and a synaptic short-term filtering (z-traces) is sufficient and necessary for this procedure; (3) the addition of recurrent projections enable the model to perform associative memory function and render it more robust compared to a feedforward-only model.

2 Related works

2.1 Models of associative memory and their limitation when learning correlated memories

The synaptic connections in the brain, especially in the neocortex, are found to be predominantly recurrent in nature (Douglas and Martin, 2007). Yet their precise role in cortical information processing remains unclear (Kar et al., 2019; Kietzmann et al., 2019; Spoerer et al., 2017; van Bergen and Kriegeskorte, 2020). One prominent hypothesis suggests that extensive recurrence facilitates associative memory, wherein distributed assemblies of coactive neurons reinforce each other (Lansner, 2009; Palm, 1980; Willshaw et al., 1969). This concept of cell assembly, known variously as associative memory (Harris, 2005; Hebb, 1949; Lansner et al., 2003), attractor (Amit, 1989; Hopfield, 1982; Khona and Fiete, 2022), ensemble (Yuste et al., 2024), avalanche (Plenz and Thiagarajan, 2007), cognit (Fuster, 2006) among others, is hypothesized to serve as the internal representations of memorized objects. Several theoretical and computational studies have shown that recurrently connected neuron-like binary units with symmetric connectivity can implement attractor dynamics: the network is guaranteed to converge to attractor states corresponding to local energy minima in analogy with statistical physics (Amit, 1989; Hopfield, 1982). Learning memories in such networks typically follows Hebbian synaptic plasticity, i.e., the synaptic connections between neurons are strengthened when they are coactive (and weakened otherwise). Subsequent work showed recurrent modular networks where each module can be in one of many possible discrete states have increased storage capacity compared to non-modular networks (Gripon and Berrou, 2011; Kanter, 1988; Knoblauch and Palm, 2020).

Associative memories functionally reflect the Gestalt nature of perception of the whole form rather than just a collection of isolated parts (Wagemans et al., 2012) and the reconstructive nature of memory discussed in psychology (Anderson et al., 1973; Bartlett and Kintsch, 1995). We describe four key functions of associative memories (Lansner, 2009; Palm, 1980; Rolls and Treves, 2012):

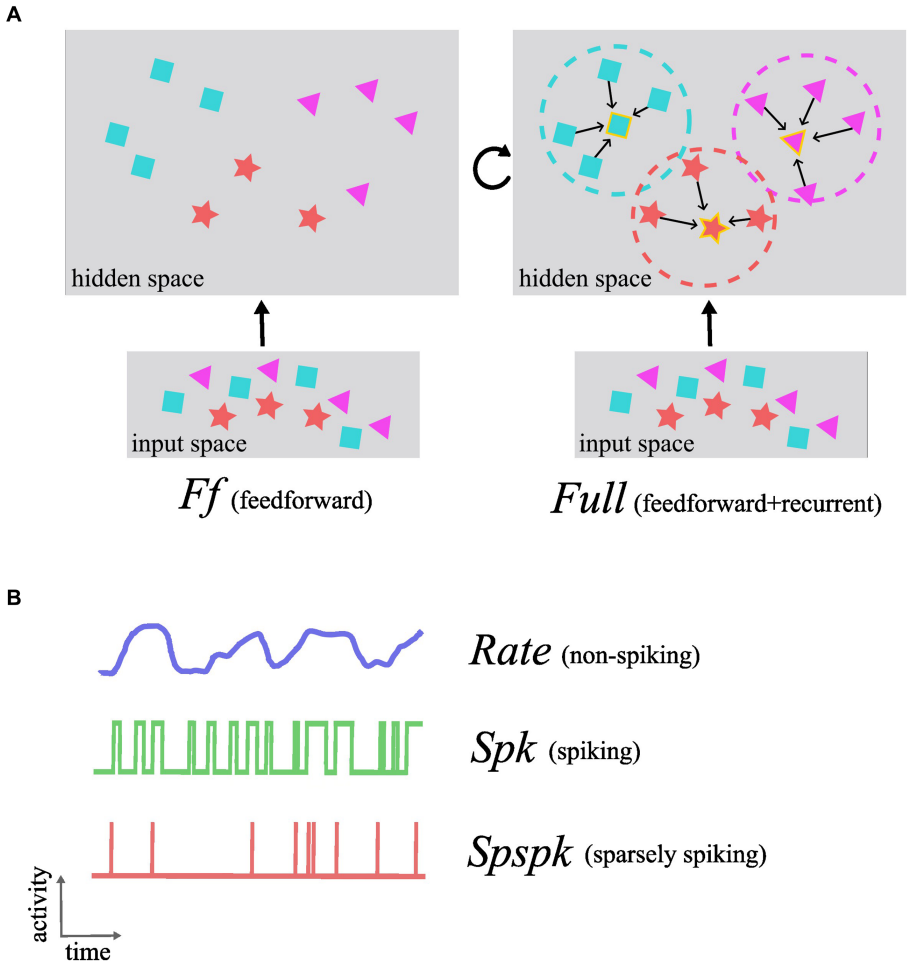

FIGURE 1
Conceptual schematic of functional roles of the different architectures and neuron-unit activation types investigated in this work. **(A)** In the feedforward-only network (Ff), the representations in the input space are highly correlated and data from distinct categories and with different features are entangled in complex non-linear relationships (shown as purple triangles, red star, and blue squares). The feedforward projections learn to map these data into the hidden space where the data points are less correlated and grouped together based on the feature similarity making them more linearly separable. The Full architecture that includes the recurrent projection utilizes the uncorrelated nature of the representations in the hidden space to form effective associative memories and group similar data points into attractors (attractor boundaries or basins of attraction shown as dashed circles and attractor states as symbol with golden border). **(B)** The activation function denoting the signal computed and communicated by each neuron-unit can be one of either Rate (non-spiking), Spk (spiking), or Spspk (sparsely spiking). The rate-based activation codes for the probability of the presence of a feature that the neuron-unit represents ("confidence") and takes continuous values in the interval [0, 1]. The Spk activation is generated as stochastic binary samples from the underlying firing rate which can reach biologically implausible levels up to 1,000 Hz. The Spspk activation is generated as stochastic binary samples, but with firing rate scaled down to biologically realistic values with the maximum of 100 Hz.

Prototype extraction relates to psychological studies on concept formation and categorical knowledge representation where concepts are stored as a set of descriptors of a prototype and novel examples are judged to be category members based on their closeness to this prototype (Rosch, 1988). The concept representations engage large-scale patterns of neural activity distributed across the neocortex and hippocampus (Fernandino et al., 2022; Handjaras et al., 2016; Kiefer and Pulvermüller, 2012).

Pattern completion involves reconstructing a complete memory pattern when cued with partial patterns. As a memory related phenomenon, it is particularly associated with the hippocampus and neocortex in humans (Horner et al., 2015; Liu et al., 2016). Interestingly, in human visual perception tasks with partially occluded objects, behavioral choices as well as neuronal dynamics show delayed responses (ca. 50–100 ms) compared to when whole objects are presented, which suggests the involvement of recurrent and top-down processing (Tang et al., 2018, 2014). Mice studies have shown that optogenetic stimulation of a specific subset of neurons belonging to an ensemble gives rise to the recall of the whole pattern (Carrillo-Reid et al., 2016) and causally trigger behavioral responses even in the absence of visual stimulation (Carrillo-Reid et al., 2019).

Pattern rivalry corresponds to scenarios where multiple conflicting pattern cues (typically two) are simultaneously presented and only one of these competing patterns "wins over" as a percept. Pattern rivalry phenomena is studied in the psychological domain such as the Necker cube and face-vase illusions where multiple distinct object representations compete for perceptual awareness (Carter et al., 2020). In binocular rivalry, two dissimilar images simultaneously presented to each eye compete for perceptual awareness (Blake and Logothetis, 2002; Lumer et al., 1998).

Distortion resistance implies the capability of the network to reconstruct the original pattern even if presented with a distorted cue, e.g., due to noise, limited viewing angle, poor contrast or illumination (Geirhos et al., 2018; Ghodrati et al., 2014; Wichmann et al., 2017).

While associative memory networks have been successful in modeling cortical dynamics and memory phenomena, they have primarily been trained on artificially generated orthogonal or random patterns (Hopfield, 1982; Rolls and Treves, 2012). Attractor networks struggle to reliably store overlapping (non-orthogonal) patterns, typical of real-world datasets, as they cause memory interference, so-called crosstalk, and lead to the emergence of spurious memories (Amit et al., 1987). This is a severe issue for considering associative memory networks as models of brain computation since the brain deals with high-dimensional sensory input with complex correlations. Consequently, attractor networks have not really been combined with high-dimensional correlated input and the problem of extracting suitable representations from real-world data has not received much attention in the context of associative memory. The associative memory systems in the brain (higher-order cortical associative areas and hippocampus, for instance) evidently use highly transformed representations. It is hypothesized that desirable neural representations in the brain are extracted from sensory input by feedforward cortical pathways (DiCarlo et al., 2012; Felleman and Van Essen, 1991; Fuster, 2006).

2.2 Representation learning algorithms and issues in transferring them to the spiking domain

The question of the nature of representations to be extracted from data has been studied under the topic of representation learning in the brain and in computational models (Bengio et al., 2013). Biological inspiration has been loosely adopted in deep neural networks (DNN) developed for pattern recognition on complex datasets, e.g., natural images, videos, audio, natural languages (LeCun et al., 2015). The success of deep learning in solving various real-world pattern recognition benchmarks has showcased the importance of learning distributed internal representations.

Compared to deep learning models SNNs still lack in their representation learning capacity. Building such SNNs has been typically addressed either by converting a (non-spiking) deep neural network model trained with gradient descent into a SNN, or by modifying supervised backprop-based gradient descent algorithms to accommodate spiking neurons (Cramer et al., 2022; Eshraghian et al., 2023; Roy et al., 2019; Wunderlich and Pehle, 2021; Zenke and Neftci, 2021). This approach has the advantage of exploiting the powerful gradient-based optimization techniques that have been developed extensively for DNNs. However, it is not straightforward to convert gradient-based backprop learning to a spiking domain, since spiking activation does not comply well with continuous differentiable activation function that backprop builds on. Several recent works have shown how the spiking activation can be smoothened into an activation function suitable for backprop and these models have demonstrated considerable success (Cramer et al., 2022). However, these methods typically carry many of the limitations of deep learning such as being predominantly supervised in their training, long training iterations, sensitivity to out-of-training noise, etc. Another critical issue with this approach is that it does not shed light on the learning in the brain and loses out on the impressive qualities that accompany a brain-like approach.

One prominent brain-like approach to learn hidden representations is to use biologically plausible spiking neuron activations and a localized form of learning rules. Early studies showed individual non-spiking neurons can develop selectivity to specific features when the hidden layer employs winner-takes-all competition (Bell and Sejnowski, 1995; Linsker, 1988; Rozell et al., 2008; Rumelhart and Zipser, 1985; Sanger, 1989). Later work incorporated spiking neurons and spike-timing-dependent plasticity (STDP) for learning and applied it to image recognition benchmarks (Diehl and Cook, 2015; Masquelier and Thorpe, 2007). The aforementioned models were restricted to hidden layers with a global winner-takes-all competition which makes each neuron learn exclusive features from the data, typically prototype clusters, and form localist coding. However, for a fully distributed spiking representation where neurons code for non-exclusive local features from the data, the hidden layer constituting multiple modules each with winner-takes-all competition was shown to learn distributed representations and perform well on machine learning benchmarks (Pfeiffer and Pfeil, 2018; Ravichandran et al., 2023c; Roy and Basu, 2017; Taherkhani et al., 2020; Tavanaei et al., 2019).

2.3 Complex network architectures integrating feedforward and recurrent projections

Neural network models can have complex architectures that combine the capacity of feedforward models to learn sparse distributed representations and employ them in a recurrent setting to form associative memory functions. Such architectures have been explored recently and benchmarked on machine learning datasets (Wyatte et al., 2012; O'Reilly et al., 2013; Tang et al., 2018, 2023; Kar et al., 2019; Kietzmann et al., 2019; Sa-Couto and Wichert, 2020; Ravichandran et al., 2023a; Sacouto and Wichert; 2023; Simas et al., 2023; Salvatori et al., 2024). O'Reilly et al. (2013) modelled a multi-layer network with feedforward, feedback, and recurrent (local inhibition) connections which were trained with a supervised error-driven learning and tested on a synthetic 3D object (CU3D-100) images dataset (O'Reilly et al., 2013; Wyatte et al., 2012). Their model showed that top-down connections can fill in missing information in partially occluded images and recurrent connections improved the robustness of the model for high levels of occlusion. Sacouto and Wichert (2023) created sparse distributed codes of images (MNIST and F-MNIST dataset) suitable for encoding into a Willshaw network with binary recurrent weights and use it storing and recalling images (Sacouto and Wichert, 2023; Sa-Couto and Wichert, 2020; Simas et al., 2023). Salvatori et al. (2021, 2024) and Tang et al. (2023) focused on combining predictive coding models aimed at associative memory tasks. The learning process is governed by a covariance-based predictive coding rule termed covPCN, which explicitly encodes the precision (covariance) matrix. Ravichandran et al. (2023b) used projections employing Hebbian-Bayesian learning with structural plasticity for feedforward and recurrent projections to create associative memories and showed recurrence improved the robustness

of the model to distortions of various kind (Ravichandran et al., 2023b). Traditional deep learning models, such as convolutional neural networks, have been augmented with recurrent connections (Hopfield network) in their penultimate layer to improve their robustness as well as to capture neural dynamics of the mammalian visual cortex and behavioral performance (Kar et al., 2019; Kar and DiCarlo, 2021; Kietzmann et al., 2019; Tang et al., 2018).

3 Model description

3.1 Architecture

The network constitutes three populations, *INP*, *HID*, and *INPRC* where the *INP* population is connected to the *HID* population with a feedforward projection to perform representation learning. The units in the *HID* population are recurrently connected to perform associative memory function, and the *INPRC* population receives feedback projection from the *HID* population for reconstructing the inputs.

Each population is modularized into hypercolumns and minicolumns following the columnar organization of the mammalian neocortex (Bastos et al., 2012; Douglas and Martin, 2004; Fransen and Lansner, 1998; Hubel and Wiesel, 1962; Mountcastle, 1997, 1957). The brain's cortical minicolumn comprises around 80–100 tightly interconnected neurons having functionally similar response properties (Buxhoeveden and Casanova, 2002; Hubel and Wiesel, 1962) and we abstract them into a single functional unit in this work. The minicolumn units (shown as white circles in Figure 2) locally compete within the hypercolumn module (shown as filled squares enclosing the white circles in Figure 2) operationally defined as the extent of local lateral inhibition. Thus, each population is composed of several identical hypercolumn modules, each of which in turn comprises many minicolumn units.

3.2 Hebbian-Bayesian learning

The learning rule changes the synaptic strength of connections using Bayesian-Hebbian synaptic plasticity. Our learning rule makes use of the local information available spatiotemporally at the synapse, making the learning mechanism and its underlying synaptic plasticity Hebbian, localized, and online. We indicate the pre- and post-synaptic population for each projection with the subscript i and j, respectively. From the pre- and post-synaptic spike trains, $s_i, s_j \in \{0,1\}$, the learning rule involves calculating a cascade of terms accumulating the short- and long-term statistics of pre-, post-, and pre-post joint spiking activity. All the spike and trace variables are time dependent (time index is dropped for notation brevity).

The z-traces compute the short-term filtered signals from the pre- and post-synaptic spikes (Equation 1). The z_i-trace is modelled after the rapid calcium influx initiated by opening of the NMDA and AMPA channels with short time-constants ($\tau_{z_i} \approx 5$–100 ms). The z_j-trace is modelled after the post-synaptic depolarization event and backpropagating action potential time-constants ($\tau_{z_j} \approx 5$–100 ms). Each spike event is scaled by a spike scaling factor, $\mu_{\text{spk}} = f_{\max}\Delta t$, where $f_{\max}$ is the hyperparameter controlling the maximal firing-rate and Δt is the timestep ($\Delta t = 1$ ms).

$$\begin{aligned} \tau_{z_i} \frac{\mathrm{d}z_i}{\mathrm{dt}} &= \frac{1}{\mu_{\text{spk}}} s_i - z_i, \\ \tau_{z_j} \frac{\mathrm{d}z_j}{\mathrm{dt}} &= \frac{1}{\mu_{\text{spk}}} s_j - z_j \end{aligned} \tag{1}$$

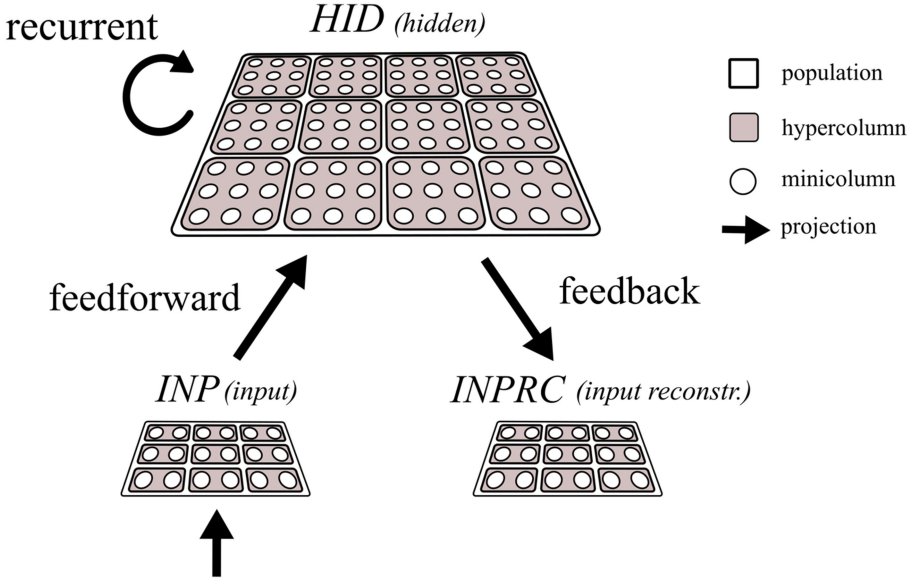

FIGURE 2
Schematic of the network architecture. The input (*INP*), hidden (*HID*), and input reconstruction (*INPRC*) populations follow the columnar/modular architecture, i.e., they are modularized into hypercolumns, which in turn constitute minicolumn units locally competing through softmax normalization. The feedforward projections connect the *INP* with the *HID* populations, recurrent projections connect the *HID* units with other *HID* units, and the feedback projections connect the *HID* population with the *INP* population.

The z -traces provide the coincidence detection window between pre- and post-synaptic spikes for subsequent plasticity induction. The z -traces are further transformed into p -traces, p_i , p_j , and p_{ij} , with long time-constants τ_p (seconds to hours) reflecting the long-term synaptic plasticity process (Equation 2).

$$\begin{aligned}\tau_p \frac{\mathrm{d}p_i}{\mathrm{dt}} &= z_i - p_i,\\ \tau_p \frac{\mathrm{d}p_{ij}}{\mathrm{dt}} &= z_i\, z_j - p_{ij},\\ \tau_p \frac{\mathrm{d}p_j}{\mathrm{dt}} &= z_j - p_j\end{aligned} \tag{2}$$

The p -traces are finally transformed to bias and weight parameters of the synapse corresponding to terms in classical artificial neural networks. The bias term represents the self-information (or surprisal, or log prior) of the post-synaptic minicolumn unit, and the weight term – the point-wise mutual information between pre- and post-synaptic minicolumn units:

$$\begin{aligned}b_j &= \log p_j,\\ w_{ij} &= \log \frac{p_{ij}}{p_i p_j}\end{aligned} \tag{3}$$

As a crucial departure from traditional backprop based DNNs, the learning rule above is local, correlative, and Hebbian, i.e., dependent only on pre- and post-synaptic activities.

Synaptic plasticity is grounded in the BCPNN framework, which integrates probabilistic inference into biologically plausible neural and synaptic operations. Our previous work with rate-based models showed that this learning rule is equivalent to the expectation maximization algorithm on a discrete mixture model where each minicolumn codes for a discrete mixture component (Ravichandran et al., 2024, 2021).

3.3 Spiking activation

The population in our network constitutes neurons producing spiking output where the firing rate reflects the confidence, i.e., probability of the presence of feature conditioned on the pre-synaptic population activity. The total synaptic input for neuron j is updated to be the weighted sum of incoming filtered spikes. The membrane voltage, v_j, is updated as the total synaptic input with a time constant τ_m as follows:

$$\tau_m \frac{\mathrm{d}v_j}{\mathrm{dt}} = b_j + \sum_{i=0}^{N_i} z_i\, w_{ij}\, c_{ij} + I_j^{\mathrm{ext}} - v_j, \tag{4}$$

where I_j^{ext} denotes the external current input and c_{ij} is the binary connection variable, $c_{ij} \in \{0,1\}$, indicating the presence of an *active* or *silent* connection (learned by the structural plasticity mechanism described in Section 3.4). The spiking probability of the neuron j , π_j , is computed as a softmax function over the membrane voltage, which induces a soft-winner-takes-all competition (lateral inhibition) between the neurons within the hypercolumn module. The output of the softmax function reflects the posterior belief probability of the minicolumn unit according to the BCPNN formalism.

$$\pi_j = \frac{\exp\left(v_j\right)}{\sum_{j'=1}^{M_j} \exp\left(v_{j'}\right)}, \tag{5}$$

In the non-spiking (rate based) BCPNN model, this activation π_j acts as the firing rate and can be directly communicated as the neuronal signal. For our spiking model, we formulate the instantaneous firing rate as the posterior belief probability (π_j) scaled by the spike scaling factor (μ_{spk}) and draw independent binary samples, s_j, with a spike probability (event with value of 1) as follows:

$$s_j \sim P\left(\text{spike between } t \text{ and } t+\Delta t\right) = \pi_j \mu_{\mathrm{spk}} \tag{6}$$

For timestep Δt smaller than the duration of changes in the underlying firing rate, the spike sampling process approximates the discrete-time version of the Poisson distribution with the underlying firing rate acting as the Poisson mean λ (Buesing et al., 2011; Ravichandran et al., 2023c). The scaling factor ($\mu_{\mathrm{spk}} = f_{\max}\Delta t$) scales the posterior belief probability to the maximum firing rate set by $f_{\max}$ (<1,000 Hz) and this renders the filtered spike statistics of the model to be equivalent to the rate-based model (Ravichandran et al., 2023c).

3.4 Structural plasticity for network rewiring

Our structural plasticity algorithm (Ravichandran et al., 2020) corresponds to the concept of structural plasticity in the brain which removes existing synaptic connections and creates new ones, thereby modifying the structure of the network in an activity- and experience-dependent manner (Bailey and Kandel, 1993; Butz et al., 2009; Holtmaat and Svoboda, 2009; Lamprecht and LeDoux, 2004; Stettler et al., 2006). Based on the current knowledge about neocortical circuits, we incorporated three key experimental findings in our algorithm: (1) the number of synaptic contacts (incoming connections) made on pyramidal (excitatory) neurons remains roughly constant throughout the neocortex, (2) neocortical connectivity is highly patchy, i.e., axons originating from pyramidal neurons branch a few times and terminate in local spatial clusters making thousands of synapses with spatial extent of the same order as a hypercolumn, and (3) many of the synaptic contacts made on the pyramidal neurons are "silent," i.e., synapses which are physical present but do not allow synaptic transmission.

Based on these observations, our rewiring algorithm computes for each *active* connection patch between every sending and receiving hypercolumn a normalized mutual information score ($\tilde{M}$). The $\tilde{M}$ score between each sending and receiving hypercolumn is defined as follows:

$$\tilde{M} = \frac{\sum_{i,j} p_{ij}\, w_{ij}}{\sum_j c_{ij}}, \tag{7}$$

where the indices i and j are minicolumn indices summing within their respective hypercolumns. The numerator is equivalent to the mutual information computed locally available at each connection

patch (w_{ij} are point wise mutual information as described in Equation 3) and the denominator is the number of outgoing connections per sending hypercolumn. For each receiving hypercolumn, if some *silent* incoming connection has greater score than some incoming *active* connection, their roles are flipped so that the *active* connection becomes *silent* and vice versa. The *silent* connections have zero weight but still act as "Hebbian probes" for statistics corresponding to the use of *silent*/*active* synapses in the (Isaac et al., 1995; Kerchner and Nicoll, 2008; Liao et al., 2001) and modeling literature (Knoblauch and Sommer, 2016; Stepanyants et al., 2002). The $\tilde{M}$ score is maximized by each receiving hypercolumn by performing flip operations where we define a flip operation as follows: converting a *silent* connection with the highest $\tilde{M}$ score into an *active* connection and converting an *active* connection with the lowest $\tilde{M}$ score to a *silent* connection. We perform $N_{\text{flip}}^{\text{conn}} = 100$ for every step of structural plasticity and we perform structural plasticity step once for every $N_{\text{intv}}^{\text{conn}} = 200$ training patterns. This way, the rewiring algorithm operates on the connectivity matrix of each projection and uses the locally available statistics on each connection patch to learn a sparse patchy connectivity.

4 Experimental setup

4.1 Core dataset

In this work, we used the MNIST hand-written digits dataset (LeCun et al., 1998), a popular image recognition benchmark dataset in the machine learning domain (accessible at http://yann.lecun.com/exdb/mnist/). MNIST consists of 60,000 training images and 10,000 test images, each with an image and the associative label denoting one of the ten classes. The MNIST images are 28×28 pixel grayscale images with one digit per image. The pixel values are grayscale intensities indicating ink stroke (1 for ink and 0 for blank space), which can be interpreted as the probability of pixel being turned on while feeding into our network. The class labels were not used for training our network model.

4.2 Test data for three associative memory tasks

We derived three distinct test image datasets from the MNIST dataset to evaluate performance in three associative memory tasks, namely pattern completion, perceptual rivalry, and distortion resistance. In each case we used the first 1,000 samples of the MNIST test dataset (for testing the model, not training). We varied the difficulty level of each task using the "difficulty level" $\in \{0.2, 0.4, 0.6, 0.8, 1\}$, and for each of the five difficulty levels, we created 1,000 patterns making it 5,000 patterns in total for each task (what "difficulty" implies for the tasks varies in each case and we describe it in detail below).

For the pattern completion task, the associative memory model was expected to recover the original memory pattern when presented with partial patterns. To simulate this, we modified the MNIST images by placing a gray bar of varying width and varying position on the image (for examples, see Figure 3A). The bar had pixel intensities of 0.5, interpreted as turning on the given pixel at a chance level (c.f. Equations 4, 5). For each difficulty level, D, the width of the bar (in pixels) was computed as follows: $\text{width} = 14 * D$ (14 is half the size of MNIST image). We chose four positions for the placement of the bars, up, down, left and, right, each amounting to 250 patterns per difficulty. For the pattern rivalry task, the associative memory model was presented with multiple conflicting patterns (typically two), and the model was expected to render one pattern to "win over" the others rival patterns. To simulate this, we modified the MNIST images by replacing a bar of varying width with pixels from another image (for examples, see Figure 3B).

For each difficulty level, D, the width of the rival image was calculated as $\text{width} = 14 * D$. We choose four positions for the placement of the bars, up, down, left and, right, each amounting to 250 patterns per difficulty. The rival images were chosen pseudo-randomly by progressing within the 250 test patterns in the reverse direction, for, e.g., the 8th image had the 242nd image as the rival (this rendered the procedure deterministic for simplicity).

For the distortion resistance task, the associative memory model was presented with patterns under various distortions and the model was supposed to restore the original, undistorted ones. To simulate this, we modified the MNIST images by performing one of five types of distortion (for examples, see Figure 3C). For each difficulty level, D, we split the 1,000 test images into 5 distortion types and created the following five distortions to the images: noise, grid, clutter, deletion and, occlusion, derived from previous work (George et al., 2017; Ravichandran et al., 2023b).

4.3 Network setup

The *INP* population constituted $H_{\text{INP}} = 784$ hypercolumns corresponding to pixels of MNIST flattened 28×28 image. Each hypercolumn was made up of $M_{\text{INP}} = 2$ minicolumns corresponding to the binary nature of the pixel intensity (ON or OFF). The *HID* population constituted $H_{\text{HID}} = 100$ hypercolumns with $M_{\text{HID}} = 100$ minicolumns per hypercolumn. The input reconstruction population, *INPRC*, had the same shape as the *INP* population, with $H_{\text{INPRC}} = 784$ and $M_{\text{INPRC}} = 2$. For all the projection types we set the parameter N_{conn} which determines the number of incoming connections per receiving hypercolumn (these connections were inherited by all minicolumns units within any given hypercolumn).

4.4 Simulation protocol

The MNIST image was injected as input into the network by setting the external current of the *INP* and *INPRC* populations. We applied the log of the image pixel intensities as the input current. Since all the populations in our network use the softmax activation function, the logged pixel intensities get transformed to the pixel intensities in the range [0, 1] (minicolumn activities). We also clipped the input current to a small positive value (1e−10), which prevents the exponential term in the softmax from becoming negative infinity (they get clipped to −10). For this, the image pixel intensity (indexed by j) is injected into the two minicolumns (indexed by $2j$ and $2j+1$) of the j-th hypercolumn as follows:

$$I_{2j}^{\text{ext}} = \log u_j, \tag{8}$$

$$I_{2j+1}^{\text{ext}} = \log\left(1 - u_j\right), \tag{9}$$

where u_j is the pixel intensity of the image normalized to be in the range [0, 1].

The network was first run in the training mode where the feedforward-driven activities are used by the network for synaptic learning and structural plasticity. This is done for each pattern by updating the p -traces (Equation 2) and computing the weights and biases on the last step of each pattern presentation. The network was then run in the evaluation mode where the test and modified test datasets (for associative memory tasks as described in Section 4.2) were run in succession without any learning.

For each pattern in the training mode the network was run in two phases, *no-input* and *ffwd* phases, and in the evaluation mode the network was run in four phases, *no-input*, *ffwd*, *overlap*, and *recr*, in succession:

1 *no-input* phase – the network is run without any input in order to clear any previous activity and avoid interference,
2 *ffwd* phase – the network is driven with the external input to the *INP* population, in turn, the *INP* population drives the *HID* population,
3 *overlap* phase – the *HID* population is driven both by the *INP* population and itself through recurrent projections, and
4 *recr* phase – the input is cutoff, and the *HID* population is running solely through recurrent self-projections.

We set the duration of each phase using parameters, $T_{\text{no-input}}$, T_{ffwd}, T_{overlap}, and T_{recr} respectively. For simulating the four phases (illustrated in Figure 4), we controlled the injection of input into the populations (Equations 8, 9) as well as the propagation of activity through each projection individually (Equation 4). Table 1 summarizes all the default parameters used in our model for the *SpspkFull* model.

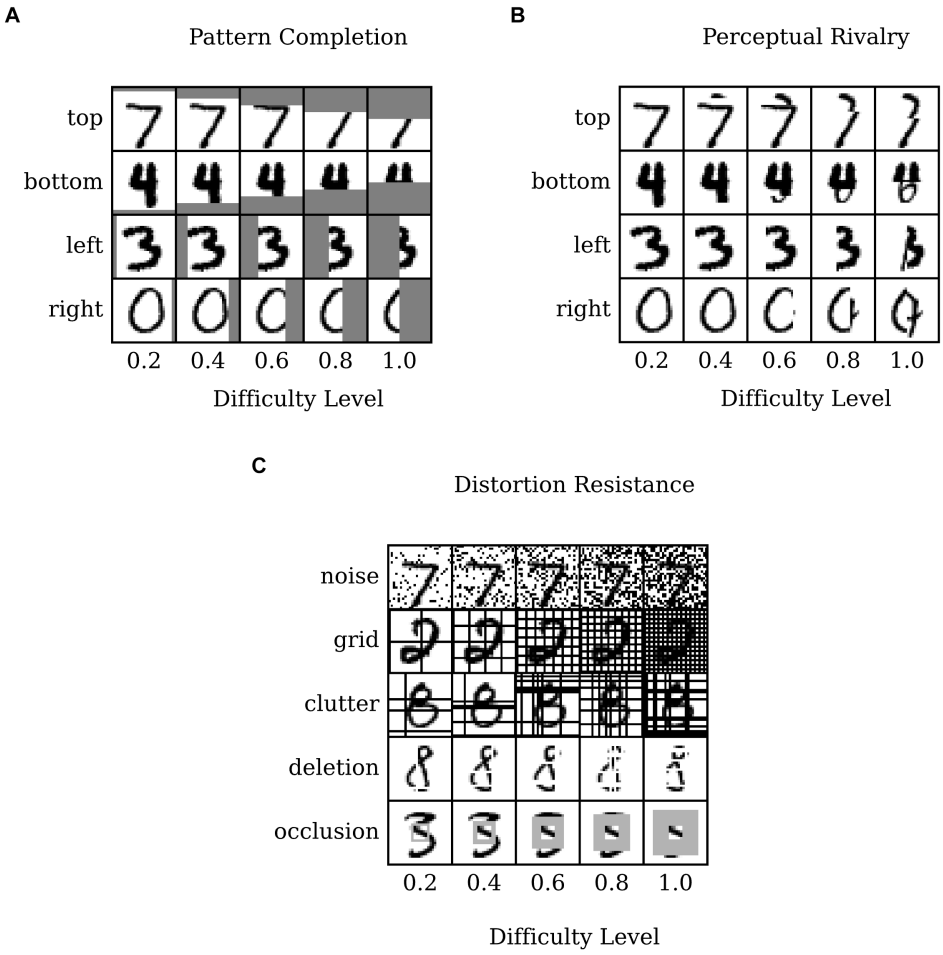

FIGURE 3
MNIST dataset modified for pattern completion, perceptual rivalry, and distortion resistance tasks sorted by difficulty level (column-wise) and type of modification (row-wise). **(A)** Pattern completion: the images are partially visible as a gray bar of varying width is placed on either the top, bottom, left, or right of image. **(B)** Perceptual rivalry: the images are partially overlapped with varying width by a randomly chosen rival image. **(C)** Distortion resistance: the images are modified by adding random flips (noise), regularly spaced grid lines (grid), randomly spaced black lines (clutter), randomly spaced white lines (deletion), or a grey square box (occlusion) with varying degree controlled by the difficulty level.

4.5 Models under comparison

We compared six different BCPNN models: (1) *RateFf*: rate-based activation in a feedforward network (without recurrent projection), (2) *RateFull*: rate-based activation in a full network (with recurrent projection), (3) *SpkFf*: spiking activation in a feedforward network, (4) *SpkFull*: spiking activation in a full network, (5) *SpspkFf*: sparsely spiking activation (with 100 Hz maximum firing rate) in a feedforward network, and (6) *SpspkFull*: sparsely spiking activation in a full network.

For the rate-based models (*RateFf* and *RateFull*), the activation was implemented by considering the softmax output value (Equation 5), π_j, directly as the neuronal signal communicated across the network. For spiking (*SpkFf* and *SpkFull*) and sparsely spiking (*SpspkFf* and *SpspkFull*) activation, we further sampled binary values (Equation 6), s_j, from the softmax output and used these for communication. Crucially, all the six models were simulated in the same code implementation by modifying the parameter values as listed in Table 2.

4.6 Evaluating representations via linear classifier

We used a linear classifier trained on the model's internal representations to decode the class labels. Although our model does not require class labels for learning, we exploited the label information to quantify the class separability as one of the evaluation methods. For this, we used a simple linear classifier with $N = 10$ softmax output units corresponding to the class labels.

We used the z-traces of the hidden population as the input to the classifier. For training the classifier, we used the cross-entropy loss function and Adam optimizer with parameters $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and, $\epsilon = 10^{-7}$ as originally defined (Kingma and Ba, 2015). We used minibatches of 64 samples and trained the network for 10 epochs.

5 Results

5.1 Sparsely firing representations show orthogonalization necessary for associative memory

Associative memory models require the patterns stored to be sparse orthogonal with minimal overlap, so that the attractor memories do not suffer interference or form spurious minima (Amit et al., 1987). We investigated with our *SpspkFull* model if the feedforward-driven activities form sparse spiking representations that have minimal overlap and benefit the formation of robust recurrent-driven associative memories.

To this end, we first visualized the spike raster and firing rate of selected hypercolumns from the *INP*, *HID*, and *INPRC* populations of the *SpspkFull* model (run on evaluation mode). The spike raster of all the populations (Figure 5; upper row) shows that the activities are mostly silent with occasional brief periods of high frequency bursts. We further computed the firing rates by convolving spike trains with a Gaussian kernel ($\sigma^2 = 20$ ms) (Figure 5; lower row). We observed that very few minicolumns produced high firing rates (typically one within a hypercolumn module) at any time due to the softmax normalization performed by the model. The peak firing rate of these units typically reaches 100 Hz confirming our model operating based on the scaling operation on the firing probability by the $f_{\max}$ parameter ($f_{\max} = 100$ Hz; Equation 6).

Next, we sought to assess the degree of orthogonalization of the representations by computing the representational similarity of all populations. For this, we computed the pair-wise cosine similarity between the z-traces for test patterns ($N_{\text{test}} = 10{,}000$) at two time points after the pattern onset: $T = 100$ ms (T_{ffwd}; feedforward-driven activities) and $T = 300$ ms ($T_{\text{ffwd}} + T_{\text{overlap}} + T_{\text{recr}}$; attractor-driven activities). For each of the similarity matrices, we computed the orthogonality ratio, s_{ortho}, that measures the ratio between average within-class similarity and average similarity across all pairs of patterns; higher value of s_{ortho} implies higher degree of orthogonalization.

The similarity matrix for *INP* population (Figure 6; left) shows the within-class similarities (class-wise diagonal values) are not very distinct from the between-class similarities (class-wise off-diagonal values), with $s_{\text{ortho}} = 1.04$, directly reflecting the nature of the MNIST dataset where images are highly correlated. Any potential associative memory directly trained on the *INP* representations would suffer from severe interference between memories owing to the high degree of overlap. The similarity matrices for *HID* representations at $T = 100$ and 300 ms (Figure 6; upper and lower center) are strikingly different from those corresponding to the *INP* representations ($s_{\text{ortho}} = 3.67$ at $T = 100$ and $s_{\text{ortho}} = 7.01$ at $T = 300$ ms), demonstrating a strong trend for the orthogonalization of *HID* representations: mostly zero similarity between-class while within-class similarity are visibly distinct and high in magnitude. The similarity matrix at $T = 300$ ms is "crisper" compared to the same at $T = 100$ ms, as expected from associative memory dynamics acting on the feedforward-driven representations. The similarity matrix for *INPRC* population at $T = 100$ and 300 ms (Figure 6; upper and lower right) shows stronger orthogonalization of the reconstructed representations ($s_{\text{ortho}} = 1.07$ at $T = 100$ ms and $s_{\text{ortho}} = 1.14$ at $T = 300$ ms) in comparison with the input representations. This demonstrated that the *HID* representations showed high degree of orthogonalization that would benefit the associative memory to store patters without considerable interference when compared to the *INP* representations.

Based on the above experiments, we concluded that the *HID* activities exhibited properties of sparse orthogonal representations that are suitable for associative memory formation. We further qualitatively examined the evolution of *INP* and *INPRC* population outputs (Figure 7) after stimulating with one example pattern for a time-period of $T = 300$ ms after the pattern onset (we skipped the no-input period). For this, we visualized the raw spiking activities (s_j) and the short-term filtered z-traces (for *INP* population we used the z_i-traces of feedforward projection and for *INPRC* population – the z_j-traces of feedback projection). The *INP* spiking activity shows highly sparse sampling of the input during the first 150 ms (feedforward and overlap phase; $T_{\text{ffwd}} = 100$ ms and $T_{\text{overlap}} = 50$ ms) and noise in the later phase, 150–300 ms (recurrent phase; $T_{\text{recr}} = 50$ ms). The *INPRC* representations reflect that there is an initial reconstruction of the presented digit (60–180 ms) corresponding to the feedforward-driven representations. After the inputs are removed (150–300 ms), the reconstruction settles to a prototypical digit, corresponding to the attractor representations in the *HID*

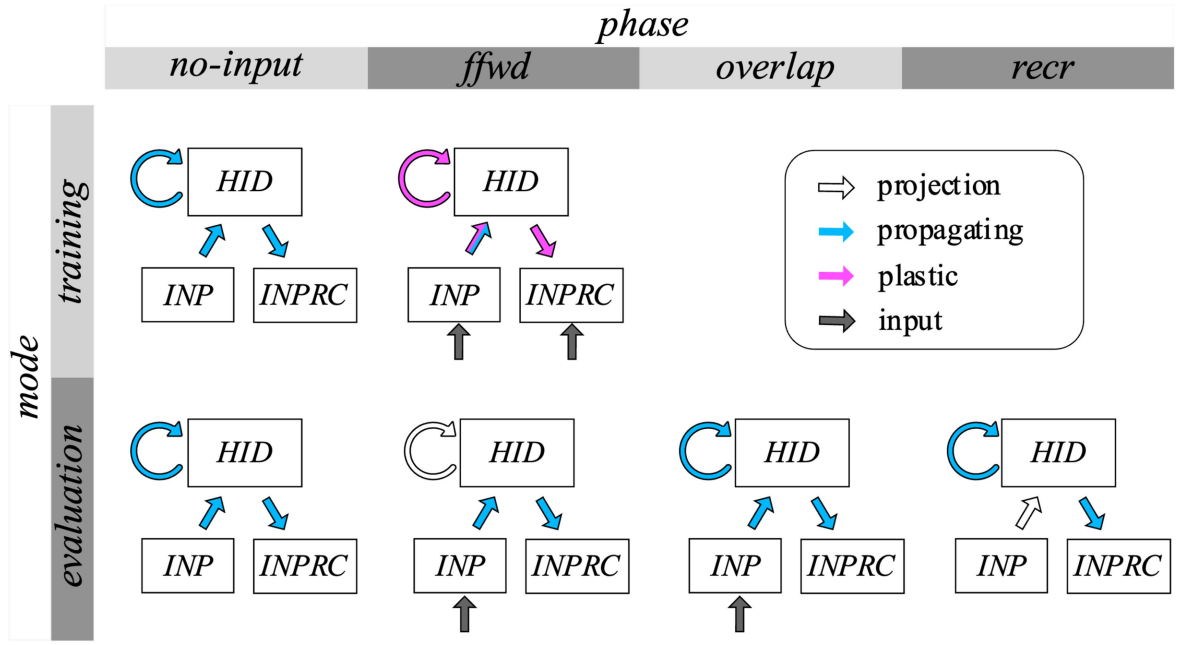

FIGURE 4
Protocol for network simulation. The network is run either in training mode (top row), where the feedforward-driven activities are used for synaptic learning and structural plasticity, or in evaluation mode (bottom row), where the feedforward-driven activities are used to cue the recurrence-driven associative memories. For each pattern, the network is run either for two phases (no-input and ffwd) in the training mode or for four phases in the evaluation mode (no-input, ffwd, overlap, and recr). The gray arrows converging on the *INP* and *INPRC* populations indicate injecting the MNIST images as inputs to the respective populations. The white unfilled arrows indicate the presence of a projection in the network connecting two populations. Blue filling on the projection arrows indicates propagation of activities through the respective projections. Purple filling on the projection arrows indicate the projection undergoes synaptic and structural plasticity updates for each pattern. Arrow with both blue and purple filling indicates the projection is both propagating and learning. In the no-input phase, the network is run without any input to clear any previous activity. In the ffwd phase the network is driven with the external input to the *INP* and *INPRC* populations and the *INP* population drives *HID* population. In the overlap phase the HID population is driven both by the *INP* population and itself through recurrent projections. In the recr phase the input is cutoff, and the HID population is running solely through recurrent self-projections.

population, which is visible as stochastic samples in the spiking activities and clearly visible from the z-traces. Based on similar experiments with many more patterns (not shown for brevity), we observed similar results where the *INPRC* representations converge to a stable attractor resembling a prototypical digit image.

5.2 Structural plasticity forms stable localized receptive fields

We evaluated the effect of the structural plasticity algorithm on the network connectivity and the formation of receptive fields over the course of training. Since the structural plasticity algorithm acts on patchy connections between hypercolumn pairs, we computed the receptive field of each hypercolumn module by using the connectivity matrix of dimensions (H_{HID} , H_{INP}) for the feedforward projections, and the transpose of the connectivity matrix of dimension (H_{INP} , H_{HID}) for the feedback projections. We plotted the receptive fields of the first ten HID hypercolumns in log steps over the course of training for the feedforward (Figure 8; left) and feedback (Figure 8; right) projections.

Each hidden hypercolumn is initialized with randomized connections with the *INP* population (Figure 8 top rows). The connections converge within the first 10,000 training patterns and remain stable over the whole course of training (bottom rows). Moreover, the connections converge to a meaningful set of spatially localized receptive fields over the input space, even though no knowledge of the input space topology was explicitly provided to the network. Furthermore, the receptive fields of the feedforward and feedback projections mirror each other for each hypercolumn module (for instance, first column of left plot and first column of right plot in Figure 8), demonstrating again that the structural plasticity finds the correlative structure between the *INP* and *HID* populations.

We observed that the number of rewiring flip operations for the feedforward and feedback projections (Figure 9; upper left and right respectively) starts off with a high value at the beginning of the training and decreases over the course of training, converging near zero. The average $\tilde{M}$ score (normalized mutual information), which is greedily maximized by each hidden hypercolumn individually, converges at a high value for both projections (Figure 9; lower left and right). It is worth noting though that the score has high variability across hypercolumns for the feedback projection. The above results demonstrated the structural plasticity algorithm identifies a set of meaningful localized receptive fields spanning the input space.

5.3 Short-term z-filtering is essential for sparsely spiking networks

The sparsely spiking models (*SpspkFf* and *SpspkFull*) were designed by scaling down the firing rates to low biologically realistic

TABLE 1 Network parameters for SpspkFull model.

Type	Parameter	Value	Description
Network architecture	H_{INP}	784	# hypercolumns in input population
	M_{INP}	2	# minicolumns per hypercolumn in input population
	H_{HID}	100	# hypercolumns in hidden population
	M_{HID}	100	# minicolumns per hypercolumn in hidden population
	H_{INPRC}	784	# hypercolumns in input reconstruction population
	M_{INPRC}	2	# minicolumns per hypercolumn in input reconstruction population
	$N_{\text{conn}}^{\text{INP}} \rightarrow \text{HID}$	78	# incoming feedforward connections per hidden hypercolumn
	$N_{\text{conn}}^{\text{HID}} \rightarrow \text{HID}$	100	# incoming recurrent connections per hidden hypercolumn
	$N_{\text{conn}}^{\text{HID}} \rightarrow \text{INPRC}$	10	# incoming feedback connections per input hypercolumn
Neural and synaptic time-constants	τ_m	0.005 s	Membrane time constant
	τ_{zi}, τ_{zj}	0.020 s	Time constant of Z-traces
	τ_p	5 s	Time constant of P-traces
Stimulation protocol	Δt	0.001 s	Simulation timestep
	T_{spk}	0.001 s	Time duration of spike
	$T_{\text{no-input}}$	0.100 s	Time duration of no-input phase
	T_{ffwd}	0.100 s	Time duration of feedforward phase
	T_{overlap}	0.050 s	Time duration of overlap phase
	T_{recr}	0.150 s	Time duration of recurrent phase
Data setup	N_{train}	60,000	Number of training patterns
	N_{test}	10,000	Number of test patterns
	N_{epoch}	20	Number of training epochs

values using the parameter $f_{\max}$ (Equation 7), for instance $f_{\max} = 100$ Hz. The crucial difference between our sparsely spiking models and other models is the use of filtering using high values for time constants τ_z (short-term z-filtering of the pre- and post-synaptic spikes) and τ_m (membrane time constant). The rate-based models (*RateFf* and *RateFull*) used $\tau_z = 1$ ms and $\tau_m = 1$ ms ($= \Delta_t$) which effectively amounts to no filtering and the spiking models (*SpkFf* and *SpkFull*) used relatively low values: $\tau_z = 5$ ms and $\tau_m = 5$ ms.

For the sparsely spiking models, we hypothesized that the effects of scaling down the spiking probability ($f_{\max} < 1{,}000$ Hz) can be countered using high values for τ_z and τ_m parameters. Furthermore, we expected the maximum firing rate $f_{\max}$ and filtering time constants (τ_z and/or τ_m) to be inversely related, i.e., lower spiking probability (lower $f_{\max}$) would be compensated by longer filtering (higher τ_z and/or τ_m) in order to recapitulate the performance of the spiking model (dense spiking; $f_{\max} = 1{,}000$ Hz). To this end, we assessed the *SpspkFull* model performance by systematically varying $f_{\max} = \{20, 50, 100, 200, 500, 1{,}000\}$ (in Hz), $\tau_m \in \{1, 2, 5, 10, 20\}$ (in ms), and $\tau_z \in \{1, 2, 5, 10, 20, 50, 100\}$ (in ms) while measuring the linear classification accuracy of each model. Since this experiment involved running many simulations ($n = 210$), we trained the models on a reduced MNIST dataset with $N_{\text{train}} = 1{,}000$ and $N_{\text{test}} = 1{,}000$, unlike the rest of the experiments which were trained and tested on the full MNIST dataset with $N_{\text{train}} = 60{,}000$ and $N_{\text{test}} = 10{,}000$.

For unrealistically high firing rates ($f_{\max} = 200$–$1{,}000$ Hz) we observed the high classification accuracy over a wide range of τ_m and τ_z since the spikes are dense samples of the underlying firing rate and filtering is not necessarily helpful (Figure 10; upper row). For biologically realistic firing rates ($\tau_z = 20$–100 Hz) performance with $\tau_z < 10$ ms is very low. This is because pre- and post-synaptic spikes are expected to coincide within this time-window for learning to occur, while the spikes are generated sparsely and irregularly from a Poisson distribution. However, for $\tau_z = 20$–50 ms the performance closely approximates the densely spiking model since this time window is sufficient to expect pre- and post-synaptic spikes to coincide and be associated through Hebbian plasticity.

All the runs irrespective of $f_{\max}$ drop sharply in performance at $\tau_z = 100$ ms, because the time window provided is too long compared to the presentation time of each pattern ($T = 300$ ms) and learning wrongly associates the current pattern with temporally adjacent patterns. We found the there was no dependence between the value of τ_m and performance across all values of $f_{\max}$. Unlike the z-traces controlling the Hebbian

TABLE 2 Parameters for the six models under comparison.

Parameter	*RateFf*	*RateFull*	*SpkFf*	*SpkFull*	*SpspkFf*	*SpspkFull*
Activity	π_j	π_j	s_j	s_j	s_j	s_j
$f_{\max}$ (Hz)	–	–	1,000	1,000	100	100
τ_{zi}, τ_{zj} (s)	0.001	0.001	0.005	0.005	0.020	0.020
τ_m (s)	0.001	0.001	0.001	0.001	0.005	0.005
$T_{\text{no-input}}$ (s)	0	0	0.025	0.025	0.100	0.100
T_{ffwd} (s)	0.005	0.005	0.025	0.025	0.100	0.100
T_{overlap} (s)	0	0	0	0.025	0	0.050
T_{recr} (s)	0	0.020	0	0.050	0	0.150

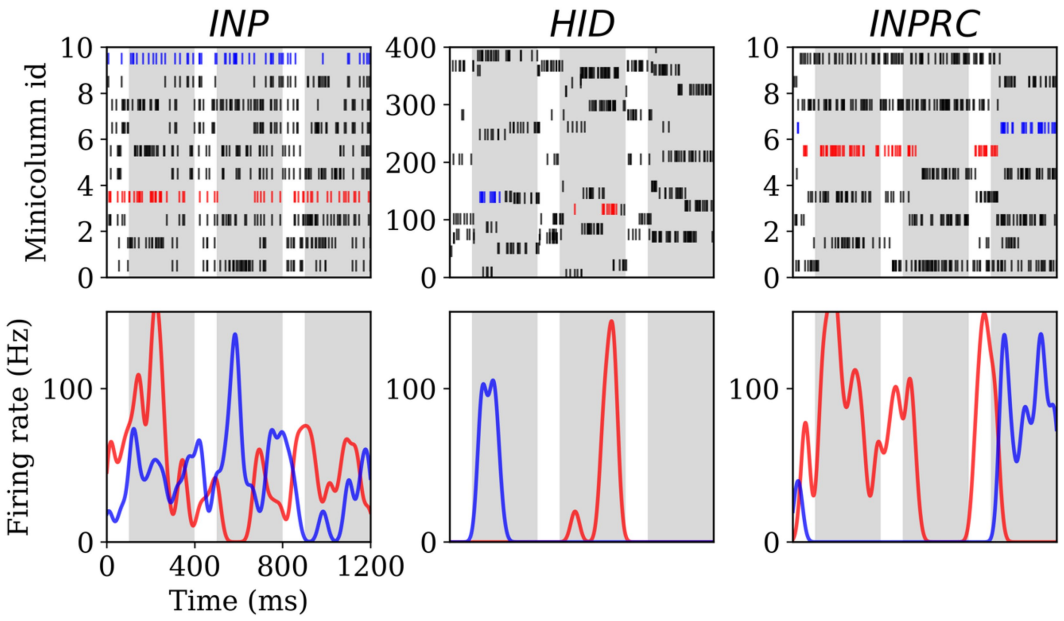

FIGURE 5
Spiking activity and firing rate for the *INP* (left), *HID* (center), and *INPRC* (right) populations. The first row shows the spike rasters for the first 10 minicolumns (arranged within 5 hypercolumns) for the *INP* and *INPRC* population and the first 400 minicolumns (arranged within 4 hypercolumns) for the *HID* population. The white bars correspond to the no-input period (100 ms) and the gray bars correspond to the pattern period (ff, overlap, and recr combined; 300 ms). The bottom row shows the firing rate of two selected minicolumns (indicated by blue and red spike trains in their respective population) computed by convolving the spike trains with a Gaussian kernel ($\sigma^2 = 20$ ms) demonstrating the minicolumns that spike with the maximum firing rate around 100 Hz.

time window of coincidence of pre- and post-synaptic spikes, the membrane time constant τ_m acts only on the post-synaptic spike and has less significance in the learning phase. The results demonstrated the importance of τ_z in the functioning of the sparsely spiking models and how longer z-filtering can compensate for the low firing rates. Based on the results, we used $\tau_z = 20$ ms for our sparse spiking models with $f_{\max} = 100$ Hz.

We finally trained all six models (parameter values are listed in Table 2) on the full MNIST dataset ($N_{\text{train}} = 60{,}000$ and $N_{\text{test}} = 10{,}000$) for $n = 5$ runs. The test accuracy results (Table 3) show that the *SpspkFf* model closely approximates the performance of *SpkFf* and *RateFf* models. Similarity, *SpspkFull* approximates *SpkFull* and *RateFull* though we notice a small decrease in performance (around 3%). In all cases, the feedforward models (*RateFf*, *SpkFf*, and *SpspkFf*) outperform the full models which feature additional recurrent projections (*RateFull*, *SpkFull*, and *SpspkFull*). This is due to the associative memory changing the feedforward-driven representations into attractor representations. Which can occasionally converge to wrong attractors. We discuss this in detail in Section 5.5 and provide scenarios where full models with recurrent projections prove to be beneficial.

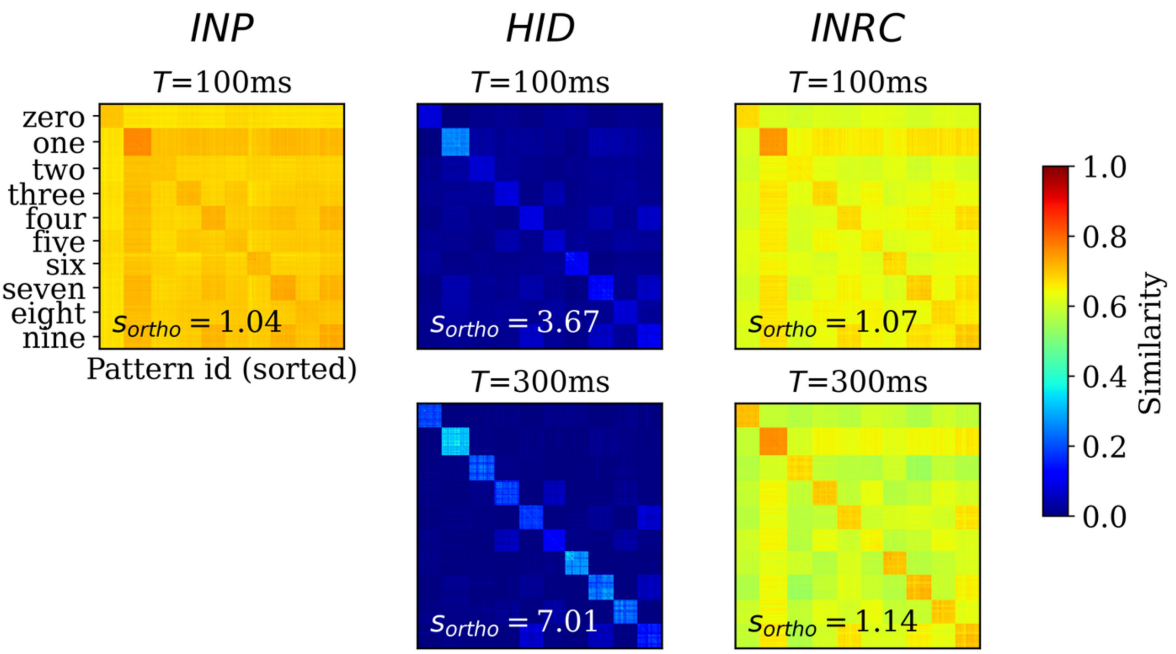

FIGURE 6
Representational similarity. Pair-wise cosine similarity matrices for N = 10,000 MNIST test patterns sorted by their labels for *INP* (left), *HID* (middle) and INPRC (right) populations at T = 100 ms (feedforward-driven; upper row) and T = 300 ms (attractor-driven; lower row) representations. The orthogonality ratio, s_{ortho}, is displayed inside each plot. The *INP* population shows low orthogonality due to the large similarity values (0.6–0.8) both within- and between-classes. The *HID* population (T = 100 and 300 ms) shows high orthogonality due to low similarity values (0–0.2) between-class owing to the sparse distributed nature of representations. The orthogonality ratio also increases from feedforward-driven representations (T = 100 ms) to attractor representations (T = 300 ms). The input reconstruction population, *INPRC*, shows more orthogonality when compared to the corresponding *INP* similarities.

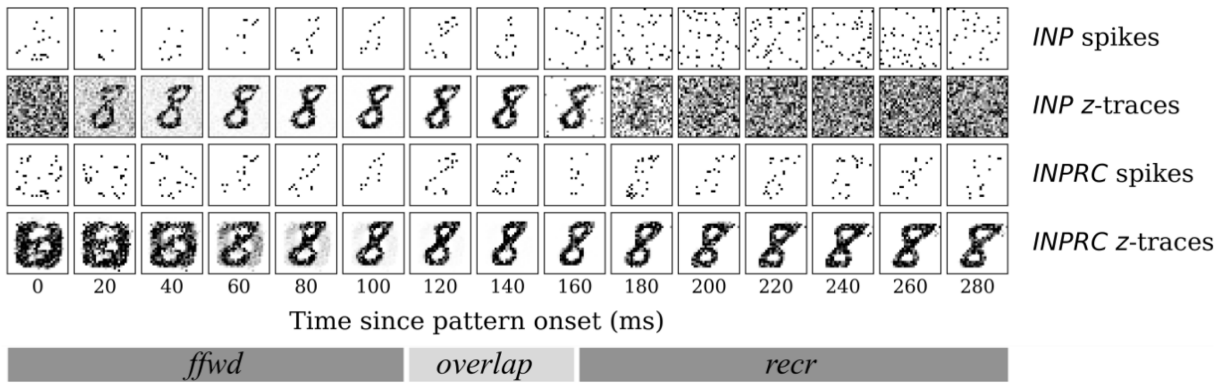

FIGURE 7
Time course of attractor representations for one pattern. The spikes and z-traces of *INP* and *INPRC* populations for one example pattern for T = 300 ms. The spike raster is highly noisy and sparse while the z-traces show a highly stable representation of digits. The INPRC population shows the initial reconstruction of feedforward-driven representations (T = 60–180 ms) and the attractor reconstructions (T = 150–300 ms) driven by the recurrent projections showing a stable convergence to the prototypical digit (corresponding to one of the class labels) even after the input is no longer fed into the network.

5.4 Prototype extraction

Associative memory involves grouping similar patterns into representative memory objects for storage. Each resulting memory object acts therefore as a representative prototype of the grouped patterns. This is analogous to clustering in machine learning. The prototypes are coded as high-dimensional distributed representations converging to attractor states (Lansner et al., 2023; Ravichandran et al., 2023b).

From the *SpspkFull* model, we expected the recurrent projections to implement prototype extraction and group similar feedforward-driven activities into common attractors. Since the activities in the HID population are sparse stochastic spikes, we used the z -traces to measure the pair-wise similarities across all the MNIST test dataset.

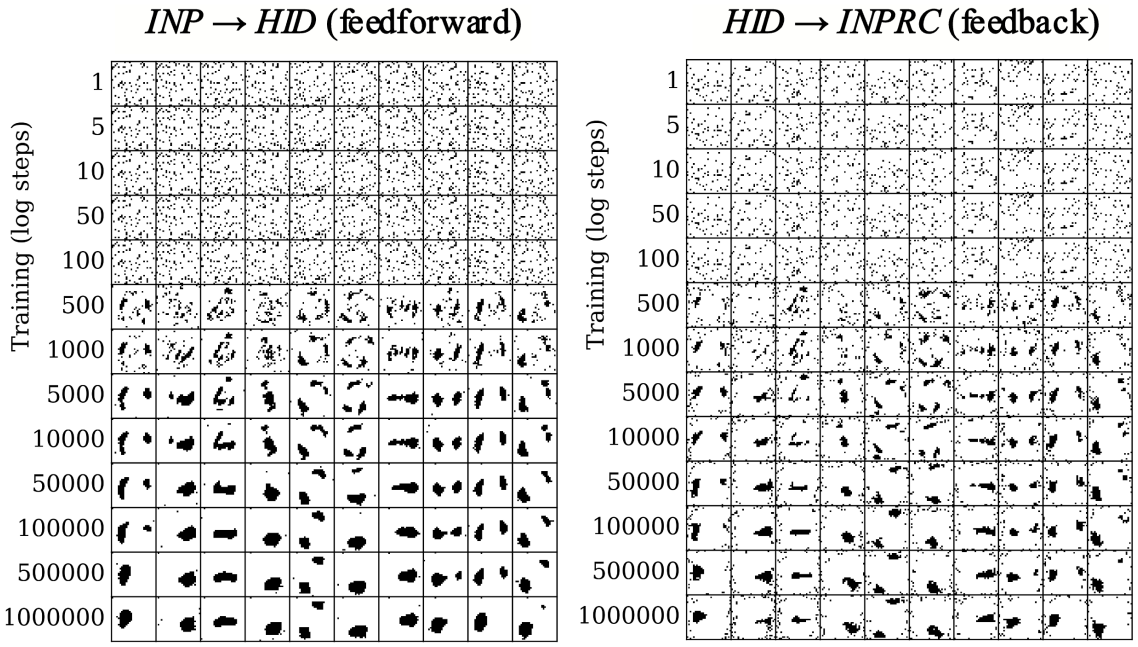

FIGURE 8
Receptive field formation for feedforward (left) and feedback (right) projections. Each column corresponds to connections between one randomly chosen hypercolumn of the *HID* population (column number corresponds to index of *HID* hypercolumn) and the *INP* population. Over the course of training the connections form spatially localized receptive fields in the image space.

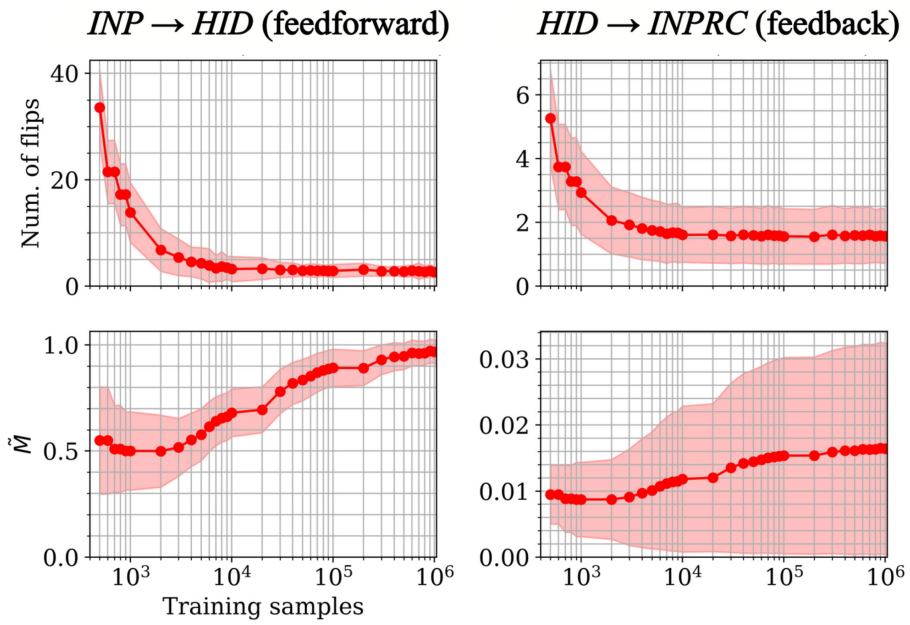

FIGURE 9
Convergence of the structural plasticity algorithm for feedforward (left) and feedback (right) projections. The number of rewiring flip operations (top row) and $\tilde{M}$ score (normalized mutual information; bottom row) per rewiring step over the course of training (mean ± std. from $n = 100$ *HID* hypercolumns) shows convergence for both feedforward and feedback projections.

For this we used the z -traces of the HID population at the last step of each pattern run ($T = 300$ ms) and computed the cosine similarity (denoted by S) removing the diagonal elements. The distribution of the cosine similarity was heavily skewed towards zero (due to the highly sparse nature of the representations) with a small fraction having a large positive value (above 0.1, for instance; Figure 11A). We also observed there were no two attractor patterns in the HID population that converged on the same unique attractor, as seen by the

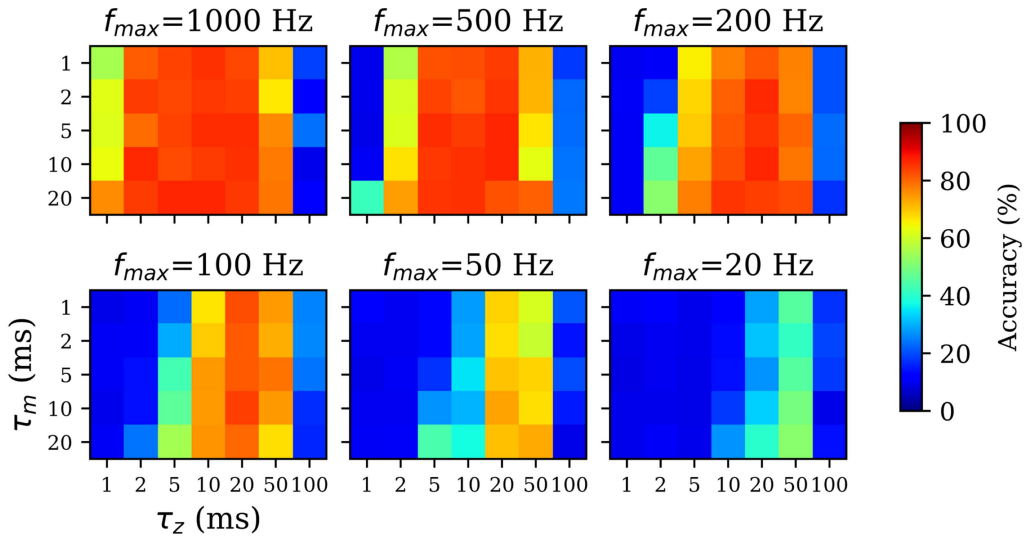

FIGURE 10
Impact of z-filtering on classification performance. Higher value of τ_z and τ_m implies longer filtering and setting the value to 1 ms (= Δt) implies essentially no filtering. Longer z-filtering compensates for low firing rates in sparsely firing spiking networks. For high firing rate (f_{max} > 200 Hz; top rows), the accuracy is high over a wide range of τ_z and τ_m values. For sparsely firing networks with biologically realistic firing rates (f_{max} < 200 Hz; bottom rows), the performance is sensitive to τ_z, optimal in range of 10–50 ms, and resistant to τ_m values.

TABLE 3 Test classification performance of all six models on MNIST test dataset (mean ± std. % from $n = 5$ runs).

	RateFf	*RateFull*	*SpkFf*	*SpkFull*	*SpspkFf*	*SpspkFull*
Test acc. (%)	98.06 ± 0.07	95.59 ± 0.14	97.93 ± 0.08	95.02 ± 0.10	97.2 ± 0.08	92.38 ± 0.17

zero count for $S = 1$. Due to this, we used a threshold value on the cosine similarity, denoted by $S_{\min}$, and considered all attractors with similarity above $S_{\min}$ as unique prototypes. Since the number of such prototypes found by our method depends on the value of $S_{\min}$, we varied $S_{\min}$ from 0 to 1 at a regular interval of 0.01 and established the relationship between number of prototypes found and $S_{\min}$ (Figure 11B). We observed that for small values, $S_{\min} < 0.25$, there were fewer number of prototypes found (less than 100). For larger values, $S_{\min} = 0.75$, the number of prototypes quicky approached the number of total test patterns ($N_{\text{test}} = 1{,}000$), since there were very few attractor patterns that had a cosine similarity $S > 0.75$ (Figure 11A) and almost all attractor patterns were categorized as unique attractors.

We selected three values of similarity threshold, $S_{\min} = 0.01$, 0.1, and 0.2, and examined the prototypes found by the model based on the input reconstructions from the INPRC population (Figure 11C). Since this involved many patterns from the test dataset converging on the same prototype, we averaged all the INPRC z-traces (from the last time step, $T = 300$ ms) that were categorized as the same prototype based on the similarity threshold. Hence, we found the average input reconstruction per prototype as well as a "attraction" index that indicates how many patterns converged on the same prototype (displayed as small text at the top of each image in Figure 11C). For $S_{\min} = 0.01$, there were 13 prototypes found that resembled most of the unique digits from the dataset (Figure 11C; upper row). For $S_{\min} = 0.1$, there were 33 prototypes found that covered all the unique digits from the dataset, as well as capturing different styles of writing the same digit, for instance, upright and slanted one (Figure 11C; middle row). For $S_{\min} = 0.2$, there were 65 prototypes found that similarly captured most of the writing styles of digits. We found a highly skewed distribution of the attraction index with only a few prototypes being popular and the majority of prototypes having a small attraction index. We also observed that the input reconstruction of the prototypes is highly stable, i.e., the average input reconstruction of a large number of attractors lead to crisp images. For instance, for $S_{\min} = 0.1$, for the three most popular prototypes the average input reconstructions of around 600 patterns clearly resembled digits "1," "8," and "4." Considering that the cosine similarity in the HID attractor space was a small value of $S_{\min} = 0.1$, the image reconstructions were highly similar to each other in the input space.

5.5 Associative memory improves the robustness of representations

The aim of the following experiments was to compare the capacity of our models on three associative memory tasks – (1) pattern completion, (2) perceptual rivalry, and (3) distortion resistance. Considering that a significant fraction of the image is corrupted, each task is considerably harder than the (clean) MNIST test dataset. We tested if the full network with recurrent projections can produce robust representations by removing the corruptions introduced. Given the highly sparse irregular firing dynamics of the *SpspkFull* model,

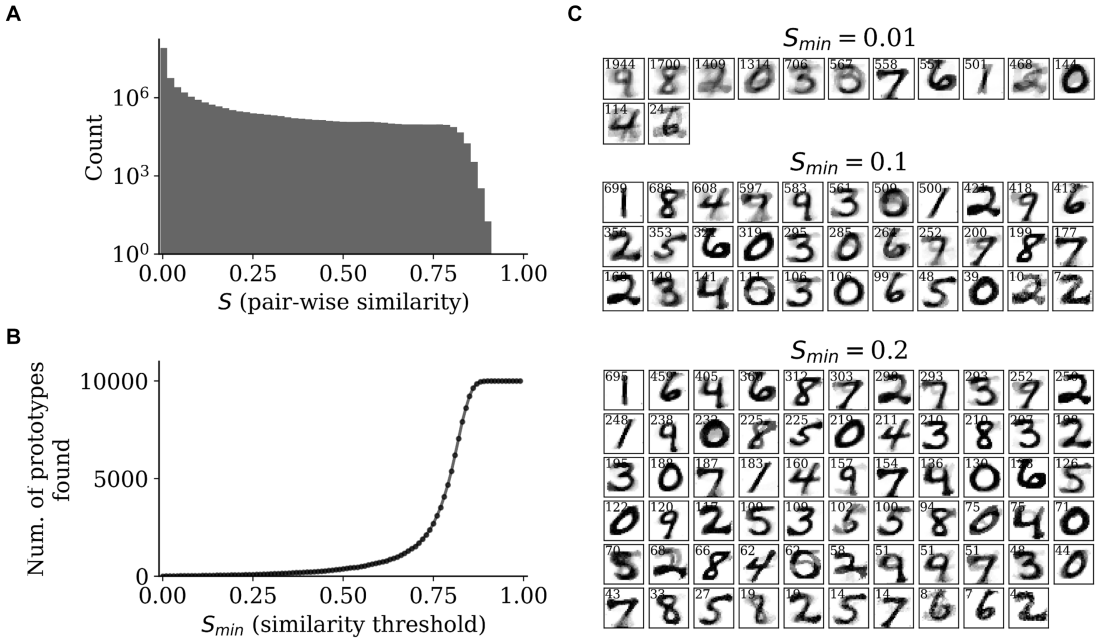

FIGURE 11
Prototype extraction. **(A)** Distribution of pair-wise similarities (*S*) of *HID* attractor representations ($T = 300$ ms) with a clear mode close to zero and only a smaller fraction of high values. **(B)** The relationship between similarity threshold, S_{min}, and the number of prototypes found by grouping attractor representations into unique prototypes. **(C)** The prototypes found for $S_{min} = 0.01$ (upper), 0.1 (middle), and 0.2 (right) by averaging the input reconstructions from the *z*-traces of the *INPRC* population.

we sought to examine if the network can handle associative memory tasks and perform competitively with the *SpkFull* and *RateFull* model. We qualitatively assessed the evolution of INP and INPRC population outputs of the *SpspkFull* model after stimulating with one example pattern from each task (difficulty level = 0.6). For this, we visualized (Figure 12) the raw spiking activities (s_j) and the short-term filtered z -traces for the INP and INPRC populations from the network at regular intervals of 20 ms after the pattern onset. For the INP population, we used the z_i -traces of feedforward projection and for the INPRC population, we used the z_j -traces of feedback projection.

For the pattern completion task (Figure 12A), the input image was covered with a gray bar on the top covering around 8 pixels, which can be seen in the filtered INP z -traces (2nd row in Figure 12A). From the feedforward-driven representations coming from the partial input, the associative memory model needs to recover the original memory pattern. In the initial period with the feedforward-driven representations (0–100 ms), the reconstructed images show corrupted content with traces of the top bar (INPRC z -traces; 60–120 ms). However, in the later phase driven exclusively through recurrent associative memory (150–300 ms), the reconstructed images reflect the convergence to a cleaned version of the corresponding digit completing the pattern (INPRC z -traces; 150–300 ms). The attractor-driven image reconstructions appear closer to the prototypical digit compared to the feedforward-driven image reconstructions.

The visualizations obtained in the perceptual rivalry (Figure 12B) and distortion resistance (Figure 12C) tasks showed similar results. The perceptual rivalry task involved presenting the network with an image combined in a smaller fraction with another rival image, as can be seen from the filtered INP z -traces (2nd row in Figure 12B). The associative memory needs to converge to the original image representation (pattern with the strongest activation) and "win-over" the rival image. The feedforward-driven reconstructions (INPRC z -traces; 60–120 ms in Figure 12B) show faithful reconstructions of the image and the rival image. However, the attractor-driven reconstructions (INPRC z -traces; 150–300 ms in Figure 12B) show that the original digit is completely recovered, i.e., the convergence to the prototypical digit without traces of the rival image.

The distortion resistance task involved presenting the network with images corrupted with various distortions and the associative memory network needs to remove the distortions and recover the original pattern. The feedforward-driven reconstructions (INPRC z -traces; 60–120 ms in Figure 12C) illustrate a highly distorted reconstruction image corrupted by the input noise. However, the attractor-driven reconstructions (INPRC z -traces; 150–300 ms in Figure 12C) reflect the convergence to the prototypical digit without any noise from the input.

Next, we sought to quantify the performance of the network using the linear classification performance. For this, we tested the six models (*RateFf*, *RateFull*, *SpkFf*, *SpkFull*, *SpspkFf*, and *SpspkFull*) on the three associative memory tasks (pattern completion, perceptual rivalry, and distortion resistance), each on 5 difficulty levels ($N = 1{,}000$ samples per difficulty level). The performance comparison for the models ($n = 5$ runs) is shown in Figure 13. We observed two main results from the experiment: (1) the sparsely spiking models (*SpspkFf*, *SpspkFull*) perform very closely to their corresponding rate (*RateFf*, *RateFull*) and spiking (*SpkFf*, *SpkFull*) models on all the associative memory tasks and on all difficulty levels. Given the highly sparse irregular spiking activity of the models, the model performance robustly recapitulates the functionality of the rate-based model. (2) The full network models (*RateFull*, *SpkFull*, and *SpspkFull*)

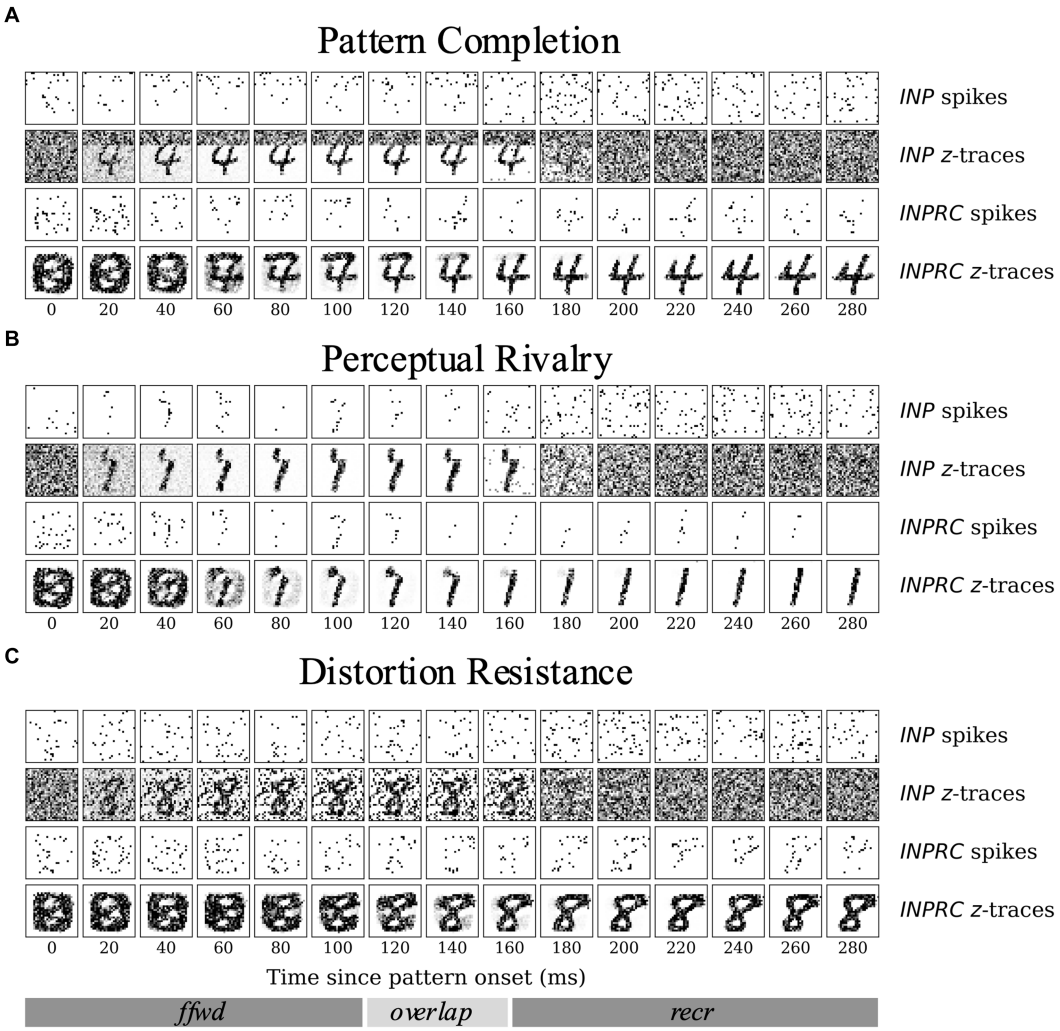

FIGURE 12
Time course of attractor representations in the (A) pattern completion, (B) perceptual rivalry, and (C) distortion resistance tasks. The spikes and z-traces of *INP* and *INPRC* populations for one example pattern are shown for each task (T = 300 ms; similar to the setup in Figure 6). The *INP* population is driven by the spiking inputs from the corrupted image (T = 0–150 ms) with (A) top gray bar, (B) occluded partially by another rival image, and (C) randomly occurring black noise. The *INPRC* population shows the reconstructed image in the feedforward-driven phase (*INPRC* z-traces; T = 60–120 ms) is the similar to the corrupted image with traces of the top bar. In the recurrent-driven phase (T = 180–300 ms) the reconstructed image is a cleaned version of the pattern and settles on a prototypical digit representation stored in the associative memory.

demonstrate a tendency for performance improvement at the high levels of task difficulty (>0.4) when compared to the feedforward-only models (*RateFf*, *SpkFf*, and *SpspkFf*) models. The associative memory function of the recurrent projections provides robustness to feedforward-driven representations, and this becomes clearly more beneficial in difficult settings when challenging out-of-training-set image samples are presented. This effect was quite pronounced in the pattern completion task and moderately so in the perceptual rivalry and distortion resistance tasks.

6 Discussion

We introduced a novel multi-population SNN model with cortex-like modular architecture based on a stochastic Poissonian spike generation process that acts in synergy with brain-like learning and structural rewiring. We systematically evaluated and compared different variants of the basic model and showed that the sparsely spiking full (*SpspkFull*) model recapitulates many of the functionalities of the non-spiking rate-based models and demonstrated the advantages of recurrent associative memory models over feedforward-only models. Crucially, all six models were simulated within the same BCPNN implementation by modifying the parameters listed in Table 2. Hence, moving from rate-based to spiking and sparsely spiking networks needs only minor changes in the network parameters, suggesting a continuum from abstract non-spiking to more detailed spiking variants for the brain-like modeling framework presented here. Our discrete-time analog of Poisson spike generation mechanism is arguably simpler than leaky-integrate-and-fire (LIF) models, but it still recapitulates the *in vivo* irregular cortical pyramidal spiking patterns with realistic firing rates. We position our spiking neuron model as an intermediary, bridging the gap between artificial neural networks with simplistic neuron-like units (e.g., rectified linear units or sigmoidal activation functions)

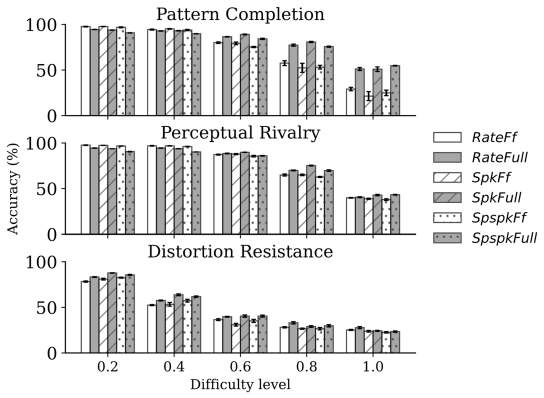

FIGURE 13
Comparison of classification performance on associative memory tasks. The sparsely spiking models (*SpspkFf, SpspkFull*) perform very closely to the rate (*RateFf, RateFull*) and spiking (*SpkFf, SpkFull*) models in all cases. For low difficulty levels (<0.4), the full network models (*RateFull, SpkFull, and SpspkFull*) do not offer a clear advantage (sometimes performance slightly worse) compared to their corresponding feedforward-only models (*RateFf, SpkFf, and SpspkFf*) models. However, there is a clear trend of improvement for the full models compared to the feedforward-only models once the difficulty level is above 0.4 in all associative memory tasks. The error bars are standard deviation from n = 5 runs.

and biologically motivated spiking neuron models (e.g., LIF, Hodgkin-Huxley neurons). Building an analogous network with LIF neurons is a logical next step and there are strong indications that the model should perform similarly. For example, in some of our previous work focused on modeling memory function using an analogous modular recurrent network with columnar architecture and excitatory-inhibitory neuron populations we demonstrated that the spiking statistics follow Poisson distribution (Lundqvist et al., 2010). Furthermore, the Hebbian-Bayesian plasticity rule employed in our model is identical to that used in earlier SNN memory models with LIF neurons (Chrysanthidis et al., 2022; Fiebig and Lansner, 2017). This consistency is achieved because the z-traces convert spikes, whether originating from Poisson neurons or LIF neurons, into temporally averaged traces, which are subsequently used for p-traces, weights, and biases.

Furthermore, the activities in the *SpspkFull* model are highly sparse, with only around 10 spikes generated at any point of time from the HID population with 10,000 minicolumn units. We posit that the Hebbian nature of our synaptic and structural plasticity algorithms can tolerate such highly irregular sparse spiking activations, which is in stark contrast to backprop-based learning algorithms that may not be best suited to accommodate spiking neurons.

Given that the spike generation process is a stochastic sampling of the underlying probabilistic activation, we do not expect any special advantage from using spiking signals and we did not observe any such improvements in performance from our experiments. This is contrast to many SNN studies where individual spiking timing and inter-spike intervals are proposed to provide additional information content in the neural coding signal (Eshraghian et al., 2023; Wunderlich and Pehle, 2021). However, it is still unclear if neocortical neurons communicate by means of such precisely timed spiking signals (Shadlen and Newsome, 1998; Softky and Koch, 1993). One disadvantage of our approach is, however, that the sparsely spiking neurons require long running times per pattern stimulation (5–50x) to reproduce the performance of their rate-based counterparts. This could possibly be mitigated by scaling up the network so that the number of incoming synapses per neuron approximates that of mammalian cortical pyramidal neurons (around thousand to tens of thousand) (DeFelipe and Fariñas, 1992). Integrating over such large number of stochastic sparsely spiking pre-synaptic inputs would provide a more robust summed synaptic input signal and lead to faster convergence during learning. We could also expect the response time of the neurons after pattern onset to be made shorter with such biologically realistic network scale, in agreement with fast response latency of first spikes observed *in vivo* in cortical visual hierarchy (Thorpe et al., 1996). The large time constants for z -traces (20–50 ms), which we showed to be necessary for networks with low firing rate (Section 5.3), could also be relaxed to shorter time constants with such large-scale networks.

We expect our network to be extendable to more commonly used spiking neuron models (such as LIF) without compromising performance. Previous modeling studies showed that local excitatory-inhibitory circuits with LIF neurons produces Poissonian statistics and reproduce well many of the *in vivo* cortical neuron spiking dynamics including oscillations and synchronization effects (Brunel, 2000; Lundqvist et al., 2010; Rullán Buxó and Pillow, 2020; Van Vreeswijk and Sompolinsky, 1996). These results strongly suggest that our spike generation process can be reproduced in more biologically detailed neuron models and integrated with other biophysical mechanisms such as spike-frequency adaptation, synaptic facilitation/depression, realistic post-synaptic potentials, axonal and dendritic delays, etc. The network can also be extended into more complex architectures, most significantly, into multilayer ones with, for instance, hierarchical feature extraction. Also, inclusion of cortical laminar organization with L4, L2/3, and L5/6 layers can allow for continuous integration of feedforward, recurrent and feedback connections reminicent of the corresponding cortical functional architecture. We used different operational phases (Section 4.4) to switch between feedforward and recurrent projections in our network to avoid one projection dominating the other. Presumably, this could be solved in a biologically plausible manner with separate laminar layers with distinct neural populations that are tightly coupled within each minicolumn, as modeled by the cortical microcircuit architecture (Douglas and Martin, 2004). Another biological mechanism could be neuromodulation as a global gating signal for synapses corresponding to specific projections.

More extensive comparison with other SNNs (trained with surrogate gradients or EventProp, for instance) and brain-inspired models will be needed to test the capacity of our model against other machine learning models. Also, traditional feedforward-driven deep learning models have been showed to severely deteriorate in performance when tested on untrained noise distortions and diverge from human behavioural performance (Bowers et al., 2022; George et al., 2017; Tang et al., 2018). Further comparisons of our model with deep learning models (such as convolutional neural networks) on noise robustness and associative memory tasks can elucidate the difference between the models.

Our work represents a step towards an integration of biologically plausible spiking models with complex brain architectures and offers exciting opportunities for scalable brain-like algorithms and multi-network models. We believe this offers

high potential for the next-generation neuromorphic algorithms and hardware systems.

Data availability statement

Publicly available datasets were analyzed in this study. The MNIST data can be found here: https://yann.lecun.com/exdb/mnist/. The code used to run the experiments is available publicly on GitHub and can be accessed here: https://github.com/nbrav/BCPNNSim-Frontiers2024.

Author contributions

NR: Conceptualization, Data curation, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. AL: Conceptualization, Data curation, Methodology, Software, Supervision, Writing – review & editing. PH: Conceptualization, Funding acquisition, Methodology, Project administration, Resources, Supervision, Writing – review & editing.

Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. Funding for the work is received from the Swedish e-Science Research Centre (SeRC), Digital Futures, Swedish Research Council 2018-05360, Vetenskapsrådet (VR2016-05871 and VR2018-05360), the European Commission Directorate-General for Communications Networks, Content and Technology grant no. 101135809 (EXTRA-BRAIN).

Acknowledgments

The computations were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) at the PDC Center for High Performance Computing, KTH Royal Institute of Technology, partially funded by the Swedish Research Council through grant agreement no. 2018-05973. We would like to thank Swedish e-Science Research Centre (SeRC), Digital Futures, Vetenskapsrådet, the EU Horizon progam for generous funding of the project.

Conflict of interest

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

Publisher's note

References

Amit, D. J. (1989). Modeling brain function. The World of Attractor Neural Networks, Cambridge University Press: Cambridge University Press.

Amit, D. J., Gutfreund, H., and Sompolinsky, H. (1987). Information storage in neural networks with low levels of activity. *Phys Rev A (Coll Park)* 35:2293. doi: 10.1103/PhysRevA.35.2293

Anderson, J.R., John, R., and Bower, G.H., (1973). Human associative memory: A brief edition

Bailey, C. H., and Kandel, E. R. (1993). Structural changes accompanying memory formation. *Annu. Rev. Physiol.* 55, 397–426. doi: 10.1146/annurev.ph.55.030193.002145

Bartlett, F.C., and Kintsch, W., (1995). Remembering: A study in experimental and social psychology

Bastos, A. M., Usrey, W. M., Adams, R. A., Mangun, G. R., Fries, P., and Friston, K. J. (2012). Canonical microcircuits for predictive coding. *Neuron* 76, 695–711. doi: 10.1016/J.NEURON.2012.10.038

Bell, A. J., and Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Comput.* 7, 1129–1159. doi: 10.1162/neco.1995.7.6.1129

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: a review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 1798–1828. doi: 10.1109/TPAMI.2013.50

Blake, R., and Logothetis, N. K. (2002). Visual competition. *Nat. Rev. Neurosci.* 3, 13–21. doi: 10.1038/nrn701

Bowers, J. S., Malhotra, G., Dujmović, M., Montero, M. L., Tsvetkov, C., Biscione, V., et al. (2022). Deep problems with neural network models of human vision. *Behav. Brain Sci.* 46:2813. doi: 10.1017/S0140525X22002813

Brunel, N. (2000). Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. *J. Comput. Neurosci.* 8, 183–208. doi: 10.1023/A:1008925309027/METRICS

Buesing, L., Bill, J., Nessler, B., and Maass, W. (2011). Neural dynamics as sampling: a model for stochastic computation in recurrent networks of spiking neurons. *PLoS Comput. Biol.* 7:e1002211. doi: 10.1371/JOURNAL.PCBI.1002211

Butz, M., Wörgötter, F., and van Ooyen, A. (2009). Activity-dependent structural plasticity. *Brain Res. Rev.* 60, 287–305. doi: 10.1016/J.BRAINRESREV.2008.12.023

Buxhoeveden, D. P., and Casanova, M. F. (2002). The minicolumn hypothesis in neuroscience. *Brain* 125, 935–951. doi: 10.1093/BRAIN/AWF110

Carrillo-Reid, L., Han, S., Yang, W., Akrouh, A., and Yuste, R. (2019). Controlling visually guided behavior by holographic recalling of cortical ensembles. *Cell* 178, 447–457.e5. doi: 10.1016/J.CELL.2019.05.045

Carrillo-Reid, L., Yang, W., Bando, Y., Peterka, D. S., and Yuste, R. (2016). Imprinting and recalling cortical ensembles. *Science (1979)* 353, 691–694. doi: 10.1126/science.aaf7560

Carter, O., van Swinderen, B., Leopold, D. A., Collin, S. P., and Maier, A. (2020). Perceptual rivalry across animal species. *J. Comp. Neurol.* 528, 3123–3133. doi: 10.1002/CNE.24939

Chrysanthidis, N., Fiebig, F., Lansner, A., and Herman, P. (2022). Traces of semantization, from episodic to semantic memory in a spiking cortical network model. *eNeuro* 9:ENEURO.0062-22.2022. doi: 10.1523/ENEURO.0062-22.2022

Cramer, B., Billaudelle, S., Kanya, S., Leibfried, A., Grübl, A., Karasenko, V., et al. (2022). Surrogate gradients for analog neuromorphic computing. *Proc. Natl. Acad. Sci. USA* 119:94119. doi: 10.1073/pnas.2109194119

DeFelipe, J., and Fariñas, I. (1992). The pyramidal neuron of the cerebral cortex: morphological and chemical characteristics of the synaptic inputs. *Prog. Neurobiol.* 39, 563–607. doi: 10.1016/0301-0082(92)90015-7

DiCarlo, J. J., Zoccolan, D., and Rust, N. C. (2012). How does the brain solve visual object recognition? *Neuron* 73, 415–434. doi: 10.1016/j.neuron.2012.01.010

Diehl, P. U., and Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Front. Comput. Neurosci.* 9:99. doi: 10.3389/FNCOM.2015.00099/BIBTEX

Douglas, R. J., and Martin, K. A. C. (2004). Neuronal circuits of the neocortex. *Annu. Rev. Neurosci.* 27, 419–451. doi: 10.1146/annurev.neuro.27.070203.144152

Douglas, R. J., and Martin, K. A. C. (2007). Recurrent neuronal circuits in the neocortex. *Curr. Biol.* 17, R496–R500. doi: 10.1016/J.CUB.2007.04.024

Eshraghian, J. K., Ward, M., Neftci, E. O., Wang, X., Lenz, G., Dwivedi, G., et al. (2023). Training spiking neural networks using lessons from deep learning. *Proc. IEEE* 111, 1016–1054. doi: 10.1109/JPROC.2023.3308088

Felleman, D. J., and Van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cereb. Cortex* 1, 1–47. doi: 10.1093/cercor/1.1.1

Fernandino, L., Tong, J. Q., Conant, L. L., Humphries, C. J., and Binder, J. R. (2022). Decoding the information structure underlying the neural representation of concepts. *Proc. Natl. Acad. Sci. USA* 119:e2108091119. doi: 10.1073/PNAS.2108091119/SUPPL_FILE/PNAS.2108091119.SD03.XLSX

Fiebig, F., and Lansner, A. (2017). A spiking working memory model based on hebbian short-term potentiation. *J. Neurosci.* 37, 83–96. doi: 10.1523/JNEUROSCI.1989-16.2016

Fransen, E., and Lansner, A. (1998). A model of cortical associative memory based on a horizontal network of connected columns. *Netw. Comput. Neural Syst.* 9, 235–264. doi: 10.1088/0954-898X_9_2_006

Fuster, J. M. (2006). The cognit: a network model of cortical representation. *Int. J. Psychophysiol.* 60, 125–132. doi: 10.1016/J.IJPSYCHO.2005.12.015

Geirhos, R., Medina Temme, C. R., Rauber, J., Schütt, H. H., Bethge, M., and Wichmann, F. A. (2018). Generalisation in humans and deep neural networks. *Adv Neural Inf Process Syst* 31.

George, D., Lehrach, W., Kansky, K., Lázaro-Gredilla, M., Laan, C., Marthi, B., et al. (2017). A generative vision model that trains with high data efficiency and breaks text-based CAPTCHAs. *Science* 358:358. doi: 10.1126/SCIENCE.AAG2612

Ghodrati, M., Farzmahdi, A., Rajaei, K., Ebrahimpour, R., and Khaligh-Razavi, S. M. (2014). Feedforward object-vision models only tolerate small image variations compared to human. *Front. Comput. Neurosci.* 8:96520. doi: 10.3389/FNCOM.2014.00074/ABSTRACT

Gripon, V., and Berrou, C. (2011). Sparse neural networks with large learning diversity. *IEEE Trans. Neural Netw.* 22, 1087–1096. doi: 10.1109/TNN.2011.2146789

Handjaras, G., Ricciardi, E., Leo, A., Lenci, A., Cecchetti, L., Cosottini, M., et al. (2016). How concepts are encoded in the human brain: a modality independent, category-based cortical organization of semantic knowledge. *NeuroImage* 135, 232–242. doi: 10.1016/J.NEUROIMAGE.2016.04.063

Harris, K. D. (2005). Neural signatures of cell assembly organization. *Nat. Rev. Neurosci.* 6, 399–407. doi: 10.1038/nrn1669

Hebb, D. O. (1949). The Organization of Behavior. Psychology Press: Psychology Press.

Holtmaat, A., and Svoboda, K. (2009). Experience-dependent structural synaptic plasticity in the mammalian brain. *Nat. Rev. Neurosci.* 10, 10, 647–658. doi: 10.1038/nrn2699

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities (associative memory/parallel processing/categorization/content-addressable memory/fail-soft devices). *Proc. Natl. Acad. Sci. USA* 79, 2554–2558.

Horner, A. J., Bisby, J. A., Bush, D., Lin, W. J., and Burgess, N. (2015). Evidence for holistic episodic recollection via hippocampal pattern completion. *Nat. Commun.* 6, 1–11. doi: 10.1038/ncomms8462

Hubel, D. H., and Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol.* 160, 106–154. doi: 10.1113/JPHYSIOL.1962.SP006837

Isaac, J. T. R., Nicoll, R. A., and Malenka, R. C. (1995). Evidence for silent synapses: implications for the expression of LTP. *Neuron* 15, 427–434. doi: 10.1016/0896-6273(95)90046-2

Kanter, I. (1988). Potts-glass models of neural networks. *Phys Rev A* 37:2739. doi: 10.1103/PhysRevA.37.2739

Kar, K., and DiCarlo, J. J. (2021). Fast recurrent processing via ventrolateral prefrontal cortex is needed by the primate ventral stream for robust Core visual object recognition. *Neuron* 109, 164–176.e5. doi: 10.1016/J.NEURON.2020.09.035

Kar, K., Kubilius, J., Schmidt, K., Issa, E. B., and DiCarlo, J. J. (2019). Evidence that recurrent circuits are critical to the ventral stream's execution of core object recognition behavior. *Nat. Neurosci.* 22, 974–983. doi: 10.1038/s41593-019-0392-5

Kerchner, G. A., and Nicoll, R. A. (2008). Silent synapses and the emergence of a postsynaptic mechanism for LTP. *Nat. Rev. Neurosci.* 9, 813–825. doi: 10.1038/nrn2501

Khona, M., and Fiete, I. R. (2022). Attractor and integrator networks in the brain. *Nat. Rev. Neurosci.* 23, 744–766. doi: 10.1038/s41583-022-00642-0

Kiefer, M., and Pulvermüller, F. (2012). Conceptual representations in mind and brain: theoretical developments, current evidence and future directions. *Cortex* 48, 805–825. doi: 10.1016/J.CORTEX.2011.04.006

Kietzmann, T. C., Spoerer, C. J., Sörensen, L. K. A., Cichy, R. M., Hauk, O., and Kriegeskorte, N. (2019). Recurrence is required to capture the representational dynamics of the human visual system. *Proc. Natl. Acad. Sci. USA* 116, 21854–21863. doi: 10.1073/pnas.1905544116

Kingma, D.P., and Ba, J.L., (2015). Adam: A method for stochastic optimization, in: 3rd international conference on learning representations, ICLR 2015 - conference track proceedings

Knoblauch, A., and Palm, G. (2020). Iterative retrieval and block coding in autoassociative and Heteroassociative memory. *Neural Comput.* 32, 205–260. doi: 10.1162/NECO_A_01247

Knoblauch, A., and Sommer, F. T. (2016). Structural plasticity, effectual connectivity, and memory in cortex. *Front. Neuroanat.* 10:180189. doi: 10.3389/FNANA.2016.00063/BIBTEX

Lamprecht, R., and LeDoux, J. (2004). Structural plasticity and memory. *Nat. Rev. Neurosci.* 5, 45–54. doi: 10.1038/nrn1301

Lansner, A. (2009). Associative memory models: from the cell-assembly theory to biophysically detailed cortex simulations. *Trends Neurosci.* 32, 178–186. doi: 10.1016/j.tins.2008.12.002

Lansner, A., and Ekeberg, Ö. (1989). A one-layer feedback artificial neural network with a Bayesian learning rule. *Int. J. Neural Syst.* 1, 77–87. doi: 10.1142/S0129065789000499

Lansner, A., Fransén, E., and Sandberg, A. (2003). Cell assembly dynamics in detailed and abstract attractor models of cortical associative memory. *Theory Biosci.* 122, 19–36. doi: 10.1007/S12064-003-0035-X/METRICS

Lansner, A., Ravichandran, N.B., and Herman, P., (2023). Benchmarking Hebbian learning rules for associative memory

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature* 521, 436–444. doi: 10.1038/nature14539

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 2278–2323. doi: 10.1109/5.726791

Liao, D., Scannevin, R. H., and Huganir, R. (2001). Activation of silent synapses by rapid activity-dependent synaptic recruitment of AMPA receptors. *J. Neurosci.* 21, 6008–6017. doi: 10.1523/JNEUROSCI.21-16-06008.2001

Linsker, R. (1988). Self-Organization in a Perceptual Network. Computer (Long Beach calif), vol. 21, 105–117.

Liu, K. Y., Gould, R. L., Coulson, M. C., Ward, E. V., and Howard, R. J. (2016). Tests of pattern separation and pattern completion in humans—a systematic review. *Hippocampus* 26, 705–717. doi: 10.1002/HIPO.22561

Lumer, E. D., Friston, K. J., and Rees, G. (1998). Neural correlates of perceptual rivalry in the human brain. *Science* 1979, 1930–1934. doi: 10.1126/SCIENCE.280.5371.1930/ASSET/9298D646-428E-4618-83F5-B9230E7DB77E/ASSETS/GRAPHIC/SE2686611003.JPEG

Lundqvist, M., Compte, A., and Lansner, A. (2010). Bistable, irregular firing and population oscillations in a modular attractor memory network. *PLoS Comput. Biol.* 6, 1–12. doi: 10.1371/JOURNAL.PCBI.1000803

Marković, D., Mizrahi, A., Querlioz, D., and Grollier, J. (2020). Physics for neuromorphic computing. *Nat. Rev. Phys.* 2, 499–510. doi: 10.1038/s42254-020-0208-2

Masquelier, T., and Thorpe, S. J. (2007). Unsupervised learning of visual features through spike timing dependent plasticity. *PLoS Comput. Biol.* 3:e31. doi: 10.1371/JOURNAL.PCBI.0030031

Mountcastle, V. B. (1957). Modality and topographic properties of single neurons of cat's somatic sensory cortex. *J. Neurophysiol.* 20, 408–434. doi: 10.1152/jn.1957.20.4.408

Mountcastle, V. B. (1997). The columnar organization of the neocortex. *Brain* 120, 701–722. doi: 10.1093/brain/120.4.701

O'Reilly, R. (1998). Six principles for biologically based computational models of cortical cognition. *Trends Cogn. Sci.* 2, 455–462. doi: 10.1016/s1364-6613(98)01241-8

O'Reilly, R. C., Wyatte, D., Herd, S., Mingus, B., and Jilk, D. J. (2013). Recurrent processing during object recognition. *Front. Psychol.* 4:124. doi: 10.3389/fpsyg.2013.00124

Palm, G. (1980). On associative memory. *Biol. Cybern.* 36, 19–31. doi: 10.1007/BF00337019

Pfeiffer, M., and Pfeil, T. (2018). Deep learning with spiking neurons: opportunities and challenges. *Front. Neurosci.* 12:774. doi: 10.3389/fnins.2018.00774

Plenz, D., and Thiagarajan, T. C. (2007). The organizing principles of neuronal avalanches: cell assemblies in the cortex? *Trends Neurosci.* 30, 101–110. doi: 10.1016/J.TINS.2007.01.005

Pulvermüller, F., Tomasello, R., Henningsen-Schomers, M. R., and Wennekers, T. (2021). Biological constraints on neural network models of cognitive function. *Nat. Rev. Neurosci.* 22, 488–502. doi: 10.1038/s41583-021-00473-5

Ravichandran, N.B., Lansner, A., and Herman, P., (2020). Learning representations in Bayesian confidence propagation neural networks, 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, 1–7

Ravichandran, N. B., Lansner, A., and Herman, P. (2021). "Brain-like approaches to unsupervised learning of hidden representations—a comparative study" in Artificial neural networks and machine learning-ICANN 2021 (Cham: Springer), 162–173.

Ravichandran, N., Lansner, A., and Herman, P., (2023a). Associative memory and deep learning with Hebbian synaptic and structural plasticity. ICML workshop on localized learning (LLW)

Ravichandran, N. B., Lansner, A., and Herman, P. (2023b). Brain-like combination of feedforward and recurrent network components achieves prototype extraction and robust pattern recognition. *Machine Learn. Optimiz. Data Sci.* 37, 488–501. doi: 10.1007/978-3-031-25891-6_37

Ravichandran, N., Lansner, A., and Herman, P., (2023c). Spiking neural networks with Hebbian plasticity for unsupervised representation learning. ESANN 2023 proceedings

Ravichandran, N., Lansner, A., and Herman, P., (2024). Unsupervised representation learning with Hebbian synaptic and structural plasticity in brain-like feedforward neural networks

Rolls, E., and Treves, A. (2012). Neural networks and brain function. Oxford Publications: Oxford University Press.

Rosch, E., (1988). Principles of categorization. Readings in Cognitive Science: A Perspective from Psychology and Artificial Intelligence, 312–322

Roy, S., and Basu, A. (2017). An online unsupervised structural plasticity algorithm for spiking neural networks. *IEEE Trans Neural Netw Learn Syst* 28, 900–910. doi: 10.1109/TNNLS.2016.2582517

Roy, K., Jaiswal, A., and Panda, P. (2019). Towards spike-based machine intelligence with neuromorphic computing. *Nature* 575, 607–617. doi: 10.1038/s41586-019-1677-2

Rozell, C. J., Johnson, D. H., Baraniuk, R. G., and Olshausen, B. A. (2008). Sparse coding via thresholding and local competition in neural circuits. *Neural Comput.* 20, 2526–2563. doi: 10.1162/neco.2008.03-07-486

Rullán Buxó, C. E., and Pillow, J. W. (2020). Poisson balanced spiking networks. *PLoS Comput. Biol.* 16:e1008261. doi: 10.1371/JOURNAL.PCBI.1008261

Rumelhart, D. E., and Zipser, D. (1985). Feature discovery by competitive learning. *Cogn. Sci.* 9, 75–112. doi: 10.1016/S0364-0213(85)80010-0

Sacouto, L., and Wichert, A. (2023). Competitive learning to generate sparse representations for associative memory. *Neural Netw.* 168, 32–43. doi:

Sa-Couto, L., and Wichert, A. (2020). Storing object-dependent sparse codes in a Willshaw associative network. *Neural Comput.* 32, 136–152. doi: 10.1162/NECO_A_01243

Sacouto, L., Wichert, A. (2023). Competitive learning to generate sparse representations for associative memory. *Neural Networks*, 168, 32–43. doi: 10.1016/J.NEUNET.2023.09.005

Salvatori, T., Millidge, B., Song, Y., Bogcaz, R., and Lukasiewicz, T. (2024). Associative memories in the feature space. *Front. Artif. Intellig. Appl.* 372, 2065–2072. doi: 10.3233/FAIA230500

Sanger, T. D. (1989). Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Netw.* 2, 459–473. doi: 10.1016/0893-6080(89)90044-0

Schuman, C. D., Kulkarni, S. R., Parsa, M., Mitchell, J. P., Date, P., and Kay, B. (2022). Opportunities for neuromorphic computing algorithms and applications. *Nat. Comput. Sci.* 2, 10–19. doi: 10.1038/s43588-021-00184-y

Shadlen, M. N., and Newsome, W. T. (1998). The variable discharge of cortical neurons: implications for connectivity, computation, and information coding. *J. Neurosci.* 18, 3870–3896. doi: 10.1523/jneurosci.18-10-03870.1998

Simas, R., Sa-Couto, L., and Wichert, A. (2023). Classification and generation of real-world data with an associative memory model. *Neurocomputing* 551:126514. doi: 10.1016/J.NEUCOM.2023.126514

Softky, W. R., and Koch, C. (1993). The highly irregular firing of cortical cells is inconsistent with temporal integration of random EPSPs. *J. Neurosci.* 13, 334–350. doi: 10.1523/jneurosci.13-01-00334.1993

Spoerer, C. J., McClure, P., and Kriegeskorte, N. (2017). Recurrent convolutional neural networks: a better model of biological object recognition. *Front. Psychol.* 8:1551. doi: 10.3389/fpsyg.2017.01551

Stepanyants, A., Hof, P. R., and Chklovskii, D. B. (2002). Geometry and structural plasticity of synaptic connectivity. *Neuron* 34, 275–288. doi: 10.1016/S0896-6273(02)00652-9

Stettler, D. D., Yamahachi, H., Li, W., Denk, W., and Gilbert, C. D. (2006). Axons and synaptic boutons are highly dynamic in adult visual cortex. *Neuron* 49, 877–887. doi: 10.1016/J.NEURON.2006.02.018

Taherkhani, A., Belatreche, A., Li, Y., Cosma, G., Maguire, L. P., and McGinnity, T. M. (2020). A review of learning in biologically plausible spiking neural networks. *Neural Netw.* 122, 253–272. doi: 10.1016/j.neunet.2019.09.036

Tang, H., Buia, C., Madhavan, R., Crone, N. E., Madsen, J. R., Anderson, W. S., et al. (2014). Spatiotemporal dynamics underlying object completion in human ventral visual cortex. *Neuron* 83, 736–748. doi: 10.1016/j.neuron.2014.06.017

Tang, M., Salvatori, T., Millidge, B., Song, Y., Lukasiewicz, T., and Bogacz, R. (2023). Recurrent predictive coding models for associative memory employing covariance learning. *PLoS Comput. Biol.* 19:e1010719. doi: 10.1371/JOURNAL.PCBI.1010719

Tang, H., Schrimpf, M., Lotter, W., Moerman, C., Paredes, A., Caro, J. O., et al. (2018). Recurrent computations for visual pattern completion. *Proc. Natl. Acad. Sci. USA* 115, 8835–8840. doi: 10.1073/pnas.1719397115

Tavanaei, A., Ghodrati, M., Kheradpisheh, S. R., Masquelier, T., and Maida, A. (2019). Deep learning in spiking neural networks. *Neural Netw.* 111, 47–63. doi: 10.1016/J.NEUNET.2018.12.002

Thorpe, S., Fize, D., and Marlot, C. (1996). Speed of processing in the human visual system. *Nature* 381, 520–522. doi: 10.1038/381520a0

van Bergen, R. S., and Kriegeskorte, N. (2020). Going in circles is the way forward: the role of recurrence in visual inference. *Curr. Opin. Neurobiol.* 65, 176–193. doi: 10.1016/J.CONB.2020.11.009

Van Vreeswijk, C., and Sompolinsky, H. (1996). Chaos in neuronal networks with balanced excitatory and inhibitory activity. *Science* 274, 1724–1726. doi: 10.1126/SCIENCE.274.5293.1724

Wagemans, J., Elder, J. H., Kubovy, M., Palmer, S. E., Peterson, M. A., Singh, M., et al. (2012). A century of gestalt psychology in visual perception I. Perceptual grouping and figure-ground organization. *Psychol. Bull.* 138:1172. doi: 10.1037/A0029333

Wichmann, F. A., Janssen, D. H. J., Geirhos, R., Aguilar, G., Schütt, H. H., Maertens, M., et al. (2017). Methods and measurements to compare men against machines. *Electr. Imaging* 29, 36–45. doi: 10.2352/ISSN.2470-1173.2017.14.HVEI-113

Willshaw, D. J., Buneman, O. P., and Longuet-Higgins, H. C. (1969). Non-holographic associative memory. *Nature* 222, 960–962. doi: 10.1038/222960a0

Wunderlich, T. C., and Pehle, C. (2021). Event-based backpropagation can compute exact gradients for spiking neural networks. *Sci. Rep.* 11:91786. doi: 10.1038/s41598-021-91786-z

Wyatte, D., Curran, T., and O'Reilly, R. (2012). The limits of feedforward vision: recurrent processing promotes robust object recognition when objects are degraded. *J. Cogn. Neurosci.* 24, 2248–2261. doi: 10.1162/jocn_a_00282

Yuste, R., Cossart, R., and Yaksi, E. (2024). Neuronal ensembles: building blocks of neural circuits. *Neuron* 112, 875–892. doi: 10.1016/J.NEURON.2023.12.008

Zenke, F., and Neftci, E. O. (2021). Brain-inspired learning on neuromorphic substrates. *Proc. IEEE* 109, 935–950. doi: 10.1109/JPROC.2020.3045625