

Connection between Hebbian Unlearning and Steady States Generated by Nonequilibrium Dynamics

Agnish Kumar Behera ¹, Matthew Du,^{1,2} Uday K. Jagadisan,³ Srikanth Sastry,⁴ Madan Rao ⁵
and Suriyanarayanan Vaikuntanathan ^{1,2}

¹*Department of Chemistry, University of Chicago, Chicago, Illinois 60637, USA*

²*The James Franck Institute, University of Chicago, Chicago, Illinois 60637, USA*

³*University of California, Berkeley, California 94720, USA*

⁴*Jawaharlal Nehru Center for Advanced Scientific Research, Bengaluru, India*

⁵*National Center for Biological Sciences, Bengaluru, India*



(Received 12 March 2025; accepted 24 September 2025; published 12 November 2025)

The classic paradigms for learning and memory recall focus on the strengths of synaptic couplings and how these can be modulated to encode memories. In this work, using analytical theory and computational inference schemes that use specialized restricted Boltzmann machines, we show that the dynamical steady state accessed due to a class of nonequilibrium detailed balance breaking dynamics is in fact similar to those accessed after the operation of a classic unsupervised scheme for improving memory recall, Hebbian unlearning, or “dreaming.” Our work suggests how nonequilibrium dynamics can provide an alternative route for controlling the memory encoding and retrieval properties of a variety of synthetic (neuromorphic) and biological systems.

DOI: [10.1103/vkt6-bltp](https://doi.org/10.1103/vkt6-bltp)

I. INTRODUCTION

Energy-based associative memory models have provided minimal yet powerful frameworks to understand how information storage and retrieval can be modulated in a variety of systems. The Hopfield model and its extensions have found applications, for example, in restricted Boltzmann machines [1–3], pattern recognition [4], understanding olfaction [5–7], etc. These ideas have also been recently applied to infer models from experimental data, for instance, data for fitness landscapes due to mutations, spike-time correlation data from the brain, etc. [8–10]. Moving beyond the original quadratic connectivity, newer variants of the Hopfield model have been explored with higher-order connectivity. These have been shown to have exotic properties like exponential capacities [11] and feature-to-prototype transitions in pattern recognition [11,12].

The typical formulation of an associative memory model relies on a learning strategy, such as the Hebbian rule, for encoding the desired memories or patterns. In seminal work, Hopfield *et al.* [13] showed how the memory capacity of such associative memory networks may be improved in an unsupervised manner through a so-called “unlearning” procedure. The unlearning algorithm provides a prescription for updating the connectivity of the associative memory network such that local minima in the free-energy landscape which do not correspond to desired patterns are cleaned out, leading to a higher capacity.

Here we consider an alternate paradigm and ask if phenomenology resembling Hebbian unlearning (or *dreaming*) can be realized by controlling the dynamics of the spins during pattern retrieval instead of explicitly changing the connectivity beforehand. Performing a series of numerical calculations (using specialized restricted Boltzmann machines) and analytical arguments, we demonstrate how such a modulation might be possible. Our central results suggest that the steady states achieved by a particular class of nonequilibrium dynamics resemble steady states achieved when the network is pruned using the Hebbian unlearning rule. The common feature in these dynamics is an element of persistence. We also show how the nonequilibrium dynamics with such *persistent* qualities may be naturally achievable in a model system of integrate-and-fire neurons. While this work is connected to previous work by some of us in Ref. [14], where we showed how dynamics with persistence can lead to better information processing properties, we substantially go beyond the scope of Ref. [14] in multiple ways, as discussed in detail in subsequent sections of the paper.

The rest of the paper is organized as follows: We begin in Sec. II, where we review the Hopfield model and the Hebbian unlearning procedure, colloquially referred to as *dreaming* in Ref. [13], for improving memory. In the Hebbian unlearning scheme, connections between neurons are altered explicitly in an unsupervised manner. In Sec. III, we consider a version of the Hopfield model and analytically show the equivalence between equilibrium steady states generated by *dreaming* and nonequilibrium steady states generated with a class of *persistent* dynamics. These calculations are done in a perturbative limit. In order to perform our analytical perturbative calculations, we resort to a version of the Hopfield model with continuous degrees of freedom. Such models have been shown to have behavior qualitatively similar to the standard Hopfield

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

model. Next, in Sec. IV, we attempt to show that the equivalence persists beyond the perturbative limit. We introduce a version of persistence in the discrete Hopfield model. We show how this can be considered as a coarse-grained version of the continuous model and numerically (using specialized restricted Boltzmann machines) demonstrate how persistence and Hebbian unlearning have similar qualitative effects on the memory storage and recall properties of the system. In Sec. V, we discuss a biologically plausible model of integrate-and-fire neurons that display persistence in their dynamics through a different mechanism. We connect this biologically plausible model with the models introduced previously and show how persistence in the dynamics potentially leads to effects that are similar to those in the previous sections. Finally, in Sec. VI we discuss the implications of our work and future directions.

II. HOPFIELD MODEL AND HEBBIAN UNLEARNING

The Hopfield model is an associative memory model consisting of N fully connected Ising-like spins, $\sigma_i \in \{\pm 1\}$ for $i = 1, \dots, N$, which represent discretized neuron firing rates. The energy of the system is given by the Hamiltonian

$$H = - \sum_{ij} J_{ij} \sigma_i \sigma_j, \quad (1)$$

where, following the Hebb rule, the connectivity matrix J is

$$J_{ij} = (1/N) \sum_{\mu=1}^p \xi_i^\mu \xi_j^\mu, \quad (2)$$

with the patterns $\vec{\xi}^1, \dots, \vec{\xi}^p \in \{\pm 1\}^N$. The patterns ξ_i^μ are drawn from the Rademacher distribution. Using this energy function, the spins are dynamically updated asynchronously according to the Glauber algorithm at a chosen temperature scale T [15]. The ability of the network to store and retrieve patterns is probed by initializing the system near a stored pattern (with $\leq 10\%$ bits flipped) and running the dynamics [16]. If the asymptotic configuration that the system settles in has a good overlap $\mathcal{O}$ with the stored pattern,

$$\mathcal{O} \equiv \lim_{t \rightarrow \infty} \frac{1}{N} \vec{\sigma}_t \cdot \vec{\xi}^{\text{stored}} \approx 1, \quad (3)$$

where $\vec{\sigma}_t$ denotes the spin configuration after t sweeps of the chosen Monte Carlo algorithm, then retrieval is deemed successful. Under such dynamics, the system can store up to $\alpha_c N$ patterns, where $\alpha_c \approx 0.14$ at $T = 0$. The retrieval capacity decreases with temperature [17,18].

The so-called ‘‘Hebbian unlearning’’ procedure—proposed in Ref. [13]—and its variants can be used to improve the critical memory capacity in an unsupervised manner. The proposed algorithm works as follows: The system is initialized in a random state and allowed to evolve till it reaches a local steady state. Then, the connectivity matrix is updated as

$$J_{ij} \leftarrow J_{ij} - \epsilon \sigma_i^* \sigma_j^*, \quad (4)$$

where $\vec{\sigma}^*$ is the configuration of the local steady state accessed by the system, and ϵ is the unlearning rate. We refer to each update as a ‘‘dream’’ following conventions in Ref. [19]. This scheme, though unsupervised, has been shown to increase the associative memory capacity up to around

$\alpha_c \approx 0.5$. The effectiveness of the Hebbian unlearning method can be heuristically explained as follows: The number of possible configurations the system can access grows exponentially with the system size N . Of these, only a polynomial fraction αN corresponds to stored patterns. If the system is initialized in a random configuration and is allowed to sample its local energy or free-energy landscape using the above-described dynamics [using (1) and (2)], it has a higher probability of settling into a spurious energy minima than into a pattern basin [17]. The unlearning update rule in Eq. (4) can hence effectively be viewed as a mechanism to raise the energies of the spurious configurations ($\vec{\sigma}^*$) [13] and destabilize them. The ability to retrieve stored memories can be relatively enhanced as a consequence.

For completeness, we note that Hebbian unlearning operates by a mechanism similar to the contrastive divergence (CD) [20,21] and forward-forward (FF) [22] algorithms in machine learning. However, there are some notable differences. In CD, learning and unlearning of spurious states occur at each step within a single training phase. In contrast, Hebbian unlearning comes after the initial connectivity matrix is learned [e.g., by the Hebb rule, Eq. (2)]. While FF also features learning and unlearning in two separate stages, the algorithm is designed for training feed-forward neural networks on supervised learning tasks like classification.

III. PLAUSIBLE EQUIVALENCE BETWEEN STEADY STATES PRODUCED BY HEBBIAN UNLEARNING AND STEADY STATES PRODUCED BY NONEQUILIBRIUM PERSISTENT DYNAMICS: A PERTURBATIVE CALCULATION

We now argue that there is a plausible equivalence between Hebbian unlearning and steady states generated by a class of nonequilibrium dynamics with persistence. We begin by establishing the equivalence analytically in a perturbative limit. To do so, we consider a version of the Hopfield model with continuous degrees of freedom. Such models have been shown to have properties that qualitatively resemble those of the original Hopfield formulation. These models are also more plausible biologically because their continuous degrees of freedom allow for a more accurate representation of neuronal dynamics. Specifically, we consider a system described by the Hamiltonian

$$H = - \sum_{ij} J_{ij} f(\sigma_i) f(\sigma_j), \quad (5)$$

where f is meant to mimic a neuronal activation function (e.g., *sigmoid* or ReLU function) which models the spiking response of real neurons. The spins σ_i are assumed to be continuous variables and evolve according to the dynamics

$$\frac{\partial \sigma_i}{\partial t} = - \frac{\partial H}{\partial \sigma_i} + \eta_i(t), \quad (6)$$

where η_i is a Gaussian white noise with statistics

$$\langle \eta_i(t) \rangle = 0, \quad \langle \eta_i(t) \eta_j(t') \rangle = 2T \delta_{ij} \delta(t - t'). \quad (7)$$

A Hebbian unlearning procedure can be readily developed for this generalized Hopfield model [23]. Specifically, a random state $\vec{\sigma}^*$ is chosen as described in Appendix D and the

connectivity matrix is updated as

$$J_{ij} \leftarrow J_{ij} - \epsilon h_i^* h_j^*, \quad (8)$$

where $h_i^* \equiv -\partial H / \partial \sigma_i = \sum_j J_{ij} f(\sigma_j) f'(\sigma_i)$ is the effective field felt by spin i when the system is in a random configuration. Plugging $\tilde{h}^*$ into the update rule and averaging over many trials, as explained in Appendix D, we find that the new connectivity matrix is equal to

$$J_{ij}^D = J_{ij} - \epsilon \sum_p J_{ip} J_{jp} f'(\sigma_i) f'(\sigma_j). \quad (9)$$

Upon many such updates to the connectivity matrix, one can show that the resulting Hamiltonian, with an activation function satisfying $f'(\sigma_i) = 1$, takes the form [23]

$$H = -\frac{1}{N} \sum_{ij\mu\nu} \sigma_i \xi_i^\mu (1 + \lambda^2 C)_{\mu\nu}^{-1} \xi_j^\nu \sigma_j, \quad (10)$$

where $\lambda^2 = \epsilon d$, where d denotes the number of updates used, C is the pattern correlation matrix given by, $C_{\mu\nu} = \frac{1}{N} \sum_i \xi_i^\mu \xi_i^\nu$. Appendix D details the procedure to obtain (10) from (9). If the number of updates, d , is small, we can express the new connectivity matrix as

$$J_{ij}^D = J_{ij} - \epsilon d \sum_p J_{ip} J_{jp} f'(\sigma_i) f'(\sigma_j). \quad (11)$$

Next, we show that, in a certain perturbative limit, spins when forced out of equilibrium with a persistent noise source reach a steady state which can be expressed using an effective Hamiltonian that has a form similar to that achieved by Hebbian unlearning. Specifically, we imagine replacing the noise source in Eq. (6) with a noise source with statistics,

$$\langle \eta_i(t) \rangle = 0,$$

$$\langle \eta_i(t) \eta_j(t') \rangle = 2T_p \delta_{ij} \delta(t - t') + \frac{T_a}{\tau} \exp\left(-\frac{|t - t'|}{\tau}\right), \quad (12)$$

where τ is the persistence time of the temporal correlations, T_p and T_a set the strengths of the uncorrelated and correlated noise sources, respectively. Systems with such persistent dynamics are known to generate nonequilibrium steady states [24] and are colloquially referred to as active matter systems. In general, the distribution of such states cannot be described using an energy function [24] but for $\tau \ll 1$, the spins effectively sample from a distribution given by, $P(\vec{\sigma}) \propto \exp(-\beta_{\text{eff}} H_{\text{eff}})$, where [24]

$$H_{\text{eff}} = H + \frac{\tau T_a}{T_{\text{eff}}} \left(\frac{1}{2} |\nabla H|^2 - T_{\text{eff}} \nabla^2 H \right) \quad (13)$$

is an effective Hamiltonian, $T_{\text{eff}} = T_p + T_a$ and $\beta_{\text{eff}} = 1/T_{\text{eff}}$. For activation functions which are monotonic in nature, like ReLU and tanh, the Laplacian of the Hamiltonian, $\nabla^2 H$, has the same sign as the original J_{ij} . For instance, when $f(\sigma) = \tanh(\sigma)$, then $f''(\sigma) = -2\text{sech}^2(\sigma) \tanh(\sigma) = -2f(\sigma)\text{sech}^2(\sigma)$ and the renormalized connectivity matrix is $J_{ij}^A = J_{ij} \{1 + \tau T_a [\text{sech}^2(\sigma_i) + \text{sech}^2(\sigma_j)]\} - \frac{\tau T_a}{T_{\text{eff}}} \sum_p J_{ip} J_{jp} |f'(\sigma_p)|^2$. The contribution of the Laplacian [namely, the $\text{sech}^2(\sigma_i)$ term] is a two-body interaction term and has the same sign as the original J_{ij} . This ensures that the Laplacian term does not change the form of the connectivity, i.e.,

excitatory connections remain excitatory and similarly for inhibitory connections. It leads to a simple spin-dependent renormalization of the connectivity matrix. We expect the Laplacian term to be insignificant for most spin configurations and for typical choices of f . When $f(\sigma) = \tanh(\sigma)$, the Laplacian term rescales J_{ij} by a factor that contains $\text{sech}^2(\sigma_i) + \text{sech}^2(\sigma_j)$, which is $O(0.1)$ for $|\sigma_i| = O(1)$, and zero as $|\sigma_i| \rightarrow \infty$. Thus, the change in the connectivity matrix due to the Laplacian term can mostly be ignored in comparison with the original J_{ij} for most of the range of spin values. Altogether, the simple scale renormalization of the spin matrix due to the Laplacian term, and its resultant low magnitude, implies that it cannot protect the corruption of a memory basin due to the multiplicity of forces from the other stored memories. The latter is precisely what unlearning seems to counteract. Given this and also for mathematical simplicity, in what follows, we set aside the Laplacian term and focus on the $|\nabla H|^2$ term.

Substituting the form of (5) (with only the $|\nabla H|^2$ term) in (13), we arrive at a renormalized connectivity matrix,

$$J_{ij}^A = J_{ij} - \frac{\tau T_a}{T_{\text{eff}}} \sum_p J_{ip} J_{jp} |f'(\sigma_p)|^2. \quad (14)$$

Hence, (14) is equivalent to (11) with the Hebbian unlearning parameter $\lambda^2 \equiv \epsilon d = \tau T_a / T_{\text{eff}}$ and activation function satisfying $f'(\sigma) = 1$. Ideally, exploring the memory capacity of systems with these updated connectivity matrices would suffice to numerically verify our equivalence. However, given that $\epsilon d = \tau T_a / T_{\text{eff}} \ll 1$, a numerical signature of this equivalence would be hard to infer. Thus, we instead perform the following indirect calculation (detailed in Appendix E). We first generate a Hebbian-like connectivity matrix J_{ij} . Viewing Eqs. (11) and (14) as update rules, we then repeatedly apply these operations to generate new connectivity matrices. The results for $\epsilon d = \tau T_a / T_{\text{eff}} = 10^{-5}$ after 10^4 updates in this manner are plotted in Fig. 1. Both models show the same qualitative behavior. This equivalence suggests that the sampling generated by activity in the limit of small persistence times can be similar to that generated by an “infinitesimal” Hebbian unlearning procedure.

Can this phenomenology hold for arbitrary persistence times? Motivated by the equivalence of connectivity from Hebbian unlearning and that due to nonequilibrium activity in the aforementioned perturbative limit, we now use numerical inference schemes built around restricted Boltzmann machine (RBM) architectures to search for a broader equivalence. In what follows, we shift back to the discrete variant of the Hopfield model for two reasons. First, for large persistence times, the analytical expansion we used for the continuous variant of the Hopfield model breaks down [14] and becomes intractable. This negates one of the main benefits (i.e., analytical tractability) associated with switching from discrete to continuous representation. Second, and more importantly, inference algorithms for problems with continuous degrees of freedom are known to suffer from issues of poor convergence [21,25]. As noted previously, we expect the qualitative features associated with both continuous and discrete spins to be similar and hence expect no loss of generality due to this transition between continuous and discrete spins. In the

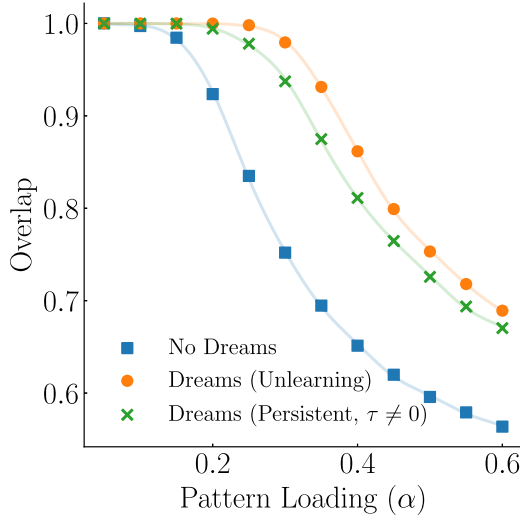


FIG. 1. Comparing the information storage limits of connectivity matrices obtained by iteratively applying (11) (Unlearning) and (14) (Persistent) for 10^4 iterations with $\epsilon d = \tau T_a / T_{\text{eff}} = 10^{-5}$. We plot the overlap $\mathcal{O}$ averaged over 10 systems (i.e., sets of patterns) with $P = \alpha N$ patterns and $N = 1000$ spins. The activation function $f(x) = \tanh(x)$ was used in all cases. For the standard Hopfield model (no dreams), the overlap decays quickly once the pattern loading is around 0.2. The profiles generated for the two iterative procedures (11) (unlearning) and (14) (persistent) have similar qualitative behavior and show the ability to store a larger number of patterns. The simulation details are given in Appendix E.

next section, we devise dynamics with discrete spins that have a persistence-like feature and use special neural network architectures to show that the steady states sampled resemble steady states that would have been sampled if an unlearning procedure were used.

IV. NUMERICAL INFERENCE SCHEMES USING HYBRID RESTRICTED BOLTZMANN MACHINES REVEAL SIGNATURES OF HEBBIAN UNLEARNING IN PERSISTENT DYNAMICS

The results in the previous section suggest that active dynamics driven by persistence noise sources can improve memory in a manner similar to Hebbian unlearning. We now use numerical inference schemes, assisted by RBM architectures, to show that such phenomenology can potentially be observed robustly across many systems. We develop a model that retains the persistent nature of the dynamics described in the previous section, but now the spins take discrete values instead of being continuous. We switch to spins that can take discrete values in order to generate configurations that are readily amenable to analysis using an RBM architecture. Specifically, we start with the standard Hopfield model [13] and modify the dynamics so that terms reminiscent of persistence are introduced.

A standard way to take a discrete spin system out of equilibrium is to introduce asymmetry into the connectivity J_{ij} such that the system is no longer described by an energy function, but the steady-state solution is still available [26,27] but here our focus is on persistent dynamics breaking detailed

balance. To mimic persistent dynamics in a discrete system, we designed a process where the evolution of the spins depends on both the current state and the previous state of the system [28]. In particular, the dynamics have an element of persistence. A spin flip (no spin flip) at an instant of time increases the probability of a spin flip (no spin flip) in the subsequent instant of time. The evolution of the spins is now Markovian only if both the current and previous state of the system are recorded together. For a single spin, such dynamics can be compactly represented by the following matrix equation:

$$\begin{pmatrix} ++ \\ +- \\ -+ \\ -- \end{pmatrix}_{t,t+1} = \begin{pmatrix} p+\gamma & 0 & p-\gamma & 0 \\ q-\gamma & 0 & q+\gamma & 0 \\ 0 & p+\gamma & 0 & p-\gamma \\ 0 & q-\gamma & 0 & q+\gamma \end{pmatrix} \times \begin{pmatrix} ++ \\ +- \\ -+ \\ -- \end{pmatrix}_{t-1,t}, \quad (15)$$

where the element $(\sigma\sigma')_{t,t+1}$ denotes the joint probability of sampling a spin configuration σ at time t and σ' at time $t+1$. Elements $(\sigma\sigma')_{t-1,t}$ have a similar interpretation. The typical equilibrium or passive transition probabilities according to Glauber dynamics, $p_i = 1/[1 + \exp(-2\beta h_i)]$ and $q_i = 1 - p_i$ are of Boltzmann form, where $\beta = 1/T$ is the inverse temperature and $h_i = \sum_j J_{ij}\sigma_j$ is the effective field felt by spin i . To capture the persistence of the continuous persistent dynamics [(6) and (12)], we introduced the factor $\gamma = a \min(p, q)$, which favors flips to be followed by flips (columns 2 and 3 of the transition matrix) and no-flips to be followed by no-flips (columns 1 and 4 of the transition matrix). The parameter a controls the persistence, with $a = 0$ corresponding to the passive dynamics.

Such discrete persistent dynamics can naturally arise from a temporal coarse-graining of the continuous persistent dynamics of the previous section (Sec. III). To see this, we consider a single continuous one-dimensional spin under noise with persistence time τ . We first suppose that the spin initially has some sign (e.g., $-$) and is kicked by the persistent noise toward the direction of opposite sign ($+$). If τ is sufficiently large, the spin is likely to have flipped ($- \rightarrow +$) after some time $t_1 \approx O(\tau)$. The persistent noise source will change direction (on average) after this and the spin is also likely to have changed its direction of movement. Then, after some time $t_2 \approx O(\tau)$, the spin is likely to have flipped again ($+ \rightarrow -$). Thus, flips are more likely to be followed by flips if time is discretized in chunks of $O(\tau)$. If we instead suppose that the spin (e.g., sign $-$) is initially kicked toward the direction of same sign ($-$), then we conclude that no-flips are more likely to be followed by no-flips. As the timescale associated with persistence is lowered, the likelihood of spin flips followed by spin flips decreases.

Figure 2 describes the phase diagram of the Hopfield model with the persistent dynamics. For a certain region of the phase diagram, the persistent dynamics is found to improve pattern retrieval, relative to passive dynamics. This behavior is qualitatively similar to that obtained in previous work with

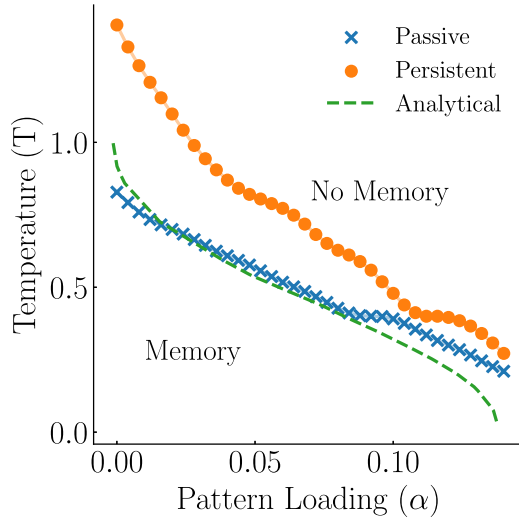


FIG. 2. Phase diagram of the discrete Hopfield model under equilibrium spin flips (passive) compared with the phase diagram obtained from the persistent active simulations in Sec. IV. The plot was made with $N = 500$ spins and $a = 0.5$. The results are an average over ten systems (i.e., sets of patterns) and attempts to retrieve each of the first ten patterns ($\xi^1, \dots, \xi^{10}$) from a corrupted version of the pattern with 5% of the spins flipped. The blue crosses denote the region of retrieval for passive dynamics, whereas the orange dots denote the retrieval region for persistent dynamics (15). The results are qualitatively similar to what is observed in the spherical Hopfield model under activity [14]. The analytical curve is taken from Ref. [17] and is provided as a comparison to the passive numerical simulations.

the spherical Hopfield model under activity [14]. Note that T here simply refers to the temperature at which the probabilities of transition, p and q , of (15) were calculated. The persistent dynamics used here does not possess a natural notion of a temperature.

In order to numerically demonstrate an equivalence similar to (11) and (14), we use numerical inference schemes for estimating the connectivity matrices J_{ij} that best explain the available data. Extracting the connectivity matrix from samples generated from dynamics is a challenging problem, and many techniques have been developed in the context of finding connectivity of real biological neuronal networks [9]. For this study, we used a RBM to extract the connectivity matrix. RBMs are a recurrent neural network architecture that have been used extensively for approximating a data distribution and generating further data from the same distribution [21,25,29]. RBMs with a single hidden layer have a direct correspondence with the Hopfield model at equilibrium (passive) [1,30,31]. Specifically, if the hidden layer consists of continuous variables with a Gaussian prior, then the probability of observing a certain configuration of hidden and visible nodes is given by

$$P(\sigma, z) \propto \exp \left(-\beta \sum_{\mu} \frac{z_{\mu}^2}{2} - \beta \sum_{\mu i} \frac{\xi_i^{\mu}}{\sqrt{N}} z_{\mu} \sigma_i \right), \quad (16)$$

where σ denotes a visible node variable, z a hidden node variable, and ξ_i^{μ} the connection strength between nodes σ_i and

z_{μ} . To see the connection to the Hopfield model, one has to marginalize the distribution of configurations by integrating out the hidden variables z_{μ} . This leads to

$$P(\sigma) \propto \exp \left(\beta \sum_{ij} J_{ij} \sigma_i \sigma_j \right), \quad J_{ij} = \frac{1}{N} \sum_{\mu} \xi_i^{\mu} \xi_j^{\mu}, \quad (17)$$

which is equivalent to the equilibrium distribution generated with the Hebbian rule for the Hopfield model. Indeed, the RBM connectivity matrix corresponds to the pattern variables of the Hopfield model. This makes the RBM an ideal system for extracting the connectivity matrix.

For our purposes, we need an RBM that can, in principle, represent the connectivity matrices that can emerge after Hebbian unlearning, the Hamiltonian for which is given by (10). This Hamiltonian has two parameters: a parameter λ that specifies the extent of Hebbian unlearning, and an effective temperature T_{RBM} . For the passive case, T_{RBM} equals the temperature T at which the simulations were carried out. The persistent simulations lack a notion of “temperature,” and thus T_{RBM} needs to be inferred. Such a Hamiltonian can be exactly represented as a three-layer RBM with a visible layer with neurons having discrete values and two hidden layers with continuous-valued neurons [19]. This can readily be seen when the partition function corresponding to the Hebbian unlearned Hamiltonian (10) is expressed as a Hubbard-Stratonovich transform,

$$Z = \int Dz D\phi \sum_{\sigma} \exp [-\beta_{\text{RBM}} H_{\text{RBM}}], \quad (18)$$

where the H_{RBM} is given by

$$H_{\text{RBM}} = \frac{z_{\mu}^2}{2} + \frac{\phi_i^2}{2} - z_{\mu} \frac{\xi_i^{\mu}}{\sqrt{N}} (\sigma_i + i\lambda \phi_i), \quad (19)$$

and $\beta_{\text{RBM}} = 1/T_{\text{RBM}}$. This expression for the partition function suggests an architecture where a visible layer of neurons is connected to a hidden layer with variables z_{μ} through weights $\xi_i^{\mu}/\sqrt{N}$ and this hidden layer is again connected to another hidden layer with imaginary variables ϕ_i through weights $i\xi_i^{\mu}\lambda/\sqrt{N}$. We train this deep hybrid RBM (DHBM) with the data from persistent and passive simulations and extract the weights ξ_i^{μ} , T_{RBM} , and λ .

We follow a procedure similar to the one outlined in Ref. [32] to train our RBM with data (see Fig. 3 for a schematic of the procedure and Appendix F for additional details). First, we generate data using the simple Hopfield dynamics (passive) and the non-Markovian dynamics (persistent). Then we train two different deep hybrid RBMs using these data, one for the passive case and one for the persistent case. This training allows us to infer values of λ for the two cases. Recall that λ is a measure of how much “dreaming” has taken place. Our simulations (Fig. 4) show that the dreaming parameter λ inferred from the persistent dynamics indeed has a statistically significant nonzero value. The inference procedure, when applied to the passive data, leads to values of λ^2 closer to zero (with some statistical uncertainty). Further, the T_{RBM} learned for passive simulations is very close to the actual temperature T at which the data were generated, whereas for the persistent case, $T_{\text{RBM}} < T$. Since the reconstruction error

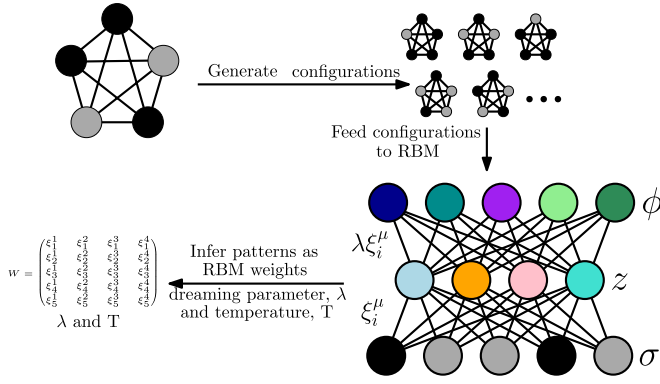


FIG. 3. Schematic for the procedure of using RBMs in testing our results. First, we generate a set of configurations at a *prescribed* temperature using a *specific* choice of dynamics. We denote this temperature T_a or T_p , where the subscript denotes either persistent or passive dynamics, respectively. Using these configurations, we infer the trainable parameters for the system, RBM weights (W^{new}), T , and λ . As described in Appendix F 3, we fix the weights of the RBM to the patterns and just learn the dreaming parameter, λ , and the (effective) temperature of the simulations.

is low (see Fig. 13 of Appendix G), the Hamiltonian generated by Hebbian unlearning (10) appears to be a good ansatz for the effective Hamiltonian arising from activity. The details of the inference procedure, along with additional numerical checks, are provided in the appendix. In Appendix G 3, we discuss how the notional temperature can be lower for persistent simulations.

We also infer λ for various combinations of α , T , and a . For the set of inferences in this section, we use mode (3) of Appendix F 2, i.e., we fix the weights of the RBM to the patterns, $W_i^\mu = \xi_i^\mu$, and infer T_{RBM} and λ values. To ease readability, only the inferred λ values have been plotted. In Fig. 5, the data were generated using $\alpha = 0.06, 0.08, 0.1$, $T = 0.4, 0.6, 0.8$, and $a = 0.0, 0.2, 0.4, 0.6$ using all combinations. The general trends indicate that λ is not sensitive to the temperature T of data generation but increases (decreases) with an increase in $a(\alpha)$. The mild dependence on T is expected because T , in the discrete formalism [as given in (15)], is like a simple noise source and should not couple well with the nonequilibrium activity parameter a . Since a is the parameter for the extent of deviation from equilibrium, the increase of λ with a is expected, although one could expect a nonmonotonic behavior. The reason why λ decreases with the pattern loading, α , is unclear, and understanding it is one of our future directions.

V. DYNAMICAL MODULATION OF MEMORY STATES IN NEURONS: A MINIMAL MODEL

We have specialized most of our results to the case where persistence is introduced in the noise degree of freedom. It is not clear if such tuning can be biologically plausible. Indeed, it is not immediately obvious how a biological system might modulate its noise sources in a manner that allows it to introduce and control an element of persistence. In this section, we explore a mechanism by which elements like persistence can be introduced into a minimal model of neuronal dynamics. For this purpose, we use an integrate-and-fire spik-

ing neuron model where, instead of treating the temporally coarse-grained spiking rate as a discrete or continuous spin, we explicitly consider the spike train, and the persistence in dynamics is a consequence of different methods of integrating the spikes. While this mechanism deviates in details from the simplified constructs of the last two sections, we find that the qualitative trends of the previous sections hold. In effect, modulating persistence in a biologically plausible way (in our minimal model) impacts the memory capacity of the system in a manner similar to unlearning. We anticipate that these findings will help enable applications of our work for biological and neuromorphic systems (Fig. 6).

The model for spiking neurons, adapted from Ref. [33], consists of integrate-and-fire neurons that integrate the spiking of neurons in their vicinity and then fire an action potential. The rate of firing of such a neuron, $R_i(t)$, contains information about the state of the entire system and is a Poisson-like process which can be mathematically expressed as,

$$r_i(t) = r_m g \left(\sum_j J_{ij} f \circ R_j(t) - \theta_i \right), \quad (20)$$

where $R_i(t) = r_i(t) + \eta_i(t)$ is the firing rate and $r_i(t)$ is the mean firing rate of neuron i . Here η_i are the fluctuations in that rate with $\langle \eta_i(t) \rangle = 0$ and $\langle \eta_i(t) \eta_j(t') \rangle = \delta_{ij} \delta(t - t') r_i(t)$ (see Appendix A for details). The function g is an activation function that modulates the all-or-nothing-like firing behavior of neurons. Here, it is chosen to be $g(x) = 1/(1 + e^{-x})$. The parameter r_m sets the maximum firing rate for neurons, θ_i is the threshold for neuron i , and J_{ij} denotes the connection strength between neurons i and j . The function f represents the postsynaptic response function in real neurons and is used as the convolution window by the postsynaptic neuron to integrate the spiking of neurons in its vicinity. The operation $f \circ R$ denotes a convolution, i.e., $(f \circ R)(t) = \int f(t - t') R(t') \Theta(t - t') dt'$ where $\Theta(t - t')$ denotes the Heaviside function and is required to ensure causality. This model [33] features stable states and therefore serves as a bridge between Hopfield models and the broad class of spiking neuron models; we refer the reader to Ref. [34] for models related to Ref. [33] and further discussion about spiking neuron models.

The function f can generically be expected to have a rise and a fall timescale, $f = \exp(-t/\tau_1) - \exp(-t/\tau_2)$. In particular, the rise time plays the role of persistence, where past spikes (from times $\sim \tau_2$ ago) have greater influence on the dynamics, while more recent spikes have less [Fig. 6(a)].

As outlined in Appendix A, by leveraging the form of f in (20), the equations for the firing rates take the form of (for a two-neuron system)

$$\tau_1 \tau_2 \dot{p}(t) + \tau_1 p(t) + \tau_2 p(t) + x(t) = r_m J (y(t) - \frac{1}{2}) + N(t), \quad (21)$$

where $y_i(t) = r_i(t)/r_m$ and, $x_i(t) \equiv g^{-1}(y_i(t))$ and $p_i(t) \equiv dx_i(t)/dt$, $y(t) \equiv \frac{1}{2}[y_1(t) + y_2(t)]$, $x(t) \equiv \frac{1}{2}[x_1(t) + x_2(t)]$, $p(t) \equiv \frac{1}{2}[p_1(t) + p_2(t)]$. Here, the subindices 1 and 2 refer to the two neurons. For large firing rates, we can approximate $N(t)$ to be a Gaussian process, $\langle N(t) \rangle = 0$, $\langle N(t) N(t') \rangle = \delta(t - t') y(t)$ [33]. When f lacks

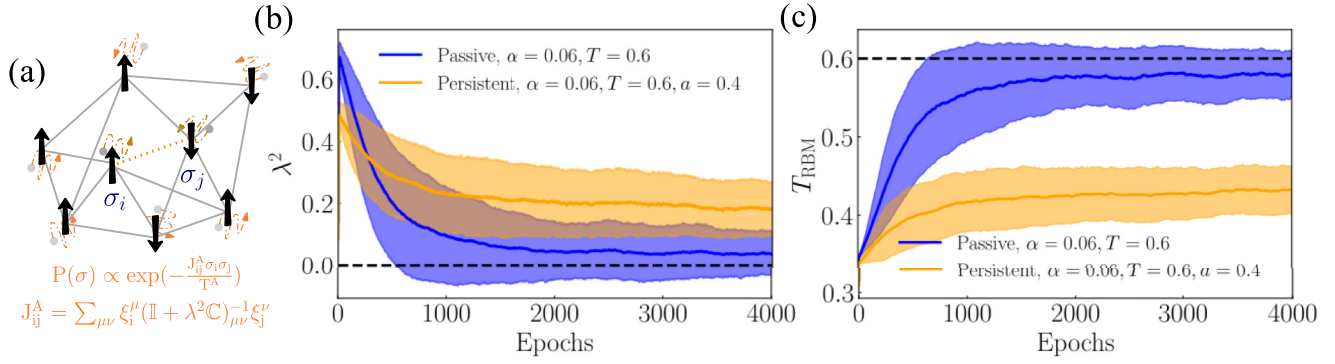


FIG. 4. Inferring T_{RBM} and λ from data. Panel (a) shows the ansatz for the sampling in the presence of persistent noise and how the couplings get modified along with the effective temperature. $\lambda = 0$ for the passive case, where the connectivity becomes the simple Hebbian connectivity. $\lambda > 0$ implies dreaming has taken place. In panels (b) and (c), we fix the patterns and let the system learn a λ and the temperature for the RBM, T_{RBM} . The persistent case learns a λ which is higher than the passive case (which is close to the ground truth of zero). This implies that the sampling due to activity is similar to dreaming. The performance of the RBM is captured with two metrics, the reconstruction loss and the pseudo log-likelihood loss. The corresponding plots are provided in Appendix G4.

a rise-time scale, i.e., $\tau_2 = 0$, (21) reduces to

$$\tau_1 p(t) = -x(t) + r_m J(y(t) - \frac{1}{2}) + N(t). \quad (22)$$

Equation (22) has features similar to an overdamped system, whereas (21) has features similar to those found in underdamped systems. Specifically, it is the persistent dynamics due to the rise time, τ_2 that gives rise to the inertial [first term on the left-hand side (LHS) of (21)] and dissipative [third term on the LHS of (21)] contributions. This analogy is discussed in more detail in Appendix B. This persistence in integration of action potentials arriving at a specific neuron from its environment can lead to a change in the distribution of the firing rates. This is depicted in Fig. 6(b). Here, the order parameter of the system is y , the scaled average of the firing rates of the two neurons. Specifically, persistence with $\tau_2 > 0$, leads to a sharpened distribution of y at its peaks [minima of $-\log(P_{ss}(y))$]. Such a change in the distribution of firing rates reflects the improved retrieval of information stored in the connectivity between neurons (as in the Hopfield model), as depicted in Fig. 6(b). The sharpening of minima with changing τ_2 is depicted in Figs. 7(a) and 7(b), where, upon increasing τ_2 , the curvature k_1 and k_2 of the observed firing rates around the minima increase.

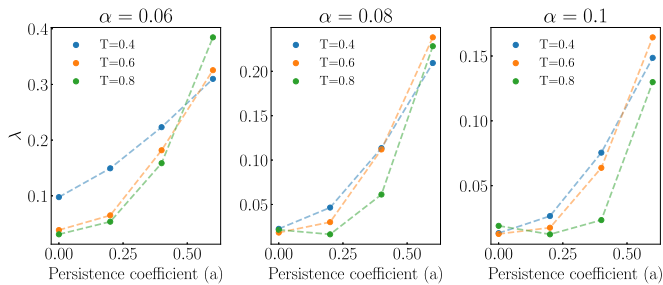


FIG. 5. Inferred λ values for various combinations of α , T , and a . Increasing T , the temperature for data generation does not seem to affect the λ significantly. Increasing the nonequilibrium driving by increasing a generally seems to increase λ , whereas increasing the pattern loading α seems to decrease λ .

When $\tau_2 = 0$, $\tau_1 \rightarrow 0$, and the activation function g is linear, the memory recall properties of the spiking neuron model resemble those of a standard Hopfield model. In the following, we analytically investigate how Hebbian unlearning changes the nature of the pattern retrieval landscape. Our calculations show that the change is qualitatively similar to the above-described change in the curvature, observed numerically, as τ_2 is modulated. We make this comparison by computing the analog of k_1 and k_2 , namely, the curvature of the free energy around the stored patterns.

Recall that the parameter λ in Eq. (10) controls the extent of unlearning. Adapting the calculation in Ref. [23], we show in the replica symmetric case that the curvature H of the free energy around the stable patterns is given by

$$H \equiv \left(1 + \frac{\lambda}{\Delta}\right) - \frac{\beta}{\Delta^2} \overline{1 - \tanh^2 \left[\frac{\beta}{\Delta} (a\xi + \sqrt{\alpha R} z) \right]}, \quad (23)$$

where β , R , a , ξ , Δ are the various order parameters of the system and $\bar{A}$ denotes averaging of A with respect to z weighted by $\exp(-z^2/2)$ (see Appendix H). Increasing the amount of unlearning, as characterized by λ , increases the curvature of the free energy around the encoded patterns. Figure 7(c) describes this dependence. Adding persistence to the system through τ_2 has a similar effect on the collective variable of the system, the firing rate y . As shown in Fig. 7(b), as τ_2 increases, the curvature of the two minima given by the harmonic approximation k_1 and k_2 also increase.

We end by acknowledging the differences in the details by which persistence is introduced in this section (in comparison with the previous sections). Nevertheless, the effective equations (21) bear some qualitative similarities to those found in systems with persistent dynamics [24]. In both cases, modifying the overall structure—whether through integration kernels or persistent noise—introduces inertia-like terms that alter the system's dynamics and lead to improved memory capacity.

VI. DISCUSSION

The information storage and processing ability of neurons is constantly modified by their inherent synaptic plasticity.

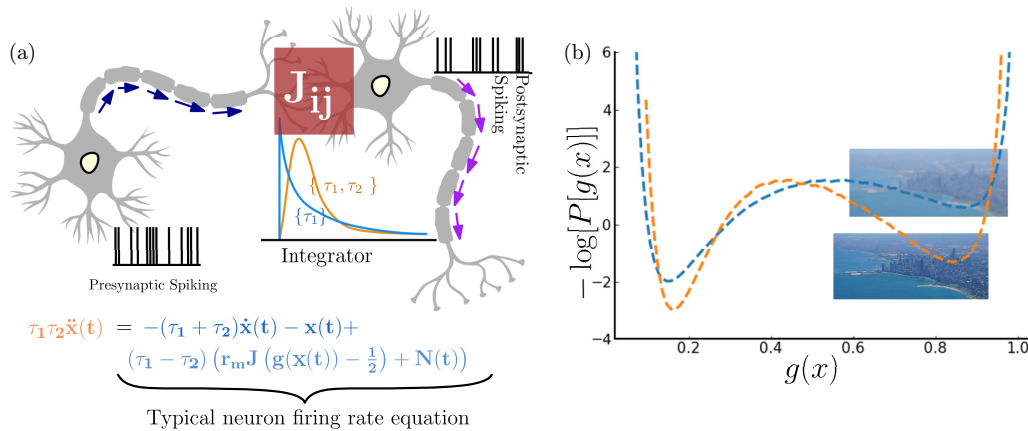


FIG. 6. (a) Temporal dynamics and memory in spiking neurons. The presynaptic signal is integrated at the synapse and helps in the firing of the postsynaptic neuron. The firing rates of the neurons is controlled in important values by the postsynaptic response function $f(t)$. The blue line denotes the case with just a single postsynaptic decay time constant, $\tau_1 \neq 0$ ($\tau_2 = 0$) (depicted by the blue integrator) whereas the orange line has two time constants, one for the rising phase and one for decay of the postsynaptic potential, $\tau_1 \neq 0, \tau_2 \neq 0$ (depicted by the orange integrator). In the equation for firing rate, $g(x)$ denotes an activation function to mimic the all-or-nothing response of neurons. (b) Introducing a new time constant for integration of the potential leads to a change in the dynamics of firing rates. The resultant steady states can have improved associative memory properties even though the synaptic connections J_{ij} remain unchanged. The distribution of the mean firing rates, in the energy-like form $-\log[P[g(x)]]$, is plotted for the cases of one (blue) and two (orange) time constants. The “Chicago skyline” image is a schematic for denoting a sharper image for a deeper energy basin and a fuzzier image for a shallower energy basin.

The interplay between synaptic dynamics and the memory storage and retrieval properties of neurons has been broadly studied in many seminal works [18]. Our work here suggests how phenomenology resembling synaptic plasticity can be achieved by simply introducing persistence in the dynamics of the neuron-like degrees of freedom.

The main analytical and numerical work presented in this paper considers a model system in which persistence is introduced in the dynamics of the neuronal degrees of freedom. This persistence is introduced as a correlation time in the noise driving the neuronal degrees of freedom. In previous work [14], we demonstrated how such persistent or active degrees of freedom can potentially lead to better memory recall

properties. In this work, we demonstrated that such persistent degrees of freedom create an information-storage landscape resembling one created by an unsupervised learning strategy commonly referred to as Hebbian unlearning or dreaming [35].

In the simple model in Fig. 6, persistence is modulated mainly by the dynamics of the postsynaptic response function f . This function mimics the receptor opening and closing probability at the postsynaptic membrane and can be influenced by a variety of factors, firing rate being one of them (Appendix C). We demonstrate that this mode of introducing persistence exhibits qualitative similarities with the modes discussed previously. Other modes for introducing persistence

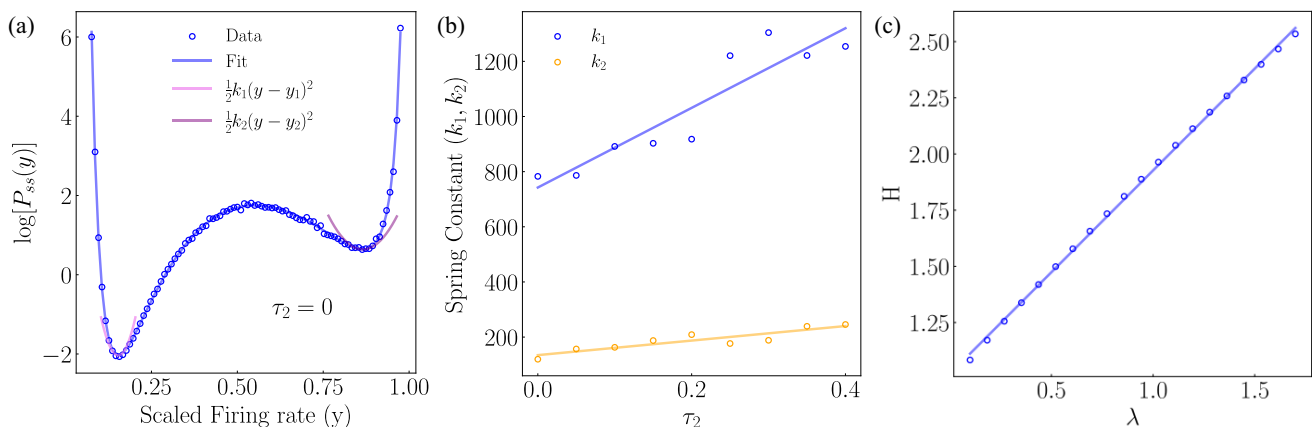


FIG. 7. (a) Distribution of the scaled firing rates for a two-neuron system for $\tau_2 = 0$. The scattered points are generated numerically and then fitted to a function, which is depicted by the solid blue curve. From the fitted curve, we calculate the curvature near the minima as a harmonic approximation. This provides us with two distinct values of spring constants, k_1 and k_2 . (b) k_1 and k_2 is plotted as a function of τ_2 and observe that it is an increasing function with respect to τ_2 . (c) H plotted as a function of the dreaming parameter λ . This shows how the curvature of the free-energy function f varies as a function of λ at the saddle points and the behavior is similar to that observed with firing rates.

include the coupling between neuronal and transcriptional dynamics, the spatial summation of postsynaptic potentials due to multiple neurons, the topology of the connections, etc. [36]. These might also lead to an improvement in the memory recall properties.

Finally, we anticipate that our findings may be applicable in other contexts, such as protein and chromatin folding problems, which can be posed as associative memory problems [37,38]. Due to the frustration implicit in these systems, the task of finding the desired ground structural state is highly nontrivial. Our work suggests how persistence in the relaxation process may help sculpt the energy landscape and aid in the process of finding the desired or programmed state. In protein folding, the persistence could arise, for example, due to the action of molecular chaperons that are known to consume ATP, and this may help sculpt the landscape and aid in the process of finding the desired or programmed state [39,40]. Together, our work simply provides a framework for understanding how nonequilibrium relaxational dynamics with multiple timescales—these occur routinely in biology—can be used to improve the effectiveness of memory recall.

ACKNOWLEDGMENTS

This work was mainly supported by funds from DOE BES Grant No. DE-SC0019765 to S.V. and M.D. A.K.B. acknowledges support from a fellowship from the Department of Chemistry at the University of Chicago. M.R. acknowledges support from the Department of Atomic Energy (India), under Project No. RTI4006, and the Simons Foundation (Grant No. 287975). M.R. and S.S. (JBR/2020/000015) acknowledge SERB, DST (India) for JC Bose Fellowships. We also acknowledge support from the National Science Foundation through the Physics Frontier Center for Living Systems (PHY-2317138). We acknowledge helpful discussions with Brent Doiron.

DATA AVAILABILITY

Data or codes supporting the findings of this paper are available from the corresponding author upon reasonable request.

APPENDIX A: INTEGRATE-AND-FIRE MODEL

Let us first define the integrators for the case with a single exponential fall phase ($f^{(1)}$) and the one with a rise and fall phase ($f^{(2)}$), and the corresponding operators for those integrators ($\mathcal{O}^{(1)}$ and $\mathcal{O}^{(2)}$):

$$f^{(1)}(t) = \frac{1}{\tau_1} \exp\left(-\frac{t}{\tau_1}\right), \quad (\text{A1})$$

$$f^{(2)}(t) = \frac{1}{\tau_1 - \tau_2} \left[\exp\left(-\frac{t}{\tau_1}\right) - \exp\left(-\frac{t}{\tau_2}\right) \right], \quad (\text{A2})$$

$$\mathcal{O}^{(1)} = \left(\frac{d}{dt} + \frac{1}{\tau_1} \right), \quad (\text{A3})$$

$$\mathcal{O}^{(2)} = \left(\frac{d}{dt} + \frac{1}{\tau_1} \right) \left(\frac{d}{dt} + \frac{1}{\tau_2} \right). \quad (\text{A4})$$

From (20), we can derive the equation of motion for the rate of firing of the neurons. We write down the equations of

motion for both the single exponential and the rise and fall integrator:

$$x_i + \theta_i = \sum_j J_{ij} f^{(1,2)} \circ (r_m y_j + \eta_j), \quad (\text{A5})$$

$$y_i = \frac{r_i}{r_m}, \quad x_i = g^{-1}(y_i), \quad p_i = \dot{x}_i = \frac{dx_i}{dt}, \quad (\text{A6})$$

$$\mathcal{O}^{(1)}(x_i + \theta_i) = \mathcal{O}^{(1)} \sum_j J_{ij} f^{(1)} \circ (r_m y_j + \eta_j), \quad (\text{A7})$$

$$\tau_1 \dot{x}_i = -x_i - \theta_i + r_m \sum_j J_{ij} \left(y_j - \frac{1}{2} \right) + N_i, \quad (\text{A8})$$

$$\mathcal{O}^{(2)}(x_i + \theta_i) = \mathcal{O}^{(2)} \sum_j J_{ij} f^{(2)} \circ (r_m y_j + \eta_j), \quad (\text{A9})$$

$$\tau_1 \tau_2 \dot{p}_i + (\tau_1 + \tau_2) p_i + x_i + \theta_i = r_m \sum_j J_{ij} y_j + N_i, \quad (\text{A10})$$

$$N_i(t) = \sum_j J_{ij} \eta_j. \quad (\text{A11})$$

It is important to note that $N_i(t)$ is a state-dependent noise function. We can choose the value of the threshold θ_i such that (A10) reduces to

$$\tau_1 \tau_2 \dot{p}_i + (\tau_1 + \tau_2) p_i + x_i = r_m \sum_j J_{ij} \left(y_j - \frac{1}{2} \right) + N_i. \quad (\text{A12})$$

APPENDIX B: CONNECTION TO ACTIVE MATTER DYNAMICS

The equation of motion for particles undergoing dynamics in the presence of active (persistent) noise is

$$\dot{x}_i = -\nabla_i \phi + \eta_i, \quad (\text{B1})$$

$$\tau \dot{\eta}_i = -\eta_i + \xi_i, \quad (\text{B2})$$

$$\langle \xi_i(t) \rangle = 0, \quad \langle \xi_i(t) \xi_j(t') \rangle = 2T \delta_{ij} \delta(t - t'), \quad (\text{B3})$$

where ϕ is the potential energy, τ is the timescale of active fluctuations η , and ξ is standard white noise. This leads to the noise η having correlations of the form

$$\langle \eta_i(t) \rangle = 0, \quad \langle \eta_i(t) \eta_j(t') \rangle = \frac{T}{\tau} \exp\left(-\frac{|t - t'|}{\tau}\right). \quad (\text{B4})$$

Denoting $\dot{x} = p$, we can rewrite this set of equations as [24]

$$\tau \dot{p}_i = -p_i - (1 + \tau p_k \cdot \nabla_k) \nabla_i \phi + \eta_i. \quad (\text{B5})$$

We will now try to connect (B5) to (A12). The connection between the two equations is the introduction of the rise-time scale τ_2 in (A12). Specifically, when analyzed around one of the stable firing rates, Eq. (A12) has an extra frictional component $\tau_2 p_i(t)$ similar to that encountered for a particle trapped in a harmonic well and driven by noise with a persistent degree of freedom [Fig. 6(b)]. Note that this analysis and analogy is limited to cases where $y(t)$ is fluctuating around one of its metastable points. In general, the noise function $N(t)$ is state-dependent, and an exact analogy is no longer possible.

The main models used in this work share some of the broad qualitative features of Eq. (A12) and active matter models

[(B5)]. The dynamics of the discrete Hopfield-like model in Sec. IV takes into account spin flips in the immediate history, qualitatively bringing in a new timescale into the picture. In order to have a more concrete connection to active matter, we need to have a coupling term like $p_k \cdot \nabla_k \nabla_i \phi$ as in (B5). This is possible in (A12) if τ_1 and τ_2 are functions of the rate of spiking, y . In the next appendix, we discuss how this is achieved in a biological setting.

APPENDIX C: RECEPTOR DYNAMICS AT THE POST SYNAPTIC MEMBRANE AND A TIMEESCALE DEPENDENT ON RATE OF FIRING

Synaptic transmission occurs when a spike arrives at the presynaptic terminal and leads to the release of neurotransmitters inside the synaptic cleft (the region between the two neurons at the synapse). These neurotransmitters are then absorbed by the postsynaptic neuron through various channel proteins [41]. The synaptic conductance of the current is thus proportional to the product of two terms, $P = P_{\text{rel}}P_s$, where P_{rel} is the release probability of neurotransmitters given an action potential arrives at the terminal and P_s is the probability that the postsynaptic channel opens given neurotransmitters were released at the cleft. We assume a constant P_{rel} and focus mainly on P_s in what follows. P_s can be modeled as a directly activated receptor channel where the transmitter binds directly with the channel to open it and then unbinds after a certain amount of time to close it [41]. The following equation can be used to describe P_s ,

$$\frac{dP_s(t)}{dt} = \alpha_s(r, t)[1 - P_s(t)] - \beta_s P_s(t). \quad (\text{C1})$$

Here, β_s is the closing rate of the channel and is usually assumed to be a constant. The opening rate $\alpha_s(r, t)$ depends on the concentration of the transmitters, which changes with the spiking rate r and time t .

Under certain conditions, we can express P_s in terms of the convolution kernel for integration, $f(t)$ in Appendix A. To see this, consider (C1) with $\alpha_s(t) = P_0 \sum_j \delta(t - t^j)$, where t^j is the time of spike j , and β_s is constant. In this limit, after every spike, we can write, $P_s \rightarrow P_s + P_0$, and between spikes, $P_s(t) = P_s(0)e^{-\beta t}$. Thus, under the assumption that $P_0 \ll 1$, the functional form of P_s after multiple spikes is given as, $P_s(t) = P_0 \sum_j e^{-\beta(t-t^j)}$. This functional form is nothing but the convolution of the spike train with the kernel, $f = e^{-t/\tau}$ with $\beta = 1/\tau$.

In Appendix C1 we describe “Model 1,” where we similarly show, for an appropriate choice of $\alpha_s(r, t)$, that P_s can be approximated by the convolution of the spike train $R(t)$ with the kernel, $f = e^{-t/\tau_1} - e^{-t/\tau_2}$. In Appendix C2, we discuss a different model, “Model 2,” having a different functional form for $\alpha_s(r, t)$. For both models, we calculate the steady-state probabilities, $P_s^{(\text{ss})}$, and equate them. By doing so, we find an effective description of Model 2 in terms of Model 1 parameters. Then we can use Model 1 with these effective parameters and connect it to the convolution kernel, $f = K(e^{-t/\tau_1} - e^{-t/\tau_2})$. For further simplification, we also assume that the spikes are uniformly distributed in time, i.e., $t^j = j/r$ for firing rate r .

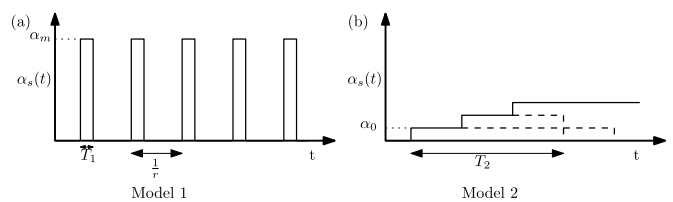


FIG. 8. Schematic for $\alpha_s(t)$ profiles for Model 1 and Model 2. The dashed lines in panel (b) denote the course of $\alpha_s(t)$ due to the first and second spikes.

1. Model 1

In Model 1, after every presynaptic spike, the postsynaptic receptors display a very fast Markovian behavior of switching between “open” and “closed” states. This can be expressed as an instantaneous rise in $\alpha_s(t)$ to its maximum value, $\alpha_m \cdot \alpha_s(t)$ remains constant at that value for a specific duration, T_1 before instantaneously dropping to zero [41]. Thus $\alpha_s(t)$ is a rectangular pulse function. Mathematically, it can be expressed as

$$\alpha_s(t) = \mathbb{T} \left(\alpha_m \sum_f [\theta(t - t^f) - \theta(t - t^f - T_1)], \alpha_m \right), \quad (\text{C2})$$

where θ is the Heaviside function, t^f are the spiking times and $\mathbb{T}(a, b)$ is the threshold function which ensures that the function does not exceed b . A schematic of this form of $\alpha_s(t)$ is plotted in Fig. 8(a). This generates a P_s profile for Model 1 for which an excellent approximation is $f \circ R$ with the convolution kernel, $f = K(e^{-t/\tau_1} - e^{-t/\tau_2})$, when a single spike is fired. The corresponding τ_1 and τ_2 are given by $\tau_1 = 1/\beta$ and $\tau_2 \sim 1/(\alpha_m + \beta)$. The exact functional form of τ_2 is pretty complicated, but it roughly goes as the inverse of the rate of rise of P_s , i.e., α_m .

Using (C2), we can calculate the steady-state value of P_s . Let us say P_s was $P_s^{(\text{ss})}$ just before the arrival of a presynaptic spike. After the spike arrives, it rises to a value P_s^{max} in time T_1 and then decays for a time $1/r - T_1$ (uniform spiking with rate r leads to interspike intervals of length $1/r$) before the arrival of a new spike. For this model, $T_1 \ll 1/r$. This limit is to ensure that $\alpha_s(t)$ decays to zero between two consecutive spikes. We want the rise in P_s and its decay to compensate for one another:

$$\frac{dP_s}{dt} = \alpha_m(1 - P_s) - \beta_s \text{ for } 0 < t < T_1, \quad (\text{C3})$$

$$P_s(t = T) = P_s^{(\text{ss})} e^{-(\alpha_m + \beta)T_1} + \frac{\alpha_m}{\alpha_m + \beta} [1 - e^{-(\alpha_m + \beta)T_1}], \quad (\text{C4})$$

$$\frac{dP_s}{dt} = -\beta P_s \text{ for } T_1 < t < \frac{1}{r}, \quad (\text{C5})$$

$$P_s \left(t = \frac{1}{r} \right) = P_s(t = T) e^{-\beta(\frac{1}{r} - T_1)} = P_s^{(\text{ss})} \quad (\text{C6})$$

$$\Rightarrow P_s^{(\text{ss}, 1)} = \frac{\alpha_m}{\alpha_m + \beta} \frac{e^{\beta T_1} - e^{-\alpha_m T_1}}{e^{\beta/r} - e^{-\alpha T_1}}. \quad (\text{C7})$$

2. Model 2

Now we discuss Model 2. In this model, the postsynaptic receptors exhibit cooperative behavior with every presynaptic spike, facilitating the opening of additional receptors in the postsynaptic membrane. Every presynaptic spike opens a fraction α_0 of postsynaptic receptors and the timescale of this cooperative process is T_2 . In Model 2, $T_2 > 1/r$. This limit is to ensure that cooperativity persists for longer than the interspike interval. Mathematically, the α_s function and its steady-state value of α_s can be expressed as

$$\alpha_s(r, t) = \alpha_0 \sum_{f=1}^n [\theta(t - t^f) - \theta(t - t^f - T_2)], \quad (\text{C8})$$

$$\alpha_s^{(ss)} = \alpha_0 r T_2. \quad (\text{C9})$$

A schematic of $\alpha_s(t)$ for Model 2 is plotted in Fig. 8(b). With this form of α_s , the final steady-state probability $P_s^{(ss),2}$ is given by

$$P_s^{(ss,2)} = \frac{\alpha_0 r T_2}{\alpha_0 r T_2 + \beta}. \quad (\text{C10})$$

3. Equating the steady-state probabilities

Now we equate the steady-state probabilities $P_s^{(ss)}$ from both the models in (C7) and (C10). We do it since we want a description of Model 2 in terms of the parameters of Model 1, i.e., we want an “effective” α_m and β for Model 2. The procedure to extract these effective parameters is as follows:

$$\frac{\alpha_m}{\alpha_m + \beta} \frac{e^{\beta T_1} - e^{-\alpha_m T_1}}{e^{\beta/r} - e^{-\alpha T_1}} = \frac{\alpha_0 r T_2}{\alpha_0 r T_2 + \beta}. \quad (\text{C11})$$

We assume that the rate of decay, β is the same in both models and we also fix the duration, T_1 and T_2 . Then we solve for α_m in (C11). This is a transcendental equation, and we use numerical tools to compute α_m . Now we use these effective parameters to obtain τ_1 and τ_2 for the convolution kernel f . We can do so because Model 1 is equivalent to f for a single spike. After solving for α_m , we use this value of α_m , β , r and T_1 to construct a new $P_s(t)$ profile for a single spike. Now, we fit this profile to the function $f(t) = B[\exp(-t/\tau_1) - \exp(-t/\tau_2)]$ and extract τ_1 and τ_2 . We repeat this procedure, keeping the variables β , T_1 , T_2 , and α_0 fixed and varying only the rate of firing r . Figure 9 shows how τ_2 varies as a function of the firing rate r in this equivalent description. This $\tau_2(y)$ can be replaced in (A12), and that leads to

$$\tau_1 \tau_2(y) \dot{p}_i + [\tau_1 + \tau_2(y)] p_i + x_i = r_m \sum_j J_{ij} \left(y_j - \frac{1}{2} \right) + N_i. \quad (\text{C12})$$

The $\tau_2(y)p_i$ term in (C12) gives us an active matter-like flavor where the “position” degrees of freedom (y_i) are now coupled to the “momentum” degrees of freedom (p_i), which in turn can give rise to novel phenomena.

APPENDIX D: DREAMING

We first show that the Hebbian unlearning rule (8) can be written as the local learning rule of (9). We then show that this

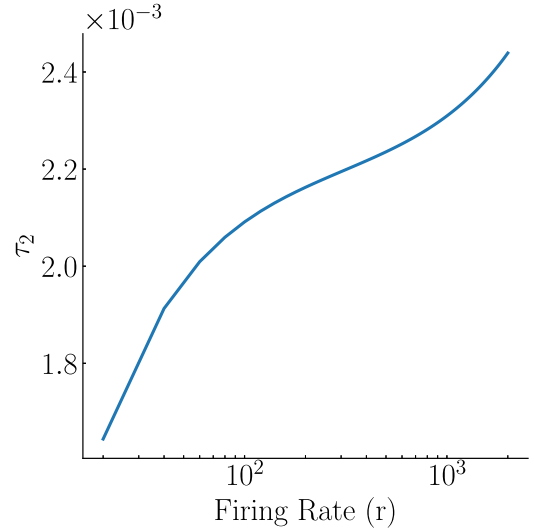


FIG. 9. τ_2 increases as the rate of firing increases. With the increase in the rate of firing the steady-state probability, $P_s^{(ss)}$ increases. In order to have a late onset of decay, the rise-time, τ_2 , has to increase. The parameters used for this simulation: $\alpha_0 = 2$, $T_2 = 0.1$, $T_1 = 5 \times 10^{-4}$, $\beta = 10$.

local rule can lead to the nonlocal pseudoinverse rule of (10) when repeated over and over again. These results were shown previously in Refs. [23,42].

The evolution of the connectivity matrix following the procedure of (8) can be expressed mathematically as

$$J_{ij}(t+1) = J_{ij}(t) - \epsilon \langle h_i^*(t) h_j^*(t) \rangle. \quad (\text{D1})$$

For the Hamiltonian in (5), the process of dreaming can be mathematically expressed as

$$J_{ij}(t+1) = J_{ij}(t) - \epsilon \sum_{pq} \langle J_{ip} J_{jq} f(\sigma_p) f(\sigma_q) f'(\sigma_i) f'(\sigma_j) \rangle \quad (\text{D2})$$

$$= J_{ij}(t) - \epsilon \sum_{pq} J_{ip} J_{jq} \langle f(\sigma_p) f(\sigma_q) f'(\sigma_i) f'(\sigma_j) \rangle \quad (\text{D3})$$

$$= J_{ij}(t) - \epsilon \sum_{pq} J_{ip} J_{jq} \delta_{pq} |f(\sigma_p)|^2 f'(\sigma_i) f'(\sigma_j) \quad (\text{D4})$$

$$\approx J_{ij}(t) - \epsilon \sum_p J_{ip} J_{jp} f'(\sigma_i) f'(\sigma_j), \quad (\text{D5})$$

which is equivalent to (9). Here we have used the fact that σ_p and σ_q are drawn independently from the uniform distribution $\mathcal{U}[-b, b]$ where $b \gg f^{-1}(1)$. f is odd, i.e., $f(\sigma) = -f(-\sigma)$. As one samples σ across $[-b, b]$, the number of configurations near $f(\sigma) \approx 1$ far outnumbers the configurations away from 1. Thus, we have also approximated $|f(\sigma)| \approx 1$ to simplify calculations. Equation (9) is a local learning rule, since J_{ij} depends only on σ_i and σ_j .

Similarly, for the Hamiltonian from persistent dynamics, we have

$$H(t+1) = H(t) + c|\nabla H(t)|^2 \quad (\text{D6})$$

$$= - \sum_{ij} J_{ij}(t) f(\sigma_i) f(\sigma_j) + c \sum_k \sum_i J_{ki}(t) \times f(\sigma_i) f'(\sigma_k) \sum_j J_{kj}(t) f(\sigma_j) f'(\sigma_k) \quad (\text{D7})$$

$$= - \sum_{ij} \left[J_{ij}(t) - c \sum_k J_{ki}(t) J_{kj}(t) |f'(\sigma_k)|^2 \right] \times f(\sigma_i) f(\sigma_j), \quad (\text{D8})$$

$$J_{ij}(t+1) = J_{ij}(t) - c \sum_k J_{ik} J_{jk} |f'(\sigma_k)|^2, \quad (\text{D9})$$

with $c = \tau T_a / T_{\text{eff}}$.

For $f'(\sigma) = 1$, (9) can be recast as

$$\frac{\delta J}{\delta t} = -\epsilon J^2, \quad (\text{D10})$$

$$\Rightarrow J(d) = \frac{J(0)}{1 + \epsilon t J(0)}, \quad J(0)_{ij} = \frac{1}{N} \sum_{\mu} \xi_i^{\mu} \xi_j^{\mu}, \quad (\text{D11})$$

$$J_{ij} = \frac{1}{N} \sum_{\mu\nu} \xi_i^{\mu} (1 + \lambda^2 C)_{\mu\nu}^{-1} \xi_j^{\nu}, \quad (\text{D12})$$

$$C_{\mu\nu} = \frac{1}{N} \sum_i \xi_i^{\mu} \xi_i^{\nu}, \quad \lambda^2 = \epsilon d. \quad (\text{D13})$$

(D12) can be obtained from its previous step using the Woodbury matrix identity. The connectivity matrix of (D12) leads to the Hamiltonian of (10). Thus the partition function with this “dreamt” Hamiltonian is given by

$$Z = \sum_{\sigma} \exp \left(\frac{\beta}{N} \sum_{\mu\nu ij} \sigma_i \xi_i^{\mu} (1 + \lambda^2 C)_{\mu\nu}^{-1} \xi_j^{\nu} \sigma_j \right). \quad (\text{D14})$$

Equation (D12) and thus (10) correspond to a nonlocal learning rule because the $(1 + \lambda C)^{-1}$ part of J_{ij} depends on ξ_k^{ν} for $k \neq i, j$, as can be seen from a Taylor expansion of $(1 + \lambda C)^{-1}$ for small λ .

APPENDIX E: SIMULATION DETAILS FOR DREAMING

Here, we describe a simulation to indirectly verify the equivalence of the connectivity matrices obtained, respectively, by Hebbian unlearning (9) [which is the rule for a single update, where (11) is the connectivity matrix obtained after multiple updates] and persistent dynamics (14) when $\epsilon d = \tau T_a / T_{\text{eff}}$. As mentioned in Sec. III, directly verifying the equivalence of (9) and (14) would be numerically difficult because they differ from each other by a perturbatively small amount. In the simulation, which is described in more detail below, we first iteratively update an initial connectivity matrix according to a rule based on either (9) or (14). Then, we simulate pattern retrieval with the new connectivity matrix. We take the spins to be continuous.

For each simulation, we begin by drawing p patterns $\vec{\xi}^1, \dots, \vec{\xi}^p \in \{\pm 1\}^N$ from the Rademacher distribution and constructing the initial connectivity matrix J_{ij} according to

the Hebb rule [(2)]. Next, we iteratively update J_{ij} using a dreaming procedure based on either Hebbian unlearning [(9)] or persistent dynamics [(14)]. We carry out d dreaming updates, each one described as follows:

(1) Randomly generate a spin state that is offset from a pattern, $\sigma_i = \xi_i^{\mu} + \Delta\sigma_i$, where each pattern (index μ) is chosen with equal probability, and the offsets $\Delta\sigma_i \sim \mathcal{N}(0, 1)$ are chosen from the standard normal distribution.

(2) For Hebbian unlearning-type dreaming, update the connectivity matrix as [(9)]

$$J_{ij} \rightarrow J_{ij} - \epsilon \sum_p J_{ip} J_{jp} f'(\sigma_i) f'(\sigma_j). \quad (\text{E1})$$

For persistent dynamics-type dreaming, update the connectivity matrix as [(14)]

$$J_{ij} \rightarrow J_{ij} - \frac{1}{d} \frac{\tau T_a}{T_{\text{eff}}} \sum_p J_{ip} J_{jp} |f'(\sigma_p)|^2. \quad (\text{E2})$$

With the new J_{ij} , we simulate pattern retrieval. We attempt to retrieve the first ten patterns, $\vec{\xi}^{\mu}$ for $\mu = 1, \dots, 10$. For the retrieval of pattern μ , the spins are initialized near the pattern, $\sigma_i(0) = \xi_i^{\mu} + \delta\sigma_i$, where $\delta\sigma_i \sim \mathcal{N}(0, 0.1)$ is a small random offset. The spins then evolve as

$$\partial_t \sigma_i = J_{ij} f(\sigma_j), \quad (\text{E3})$$

where f is the activation function. This is inspired from Ref. [35], Eq. (5). Unlike Eq. (5) of Ref. [35], we do not have a $-\sigma_i$ term in the right-hand side (RHS) of our equation of motion. This term bounds the spin values from blowing up, but we do not need it in our case since we only consider retrieval dynamics near the patterns, and we deal with $f(\sigma)$ which is already bounded by $[-1, +1]$. After evolving the spins for 11 000 time steps of $\Delta t = 0.01$, we record the overlap with the target pattern, $\frac{1}{N} \vec{\sigma} \cdot \vec{\xi}^{\mu}$.

We carry out the simulations for ten different systems, where each system has a distinct set of patterns. The results in Fig. 1 are an average over these ten systems.

APPENDIX F: RESTRICTED BOLTZMANN MACHINES FOR NUMERICAL EXPERIMENTS

In this section, we look at each of the steps in detail. We follow Ref. [32] but make changes for the three-layered RBM architecture.

1. Data generation

First, we choose the pattern loading of the system (α). We choose two temperatures for sampling with passive dynamics (T_p) and with persistent dynamics (T_a). These temperatures and α are chosen such that they are regions where the pattern configurations are the global minima.

2. Training the RBM

The RBM architecture consists of N visible nodes corresponding to the N spins of the system, M hidden nodes where M is the number of patterns stored in the original system, and N hidden nodes to account for the “pseudoinverse” structure of the connectivity matrix. The RBM weights are initialized

as

$$W_i^\mu = \xi_i^\mu + z, \quad z \sim \mathcal{N}(0, 1). \quad (\text{F1})$$

In this section, for convenience, we denote the temperature at which the RBM samples as T instead of T_{RBM} (and similarly, β for β_{RBM}). It should be noted that β_{RBM} is a parameter of the model, not the inverse temperature of the simulations. For the passive case, we know the ground truth that β_{RBM} is equal to β of the simulations.

For each sample of the data, the visible nodes are set to the sample configuration $\sigma_i = \sigma_i^{\text{data}}$. Now given the data, we generate a configuration for the hidden nodes (z and ϕ). We can use the conditional probabilities for this purpose:

$$P(z, \phi | \sigma) = \frac{P(z, \phi, \sigma)}{P(\sigma)} = \frac{\exp(-\beta H)}{\int Dz D\phi \exp(-\beta H)} \quad (\text{F2})$$

$$\begin{aligned} &= \exp \left[-\frac{\beta}{2} \left(\phi_i - \frac{i\lambda}{\sqrt{N}} z_\mu W_i^\mu \right)^2 \right] \\ &\times \exp \left[-\frac{\beta}{2} \left(z_\mu - \frac{W_i^\kappa \sigma_i}{\sqrt{N}} A_{\kappa\mu}^{-1} \right) \right. \\ &\left. \times A_{\mu\nu} \left(z_\nu - \frac{W_j^\chi \sigma_j}{\sqrt{N}} A_{\chi\nu}^{-1} \right) \right], \end{aligned} \quad (\text{F3})$$

$$A_{\mu\nu} = (\delta_{\mu\nu} + \lambda^2 C_{\mu\nu}), \quad C = \frac{1}{N} W_i^\mu W_i^\nu. \quad (\text{F4})$$

Now, the configuration of the hidden nodes is generated as

$$\mathbf{z} \sim \mathcal{N} \left(\frac{\mathbf{W} \sigma \mathbf{A}^{-1}}{\sqrt{N}}, T \mathbf{A}^{-1} \right), \quad (\text{F5})$$

$$\phi \sim \mathcal{N} \left(\frac{\lambda}{\sqrt{N}} \mathbf{z} \mathbf{W}, T \mathbf{I} \right). \quad (\text{F6})$$

Now, we can regenerate a configuration of the visible nodes using $P(\sigma | z, \phi)$,

$$P(\sigma | z, \phi) = \frac{P(z, \phi, \sigma)}{P(z, \phi)} = \frac{\exp(-\beta H)}{\sum_\sigma \exp(-\beta H)} \quad (\text{F7})$$

$$= \frac{1}{1 + \exp \left(-2\beta \frac{W_i^\mu}{\sqrt{N}} z_\mu \sigma_i \right)}. \quad (\text{F8})$$

Thus, we regenerate the visible nodes as

$$\sigma'_i = \text{sgn} \left(\frac{1}{1 + \exp(-2\beta \sum_\mu W_{\mu i} z_\mu)} - r \right), \quad r \sim U(0, 1). \quad (\text{F9})$$

We again recompute the hidden nodes from these recomputed σ'_i as

$$\mathbf{z}' \sim \mathcal{N} \left(\frac{\mathbf{W} \sigma' \mathbf{A}^{-1}}{\sqrt{N}}, T \mathbf{A}^{-1} \right), \quad (\text{F10})$$

$$\phi' \sim \mathcal{N} \left(\frac{\lambda}{\sqrt{N}} \mathbf{z}' \mathbf{W}, T \mathbf{I} \right). \quad (\text{F11})$$

The weight matrix and λ can now be updated using contrastive divergence. Essentially, what the RBM is trying to do is reproduce the same distribution as the one provided to it from the real-world model. If we denote the real data distribution as

$Q(\sigma)$ and the model as $P(\sigma; \theta)$ where θ are the parameters we want to optimize.

$$D_{KL}(Q||P) = \sum_\sigma Q(\sigma) \ln \frac{Q(\sigma)}{P(\sigma)}, \quad (\text{F12})$$

$$\frac{D_{KL}(Q||P)}{\partial \theta} = - \sum_\sigma Q(\sigma) \partial_\theta \ln P(\sigma; \theta) \quad (\text{F13})$$

$$= - \sum_\sigma Q(\sigma) \partial_\theta \ln \frac{e^{-\tilde{H}(\sigma; \theta)}}{\sum_{\sigma'} e^{-\tilde{H}(\sigma'; \theta)}} \quad (\text{F14})$$

$$= - \sum_\sigma Q(\sigma) \left[- \frac{\tilde{H}(\sigma; \theta)}{\partial \theta} + \sum_{\sigma'} \frac{e^{-\tilde{H}(\sigma'; \theta)}}{\sum_{\sigma''} e^{-\tilde{H}(\sigma''; \theta)}} \frac{\partial \tilde{H}(\sigma'; \theta)}{\partial \theta} \right] \quad (\text{F15})$$

$$= \left\langle \frac{\partial \tilde{H}(\sigma; \theta)}{\partial \theta} \right\rangle_{\text{data}} - \left\langle \frac{\partial \tilde{H}(\sigma; \theta)}{\partial \theta} \right\rangle_{\text{model}}. \quad (\text{F16})$$

The trainable parameters for our system are the RBM weights W , the dreaming parameter λ and the (effective) inverse temperature β ; and $\tilde{H} = \beta \left[\frac{z_\mu^2}{2} + \frac{\phi_i^2}{2} - z_\mu \frac{\xi_i^\mu}{\sqrt{N}} (\sigma_i + i\lambda \phi_i) \right]$. Using this, we have the relaxation equations for the trainable degrees of freedom,

$$\frac{\partial W_i^\mu}{\partial t} = - \frac{\partial D_{KL}(Q||P)}{\partial W_i^\mu} \quad (\text{F17})$$

$$= \left\langle \frac{\partial \tilde{H}(\sigma; \theta)}{\partial \theta} \right\rangle_{\text{model}} - \left\langle \frac{\partial \tilde{H}(\sigma; \theta)}{\partial \theta} \right\rangle_{\text{data}} \quad (\text{F18})$$

$$= \frac{\beta}{\sqrt{N}} [z_\mu (\sigma_i + i\lambda \phi_i) - z'_\mu (\sigma'_i + i\lambda \phi'_i)], \quad (\text{F19})$$

$$\frac{\partial \lambda}{\partial t} = i \frac{\beta}{\sqrt{N}} \sum_{\mu, i} W_i^\mu (z_\mu \phi_i - z'_\mu \phi'_i), \quad (\text{F20})$$

$$\begin{aligned} \frac{\partial \beta}{\partial t} &= \frac{1}{\sqrt{N}} \sum_{\mu, i} W_i^\mu [z_\mu (\sigma_i + i\lambda \phi_i) - z'_\mu (\sigma'_i + i\lambda \phi'_i)] \\ &+ \sum_\mu \frac{z_\mu^2 - z_\mu'^2}{2} + \sum_i \frac{\phi_i^2 - \phi_i'^2}{2}. \end{aligned} \quad (\text{F21})$$

At each step of the update, the columns of the W matrix are regularized such that their norm is $\sqrt{N}$.

3. Different modes of training

For data generated using equilibrium dynamics, the patterns ξ_i^μ , the inverse temperature β , and $\lambda = 0$ are the ground truths of the system, whereas for the data generated using persistent dynamics, only the patterns are the ground truths and one needs to infer the effective temperature and λ . We train the system using three different modes:

(1) Learn all ξ , λ , and β . If we initialize the parameters at a small distance away from the actual parameters, the RBM gets stuck in a set of parameters which are not the ground truths of the system. We can characterize this well because we know the ground truth ξ , λ and β for the passive simulations exactly. The RBM fails to reach the target β and λ .

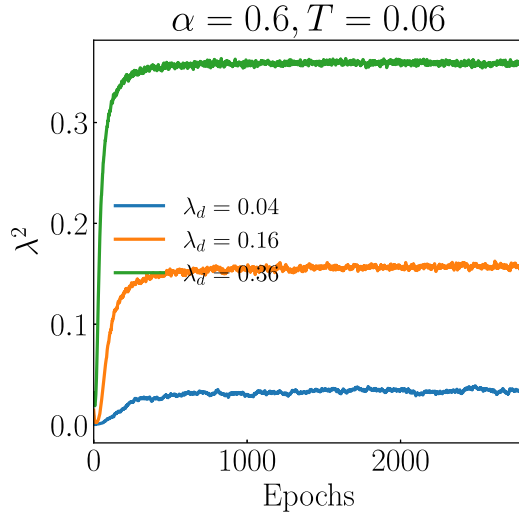


FIG. 10. The inferred value of λ matches closely with the value used for generating the data. The details are provided in Appendix G 1.

(2) Fix the β as the inverse temperature used to generate data in both persistent and passive simulations. Learn ξ and λ . This setting is not ideal for persistent simulations because we do not know the effective temperature for persistent simulations.

(3) Fix the weights of the RBM as the patterns ξ . Learn the inverse temperature β and λ . This is the ideal setting for comparing persistent and passive simulations. The patterns ξ are the ground truths for both passive and persistent simulations, so constraining the RBM weights using these is a valid thing to do. Since the effective temperature and λ for persistent simulations are unknown, whereas that for passive simulations is known, this mode is ideal for comparing the learning parameter λ .

For generating Figs. 4(b) and 4(c), we use mode (3) for inferring the parameters. In the next section, we benchmark our RBM inference procedures.

APPENDIX G: SIMULATION DETAILS AND CHECKS

We perform all simulations and inference with $N = 50$ spins. The loading capacity α is varied from 0.06 to 0.1. The temperature for generating data is varied from $T = 0.4$ to $T = 0.8$. The degree of activity is varied between $a = 0.0$ to $a = 0.8$ with $a = 0.0$ corresponding to the passive case. The learnable parameters are λ , T_{RBM} , and the patterns $\xi_i^\mu (W_i^\mu)$.

1. Benchmarking the model with data generated from a pseudo-inverse Hopfield Hamiltonian

In this benchmarking procedure, T_{RBM} is set to T since the data are generated using passive simulations at a specific temperature T using the pseudoinverse Hamiltonian, and we know exactly that $T_{\text{RBM}} = T$. The system infers the λ and the patterns ξ_i^μ . To be concise, only the inferred λ is plotted. The three plots in Fig. 10 are for data generated using three different values of λ as it is varied between 0.2 to 0.6. This shows that our system is capable of inferring the extent of dreaming λ , accurately.

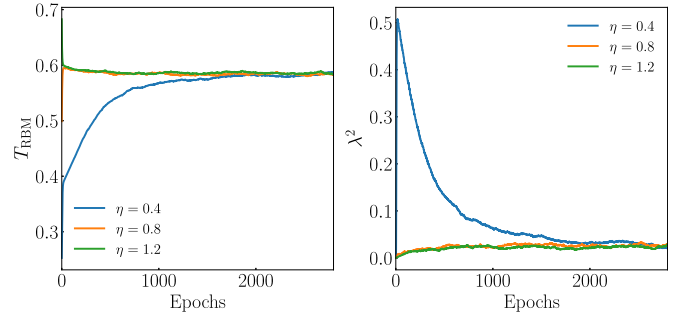


FIG. 11. The system learns the T_{RBM} from data when initialized at $0.4T$, $0.8T$, and $1.2T$. Also, the λ learned is very close to zero, as it should be. The details about the simulations are provided in Appendix G 2.

2. Varying initial conditions

For this set of simulations, we fix the weights of the RBM to be equal to the patterns, $W_i^\mu = \xi_i^\mu$ and learn only the λ and T_{RBM} . The initial conditions for learning T_{RBM} are varied from $T_{\text{RBM}}^{\text{init}} = 0.4T$ to $1.2T$. The data for plots in Fig. 11 are generated using $\alpha = 0.06$, $T = 0.6$, $a = 0$, i.e., these are passive simulations for which we know that $T_{\text{RBM}} = T$ and $\lambda = 0$. This shows that our system is robust to initial conditions of T_{RBM} and infers it with good accuracy while inferring the corresponding λ as well. We expect the small deviation from actual values to vanish with increased sampling of states and the number of training epochs.

3. Vary the T_{RBM} and run various persistent and passive simulations

For the set of inference in Fig. 12, we fix T_{RBM} and learn λ and the weights W_i^μ . T_{RBM} is set to ηT , where η varies

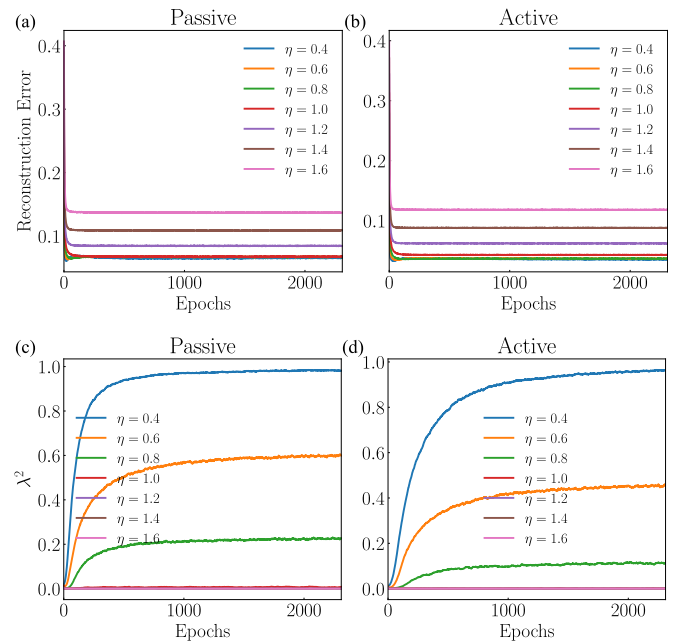


FIG. 12. The λ corresponding to the saturation of RCE is nonzero for the persistent (active) case, whereas it is zero for the passive case. The details of the simulations are provided in Appendix G 3.

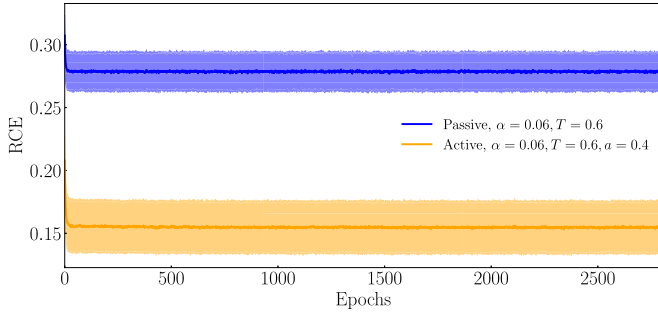


FIG. 13. Reconstruction error as a function of epochs for $\alpha = 0.06$, $T = 0.6$ for both persistent (active) and passive cases.

from 1.6 to 0.4 in steps of 0.2 and T is the temperature at which the data has been generated and is set to $T = 0.4$. Both passive and persistent data are generated with pattern loading, $\alpha = 0.06$. For persistent data, $a = 0.4$. The reconstruction error (RCE) is plotted in Figs. 12(a) and 12(b) and the corresponding λ values are plotted in Figs. 12(c) and 12(d). As T_{RBM} is decreased from $1.6T$ to $0.4T$, the steady-state RCE value goes down but saturates at some point, i.e., does not go down any further. This value of T_{RBM} is closest to the actual temperature used to generate the data. For the passive data, RCE saturates at $T_{\text{RBM}} = 1.0T$, and the corresponding λ value is close to zero. This is expected. For persistent simulations, RCE saturates at $T_{\text{RBM}} = 0.8T$ indicating that $0.8T$ is closest to the notional temperature for the persistent case, and the corresponding λ is nonzero. Since there does not exist a notion of temperature for persistent simulations, this is a good way to understand that when we try to approximate the distribution generated by persistent simulations with our ansatz, the effective temperature is different from the temperature parameter used to carry out the simulations. Thus, in order to compare the passive and persistent data on an equal footing, we must learn T_{RBM} along with other parameters. Fixing T_{RBM} might lead to incorrect inference in the persistent case.

4. Reconstruction error and pseudolog-likelihood loss (PLL) for Fig. 4 of main text

The following two Figs. 13 and 14 correspond to the generative performance of the inference procedure we employed for Fig. 4. Lower reconstruction error means better performance.

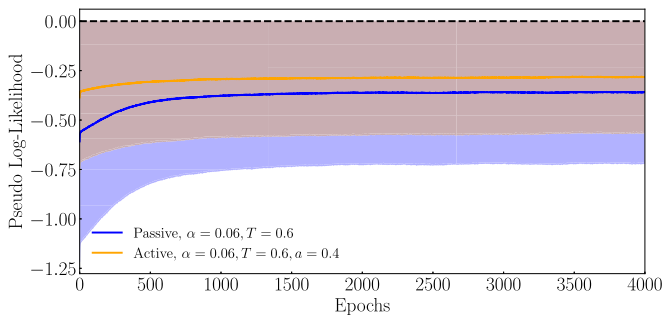


FIG. 14. Pseudolog-likelihood loss as a function of epochs for $\alpha = 0.06$, $T = 0.6$ for both persistent (active) and passive cases.

The other measure we use is the pseudolog-likelihood function. A likelihood function provides a quantitative measure of how well a statistical model explains the given observations. In our case, the observations are the configurations generated using either passive or active dynamics and the model parameters correspond to the weights and temperature, and unlearning rate of the RBM used to model it. Computing the likelihood for the RBM model using the dataset is tough because computing the partition function is computationally intractable. Thus, we resort to a different measure that is computationally tractable, the pseudolog-likelihood. The pseudolog-likelihood increases with training and then saturates for both persistent and passive cases. For a set of random variables, $\vec{\sigma} = \{\sigma_i\}$, the pseudolog-likelihood of $\vec{\sigma}$ is defined as $\log L(\vec{\theta})$ where $L(\vec{\theta}) = P_{\vec{\theta}}(\sigma_i = k_i | \sigma_j = k_j \forall j \neq i)$ and $\vec{\theta}$ are the parameters of the model that best describes the data which we are trying to estimate. In the case of the RBM, the pseudolog-likelihood loss (PLL) computes the sum of the log of conditional probabilities of a spin given all the other spins in the configuration for the given model. It is given as

$$PLL(\vec{\sigma}) = \sum_{i=1}^N \log P(\sigma_i | \vec{\sigma}_{\setminus i}) \quad (G1)$$

$$= \sum_{i=1}^N \log \left(\frac{e^{-\beta E_0}}{e^{-\beta E_0} + e^{-\beta E_i}} \right) \quad (G2)$$

$$= - \sum_{i=1}^N \log (1 + e^{-\beta(E_i - E_0)}). \quad (G3)$$

Here, $\vec{\sigma}_{\setminus i}$ denotes all the spins except spin i , E_0 is the energy of the original configuration, E_i is the energy of the configuration with spin i flipped, and β is the inverse temperature. The Boltzmann form for the probabilities has been used for this computation. Intuitively, the PLL measures how likely the model is to produce a configuration from the dataset if all but one of the spins are held fixed. The PLL can approach zero but never exceed it. The PLL should converge by the end of training.

We train until the PLL saturates and continue training till the other learnable parameters, namely, the weights λ of the network and the temperature T , also converge.

APPENDIX H: FREE ENERGY FOR THE PSEUDOINVERSE HAMILTONIAN AND HESSIAN CALCULATION

The pseudoinverse Hamiltonian and the overlap parameter m are given as

$$H = \frac{1}{N} \sum_{ij\mu\nu} \sigma_i \xi_i^\mu (1 + \lambda C)_{\mu\nu}^{-1} \xi_j^\nu \sigma_j, \quad (H1)$$

$$C_{\mu\nu} = \frac{1}{N} \sum_i \xi_i^\mu \xi_i^\nu, \quad (H2)$$

$$m = \frac{1}{N} \sum_i \xi_i^1 \langle \sigma_i \rangle. \quad (H3)$$

From Ref. [23], the free energy for this system in the replica symmetric case is given as $f(a, Q_0, Q, R, \Delta)$,

$$\begin{aligned} f(a, Q_0, Q, R, \Delta) &= \frac{1}{2} \left(1 + \frac{\lambda}{\Delta} \right) a^2 + \frac{\alpha}{2\beta} \left(\log [1 - \beta(Q_0 - Q)] \right. \\ &\quad \left. - \frac{\beta Q}{1 - \beta(Q_0 - Q)} \right) + \frac{1}{2} \alpha \beta R (Q_0 - Q) \\ &\quad + \frac{\Delta - 1}{2\lambda} Q_0 + \frac{\lambda \alpha R}{2\Delta} + \frac{1 - \Delta}{2\lambda \Delta} + \frac{\log \Delta}{2\beta} \\ &\quad - \frac{1}{\beta} \log \cosh \left[\left(\frac{\beta}{\Delta} \right) (a\xi + \sqrt{\alpha R} z) \right], \end{aligned} \quad (\text{H4})$$

where a, Q_0, Q, R, Δ are the various order parameters of the system and $\bar{A}$ denotes averaging of A with respect to z weighted by $\exp(-z^2/2)$. λ denotes the degree of unlearning and is the proxy for τ_2 in the case of spiking neurons. The order parameter of interest for us here is a , which denotes the retrieval of the various patterns stored in the system and is given as $a = \frac{1}{N} \sum_{v,i} (1 + \lambda C)_{1,v}^{-1} \xi_i^v \langle \sigma_i \rangle = (1 + \lambda)^{-1} m$ (when in retrieval state). We solve for the saddle-point equations of (H4) to find the saddle points. To study the curvature of the function at the various saddle points with respect to a , we compute the Hessian of $f(a, Q_0, Q, R, \Delta)$. $\partial^2 f / \partial a^2$ tells us about the dependence of f on a at the saddle points.

-
- [1] C. Marullo and E. Agliari, Boltzmann machines as generalized hopfield networks: A review of recent results and outlooks, *Entropy* **23**, 34 (2020).
 - [2] N. L. Roux and Y. Bengio, Representational power of restricted Boltzmann machines and deep belief networks, *Neural Comput.* **20**, 1631 (2008).
 - [3] K. Shimagaki and M. Weigt, Selection of sequence Motifs and generative Hopfield-Potts models for protein families, *Phys. Rev. E* **100**, 032128 (2019).
 - [4] O. H. Schnaack, L. Peliti, and A. Nourmohammad, Learning and organization of memory for evolving patterns, *Phys. Rev. X* **12**, 021063 (2022).
 - [5] P. Brennan, H. Kaba, and E. B. Keverne, Olfactory recognition: A simple memory system, *Science* **250**, 1223 (1990).
 - [6] L. B. Haberly, and J. M. Bower, Olfactory cortex: Model circuit for study of associative memory? *Trends Neurosci.* **12**, 258 (1989).
 - [7] D. A. Wilson, A. R. Best, and R. M. Sullivan, Plasticity in the olfactory system: Lessons for the neurobiology of memory, *Neuroscientist* **10**, 513 (2004).
 - [8] J. P. Barton, M. Kardar, and A. K. Chakraborty, Scaling laws describe memories of host–pathogen riposte in the HIV population, *Proc. Natl. Acad. Sci. USA* **112**, 1965 (2015).
 - [9] S. Cocco and R. Monasson, Adaptive cluster expansion for inferring Boltzmann machines with noisy data, *Phys. Rev. Lett.* **106**, 090601 (2011).
 - [10] S. E. Palmer, O. Marre, M. J. Berry, and W. Bialek, Predictive information in a sensory population, *Proc. Natl. Acad. Sci. USA* **112**, 6908 (2015).
 - [11] D. Krotov, A new frontier for Hopfield networks, *Nat. Rev. Phys.* **5**, 366 (2023).
 - [12] N. E. Boukacem, A. Leary, R. Thériault, F. Gottlieb, M. Mani, and P. François, Waddington landscape for prototype learning in generalized Hopfield networks, *Phys. Rev. Res.* **6**, 033098 (2024).
 - [13] J. J. Hopfield, D. I. Feinstein, and R. G. Palmer, Unlearning has a stabilizing effect in collective memories, *Nature (London)* **304**, 158 (1983).
 - [14] A. K. Behera, M. Rao, S. Sastry, and S. Vaikuntanathan, Enhanced associative memory, classification, and learning with active dynamics, *Phys. Rev. X* **13**, 041043 (2023).
 - [15] B. Derrida, E. Gardner, and A. Zippelius, An exactly solvable asymmetric neural network model, *Europhys. Lett.* **4**, 167 (1987).
 - [16] J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, *Proc. Natl. Acad. Sci. USA* **79**, 2554 (1982).
 - [17] D. J. Amit and D. J. Amit, *Modeling Brain Function: The World of Attractor Neural Networks* (Cambridge University Press, Cambridge, UK, 1989).
 - [18] D. J. Amit, H. Gutfreund, and H. Sompolinsky, Storing infinite numbers of patterns in a spin-glass model of neural networks, *Phys. Rev. Lett.* **55**, 1530 (1985).
 - [19] E. Agliari, F. Alemanno, A. Barra, and A. Fachechi, Dreaming neural networks: Rigorous results, *J. Stat. Mech.* (2019) 083503.
 - [20] G. E. Hinton, Training products of experts by minimizing contrastive divergence, *Neural Comput.* **14**, 1771 (2002).
 - [21] M. A. Carreira-Perpinan and G. Hinton, On contrastive divergence learning, in *International Workshop on Artificial Intelligence and Statistics* (PMLR, 2005), pp. 33–40.
 - [22] G. Hinton, The forward-forward algorithm: Some preliminary investigations, *arXiv:2212.13345*.
 - [23] V. S. Dotsenko, N. D. Yarunin, and E. A. Dorotheyev, Statistical mechanics of Hopfield-like neural networks with modified interactions, *J. Phys. A: Math. Gen.* **24**, 2419 (1991).
 - [24] É. Fodor, C. Nardini, M. E. Cates, J. Tailleur, P. Visco, and F. van Wijland, How far from equilibrium is active matter? *Phys. Rev. Lett.* **117**, 038103 (2016).
 - [25] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, A learning algorithm for Boltzmann machines, *Cognitive Sci.* **9**, 147 (1985).
 - [26] M. Mézard and J. Sakellariou, Exact mean-field inference in asymmetric kinetic Ising systems, *J. Stat. Mech.* (2011) L07001.
 - [27] M. Aguilera, S. A. Moosavi, and H. Shimazaki, A unifying framework for mean-field theories of asymmetric kinetic Ising systems, *Nat. Commun.* **12**, 1197 (2021).
 - [28] S. Jose, D. Mandal, M. Barma, and K. Ramola, Active random walks in one and two dimensions, *Phys. Rev. E* **105**, 064103 (2022).
 - [29] P. Smolensky *et al.*, *Information Processing in Dynamical Systems: Foundations of Harmony Theory* (Department of Computer Science, University of Colorado, Boulder, US, 1986).

- [30] M. Smart and A. Zilman, On the mapping between Hopfield networks and restricted Boltzmann machines, [arXiv:2101.11744](#).
- [31] A. Barra, A. Bernacchia, E. Santucci, and P. Contucci, On the equivalence of Hopfield networks and Boltzmann machines, [Neural Netw.](#) **34**, 1 (2012).
- [32] F. E. Leonelli, E. Agliari, L. Albanese, and A. Barra, On the effective for restricted Boltzmann machines via duality with Hopfield model, [Neural Netw.](#) **143**, 314 (2021).
- [33] M. Caira and W. Bialek, Non-Boltzmann dynamics in networks of spiking neurons, *Adv. Neural Inf. Proc. Syst.* **2**, 109 (1989).
- [34] W. Maass, Networks of spiking neurons: The third generation of neural network models, [Neural Netw.](#) **10**, 1659 (1997).
- [35] J. J. Hopfield, Neurons with graded response have collective computational properties like those of two-state neurons, [Proc. Natl. Acad. Sci. USA](#) **81**, 3088 (1984).
- [36] G. Stuart, N. Spruston, H. Michael *et al.*, *Dendrites* (Oxford University Press, Oxford, 2016).
- [37] W. Zheng, N. P. Schafer, A. Davtyan, G. A. Papoian, and P. G. Wolynes, Predictive energy landscapes for protein–protein association, [Proc. Natl. Acad. Sci. USA](#) **109**, 19244 (2012).
- [38] M. Beissinger and J. Buchner, How chaperones fold proteins, [Biol. Chem.](#) **379**, 245 (1998).
- [39] P. Goloubinoff, A. S. Sassi, B. Fauvet, A. Barducci, and P. De L. Rios, Chaperones convert the energy from ATP into the nonequilibrium stabilization of native proteins, [Nat. Chem. Biol.](#) **14**, 388 (2018).
- [40] H. Saibil, Chaperone machines for protein folding, unfolding and disaggregation, [Nat. Rev. Mol. Cell Biol.](#) **14**, 630 (2013).
- [41] P. Dayan and L. F. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems* (MIT Press, Cambridge, MA, 2005).
- [42] A. Y. Plakhov and S. A. Semenov, The modified unlearning procedure for enhancing storage capacity in Hopfield network, in *Proceedings 1992 RNNS/IEEE Symposium on Neuroinformatics and Neurocomputers* (IEEE, 1992), pp. 242–251.