

Maximizing Memory Capacity in Heterogeneous Networks

Kaining Zhang¹ and Gaia Tavoni^{1,2,3,*}

¹*Department of Neuroscience, Washington University in St. Louis, St. Louis, Missouri 63130, USA*

²*Department of Physics, Washington University in St. Louis, St. Louis, Missouri 63130, USA*

³*Department of Electrical and Systems Engineering, Washington University in St. Louis, St. Louis, Missouri 63130, USA*



(Received 25 September 2024; accepted 22 May 2025; published 20 June 2025)

A central question in neuroscience is which neuronal and connectivity properties determine a network's ability to store information. The theory of Hopfield models has provided powerful tools to investigate this question. Here, we generalize classical results on the memory capacity of homogeneous Hopfield models to a broader class of heterogeneous networks. We derive an analytical formula for the maximal capacity of networks with arbitrary, and generally heterogeneous, activation rates (coding levels) of neurons in memory patterns, as well as arbitrary connectivity architectures. We use this result to make normative predictions about the properties that maximize capacity in brain-inspired networks. We show that, although heterogeneity in neuron coding levels and inward connection counts (in-degrees) generally reduces capacity, maximal capacity is retained when these two parameters are correlated. This prediction holds across various biologically relevant scenarios, including the storage of independent patterns in both classical and dendritic networks, as well as the storage of example patterns clustered around prototypes representing concepts. In the latter case, heterogeneity similarly affects the capacity for both examples and concepts. Finally, we analyze bipartite models of the CA3-DG circuit in the hippocampus, known to be crucial for memory encoding. In these networks, capacity is maximized by a quasi-indexing encoding scheme, where each neuron in the DG binds subsets of features from a few memory patterns stored in CA3. Compared to a complete-indexing scheme, where each DG unit binds all the features of a single pattern, quasi-indexing significantly improves both network capacity and memory robustness to neuron ablation. These findings introduce new dimensions to the hippocampal index theory of memory encoding and retrieval, while suggesting specific neural substrates for these functions.

DOI: [10.1103/qb9h-qxkx](https://doi.org/10.1103/qb9h-qxkx)

I. INTRODUCTION

In a seminal paper [1], Hopfield introduced a class of artificial neural networks—drawing inspiration from models in statistical physics [2–4]—that can store memories and retrieve them from partial cues, providing a rigorous theoretical framework for studying memory function in the brain [5–14]. The classic Hopfield model is a network of binary (active or silent) neurons that are potentially fully connected. Memories are stored by adapting the synaptic weights according to biologically inspired Hebbian plasticity rules [1,15]. After this learning process, the memories are encoded in the matrix of synaptic connections and function as attractors or fixed points of the network dynamics: starting from any given activity configuration, neurons receiving strong positive inputs become active, while the others become silent, until the network converges to one of the memory patterns [5].

Memories can only be stored up to some maximum capacity. The quantification of the capacity of Hopfield networks

has been the subject of an extensive literature. Early theoretical research demonstrated that the maximum number of random patterns that can be stored by a simple Hebbian learning rule is $C \sim 0.14N$ [1,16,17]: the capacity increases linearly with the number of neurons. Gardner later introduced a method to compute the maximal memory capacity, independent of the learning rule used [18–20]. This upper bound is $C = 2N$ for uncorrelated random patterns and is attainable with a local learning algorithm, also introduced in [18]. This research focused on abstract networks of fully connected binary neurons with uniform coding levels, i.e., identical activation probabilities for all cells in the memory patterns. Other studies have considered diluted Hopfield networks, in which connections are randomly dropped with a fixed probability [6,21–23], and fully or sparsely connected homogeneous Hopfield networks with discrete synapses [24]. A few studies have addressed the conditions for memory retrieval in heterogeneous networks [6,25,26], but these extensions are limited to symmetric graphs (where each connection is identical in both directions) and symmetric Hebbian learning. All these analyses also maintained homogeneous coding levels across all neurons. Through this fundamental theoretical research, powerful tools have been developed to study memory in neural networks. However, the application of these tools to biologically realistic cases has remained a relatively underexplored field. Notably, biological networks are neither fully nor homogeneously or symmetrically connected, and

*Contact author: gaia.tavoni@wustl.edu

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.

neurons may have different coding levels. In the spirit of gaining insights into memory function in real brains, we generalize previous approaches to derive an analytical solution for the maximal memory capacity of networks with arbitrary architectures—that is, any possible arrangement of connections between cells—and arbitrary distributions of coding levels across neurons. Our analysis is relevant for understanding how various structural and neurophysiological constraints influence a network’s ability to store information.

The problem of estimating the memory capacity of Hopfield networks is tightly linked to the perceptron capacity problem. A perceptron is a simple feedforward network with a binary output unit that linearly separates N -dimensional input patterns into two alternative classes. The perceptron capacity is the maximum number of input patterns that can be linearly assigned to their classes or labels. The methods developed to analyze perceptrons and their capacity can often be employed to study Hopfield networks, and vice versa [18,27–33]. Previous research has analyzed perceptrons and Hopfield networks with uncorrelated and correlated units or patterns [18,29,34–38], correlated input-label pairs [39,40], and other relevant classification and learning scenarios beyond storage capacity [28,35–37]. In particular, Refs. [29,34,35] demonstrated that perceptron capacity is independent of the input distributions as long as the inputs share the same covariance. However, these studies assumed a symmetric activation probability for the output unit, i.e., a 50% split of input patterns into the two possible classes. This limitation prevents a direct application of their results to Hopfield networks with arbitrary coding levels. On the other hand, Gardner’s original result [18] provided a formula for perceptron capacity in the general case of an arbitrary activation probability for the output unit. However, that result was derived under the assumption of independently and identically distributed (i.i.d.) inputs. Here, we bridge the gap between these two sets of results. Using a simple extension of Gardner’s method, we derive a formula for perceptron capacity that simultaneously accounts for both heterogeneous input distributions and arbitrary activation probability of the output unit. This formula is instrumental in determining the memory capacity of Hopfield networks with arbitrary—and generally heterogeneous—coding levels, as well as heterogeneous architectures. We find that capacity depends on both the coding level and the number of inward connections of each cell.

We then use our analytical formula to study the memory capacity of biologically inspired networks. We first consider a general class of heterogeneous networks in which the number of inward connections (in-degree) and the coding levels of neurons are randomly sampled from distributions with a given variance. This variance determines the degree of network heterogeneity. We find that increasing heterogeneity generally reduces network memory capacity, except when coding levels and in-degrees are correlated: if neurons with more inward connections have proportionally higher coding levels, memory capacity is maximized. We also show that this result generalizes to other biologically relevant memory capacity problems: the storage of “examples”—i.e., patterns clustered around prototypes or “concepts” [41,42]—and the storage of memories in networks with dendritic structures, where neu-

rons are modeled as treelike committee machines that emulate the properties of neuronal dendritic arbors [43–49].

Next, we analyze a class of bipartite networks composed of two reciprocally connected but internally disconnected layers. We interpret one layer as storing the features or informational content of memories and the other layer as storing an internal representation of memory patterns [6,8,50–52]. We use this network architecture to represent, at a high level, the CA3-DG (Cornu Ammonis 3–Dentate Gyrus) circuit in the hippocampus, during the initial memory encoding stage. In this model, we show that the capacity for storing information is maximized when DG units have sparse coding levels. The interlayer connectivity that maximizes capacity supports a “quasi-indexing” function for the DG, where each unit tends to index—i.e., bind—subsets of features (in CA3) from a small number of patterns. This quasi-indexing scheme contrasts with a complete-indexing scheme, in which each DG unit binds all and only the features of a distinct pattern, as well as with a nonindexing scheme, in which each DG unit binds random features across patterns. While indexing supports the pattern separation and pattern completion functions of the hippocampus, as suggested by the Hippocampal Index Theory [53–57], it also limits capacity and minimizes memory robustness to unit ablation. We find that the quasi-indexing scheme achieves maximum capacity while preserving robustness, especially in large networks.

II. RESULTS

A. Capacity of perceptrons with arbitrary input and output states

1. Perceptrons with independent input units

A perceptron [Fig. 1(a)] is a simple feedforward network with one output unit that responds to N input units according to the rule

$$g^{\text{out}} = \text{sgn} \left(\sum_{j=1}^N J_j g_j^{\text{in}} \right), \quad (1)$$

where g^{out} and g_j^{in} represent the output and the j th input state, respectively, and $\{J_j\}$ are the synaptic weights: the output state is the sign of the total input to the output unit.

A perceptron can learn to associate N -dimensional input patterns $\{\xi^\mu = (\xi_1^\mu, \dots, \xi_N^\mu)\}$ with specified binary classes or labels $\{\chi^\mu = \pm 1\}$. The input-label pairs are learned by adjusting the synaptic weights $\{J_j\}$ so that they satisfy the classification constraint

$$\chi^\mu \left(\sum_j J_j \xi_j^\mu \right) > 0 \quad (2)$$

for every input pattern indexed by μ . Constraints (2) define a hyperplane in the input space $\{\xi_j\}$ ($j = 1, \dots, N$) that separates the patterns labeled as $+1$ from those labeled as -1 [Fig. 1(b), left, solid line]. A more restrictive condition can be enforced, requiring that a minimum distance exists between the patterns and the hyperplane separating the classes. This condition ensures that the classification is stable against a proportional amount of noise. The minimum distance between the patterns and the hyperplane is

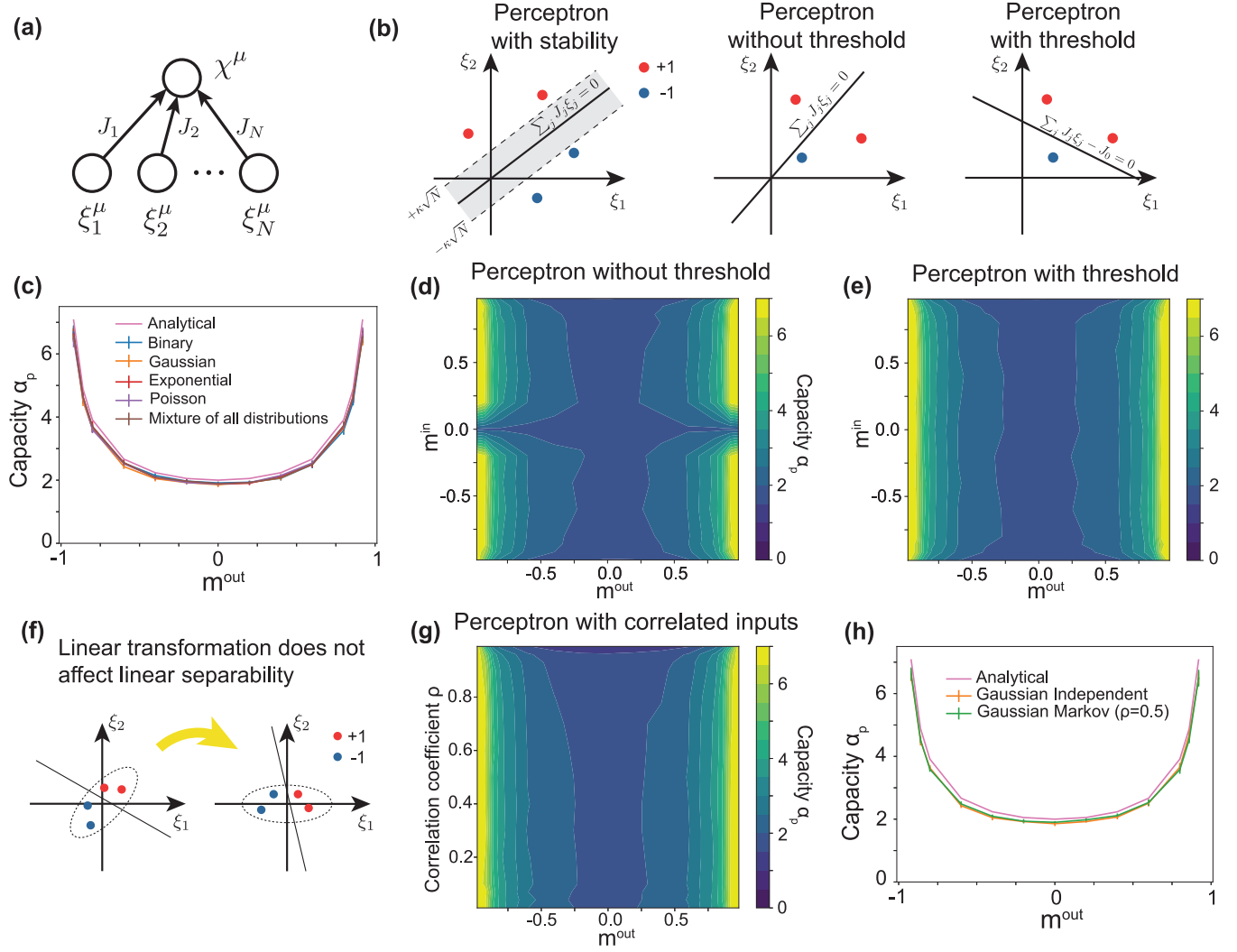


FIG. 1. Capacity of a perceptron with arbitrary input and output states. (a) A perceptron is a feedforward network that stores pattern-label pairs in the weights $\{J_j\}$; inputs ξ_j^μ can have arbitrary (independent) distributions with distinct finite means and variances; labels χ^μ are binary. (b) A perceptron is a linear classifier. Axes: input states; dots: input patterns (here $N = 2$ for simplicity). Within capacity, the perceptron can find a hyperplane (solid line) that separates the patterns according to their labels (red and blue). Left: to enforce classification stability against noise, the patterns must maintain a minimum margin (κ) from the hyperplane (dashed lines). Middle: for perceptrons with a zero activation threshold, the decision hyperplane must pass through the origin. Right: for perceptrons with a nonzero activation threshold, the decision hyperplane is not constrained to pass through the origin, improving linear separability in low dimension. (c) Maximal capacity of a perceptron as a function of the output mean m^{out} ; analytical solution, Eqs. (3) (magenta), and mean $\pm$ std over 10 numerical estimates ($N = 1000$) for perceptrons with a nonzero threshold and different input distributions (different colors). (d) Numerical estimates of the maximal capacity of a perceptron with a zero threshold. Input (output) states are independently sampled from a binary distribution with mean m^{in} (m^{out}). (e) Same as (d) for a perceptron with a nonzero threshold. In the large- N limit, the capacity depends only on m^{out} (not on m^{in}). Slight variations in capacity with m^{in} are due to finite-size effects. (f) Linear separability of the input patterns is not affected by linear transformations. (g),(h) The maximal capacity is unaffected by input correlations, provided these can be removed through a linear transformation of the patterns. (g) Input states are sampled from a Gaussian Markov chain, and numerical estimates of the maximal capacity are shown as a function of m^{out} and the correlation coefficient ρ between adjacent input states ξ_j^μ and ξ_{j+1}^μ . (h) Analytical solution as a function of m^{out} , Eqs. (3) (magenta), coincides with numerical estimates for uncorrelated ($\rho = 0$, orange) and correlated ($\rho = 0.5$, green) input states.

determined by a stability parameter $\kappa > 0$ [Fig. 1(b), left, dashed lines].

The capacity C_p is the maximum number of input patterns that the perceptron can learn to classify with probability 1. We compute the normalized capacity, $\alpha_p = C_p/N$, in the general case of input-pattern states ξ_j^μ that are *independent* but otherwise *arbitrarily distributed*, each with a distinct and

finite mean m_j and *finite variance* σ_j^2 . Different means correspond to different activation probabilities of the units in the input patterns, i.e., different neuron coding levels. We assume that the labels follow a Bernoulli distribution with $p(\chi^\mu = 1) = (1 + m^{\text{out}})/2$ and $p(\chi^\mu = -1) = (1 - m^{\text{out}})/2$; m^{out} is the output (or label) mean, which can be different from the input means $\{m_j\}$.

To calculate the capacity of this general class of perceptrons, we use a method pioneered by Gardner, based on estimating (for large N) the typical volume of the synaptic weights that satisfy the classification constraints for all the input patterns [18,19,58] (details in Appendix A). The capacity is maximized when this volume shrinks to zero, i.e., no input pattern can be added without breaking the constraints. To generalize the method to inputs with arbitrary variances, we use an elliptical regularization on the synaptic weights, which effectively normalizes the postsynaptic input variance: $\sum_{j=1}^N \sigma_j^2 J_j^2 = N$. This regularization reduces to the standard spherical regularization used by Gardner when the input variances are identical.

In the $\kappa = 0$ case, the solution to the classification constraints (2) is completely determined by the following joint equations, which can be evaluated numerically (see Appendix A and the Methods section):

$$\alpha_p = 2 \left\{ 1 - m^{\text{out}} \int_{-M}^M Dy \right\}^{-1}, \quad (3a)$$

$$\frac{1 + m^{\text{out}}}{2} \int_M^\infty Dy (y - M) = \frac{1 - m^{\text{out}}}{2} \int_{-M}^\infty Dy (y + M). \quad (3b)$$

The integration limits ($\pm M$) in Eq. (3a) are given by Eq. (3b) when $m_j \neq 0$ for at least one input unit. When $m_j = 0$ for all input units, $M = 0$ and $\alpha_p = 2$.

The parameters $\{m_j, \sigma_j^2\}$, or any other details of the input-state distributions, do not factor into the solution, Eqs. (3). Thus, the perceptron capacity is independent of the statistics of input states and depends only on the output mean activation, m^{out} . The capacity is maximized at extreme (low or high) values of m^{out} [Fig. 1(c), magenta curve]. The only exception is when the distribution of input states is centered at the origin ($m_j = 0$ for all j): as mentioned above, at this singular point, the maximal capacity is $\alpha_p = 2$, regardless of m^{out} [Fig. 1(d)].

Enforcing classification stability against noise ($\kappa > 0$) does not change these results, except for an overall reduction in capacity as κ increases (see Fig. 1 of the Supplemental Material (SM) [59]).

2. Perceptrons with nonzero activation thresholds and correlated input units

A more general definition of the perceptron includes a bias term corresponding to an activation threshold of the output unit:

$$g^{\text{out}} = \text{sgn} \left(\sum_{j=1}^N J_j g_j^{\text{in}} - J_0 \right). \quad (4)$$

Introducing the activation threshold J_0 is equivalent to adding a formal input unit with weight J_0 and constant state $\xi_0^\mu = -1$ for every pattern μ [18]. This is equivalent to sampling an additional state from a binary distribution with $p(\xi_0^\mu = 1) = (1 + m_0)/2$, $p(\xi_0^\mu = -1) = (1 - m_0)/2$, where $m_0 \rightarrow -1$.

Since the perceptron capacity does not depend on the distributions of input states, and the increase in dimension from N to $N + 1$ is irrelevant in the large- N limit, the capacity

formula (3) is still valid in this case. Due to the additional formal unit with $m_0 = -1$, the same formula also applies when $m_j = 0$ for $j = 1, \dots, N$ in perceptrons with nonzero activation thresholds, removing the singularity present in the case with zero threshold. In summary, in the large- N limit, the capacity of the perceptron with nonzero threshold is determined solely by the mean value of the labels m^{out} and is independent of the input statistics. We confirm this result in numerical estimates of the solution [Figs. 1(c) and 1(e)].

The general equivalence between the capacities of perceptrons with and without an activation threshold breaks down in low dimensions (small N). This is illustrated geometrically in Fig. 1(b) for $N = 2$. For perceptrons without a threshold, the hyperplane ($\sum_j J_j \xi_j = 0$) separating the input patterns into binary classes must intersect the origin of the input space [Fig. 1(b), left and middle]; in contrast, the threshold allows the hyperplane ($\sum_j J_j \xi_j = J_0$) to be shifted and tilted in such a way that it can better separate the input patterns without the restriction of passing through the origin [Fig. 1(b), right]. While in high dimension adding a single constraint—the intersection of the hyperplane with the origin—has a vanishing impact on classification performance (except when all $m_j = 0$), in low dimension it can substantially decrease the perceptron capacity.

Our solution for the perceptron capacity further generalizes to the case of *correlated input states*, even if Eqs. (3) are derived under the assumption that the input states ξ_j^μ are independent. This invariance property follows from the fact that linear separability is not affected by linear transformations of the input patterns. Thus, the capacity of the perceptron with threshold remains unchanged when the input units are correlated, as long as they can be decorrelated through a linear transformation of the input patterns [Fig. 1(f)]. Such a transformation can be obtained by applying singular value decomposition (SVD) either to the correlation matrix or directly to the pattern matrix, as described in detail in [34]. This invariance was already observed in the case of a symmetric output ($m^{\text{out}} = 0$) in [34], where the capacity was shown to be proportional to the rank of the pattern matrix and independent of the distribution of its nonzero eigenvalues. We validate our result for general m^{out} by numerically estimating the capacity of a perceptron with input states sampled from a Gaussian Markov chain, so that each unit is correlated with its neighbors (similar to the case considered in [29]; details are in Methods). The estimated capacity aligns with our analytical and numerical solutions for uncorrelated inputs [Figs. 1(g) and 1(h)].

B. Capacity of Hopfield networks with arbitrary architectures and coding levels

We can now use the perceptron results to determine the capacity of Hopfield networks with heterogeneous architectures (arbitrary arrangements of connections between cells) and heterogeneous coding levels (different activation probabilities across neurons in the memory patterns), Fig. 2(a). We define a network architecture—or structure—in terms of a binary matrix J^S with $J_{ij}^S = 1$ if neuron j is connected to neuron i (in the $j \rightarrow i$ direction) and $J_{ij}^S = 0$ if the neurons are not

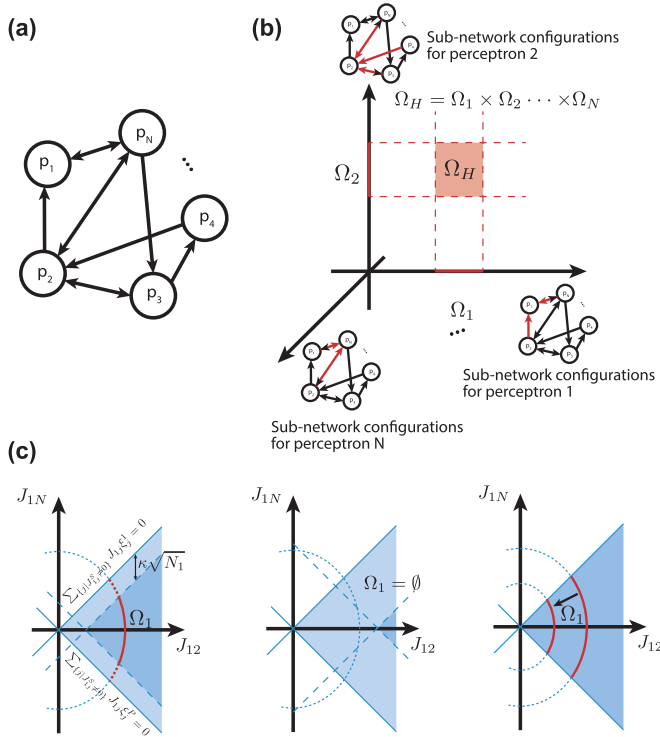


FIG. 2. Calculating the capacity of heterogeneous Hopfield networks. (a) A heterogeneous Hopfield model is defined by an arbitrary architecture, or an arrangement of directed connections $J_{ij}^S = 1$ (arrows), and arbitrary neuron coding levels $\{p_i\}$. (b) Sketch of the solution set Ω_H for the Hopfield model, i.e., the ensemble of network configurations $\{J_{ij}\}$ that satisfy the fixed-point conditions (5) for the P memory patterns. Ω_H is the product of the solution sets Ω_i for the N independent perceptrons defined on the network. Ω_i is the ensemble of subnetwork configurations for perceptron i (red arrows) that satisfy the subset of conditions (5) for that perceptron. The product $\Omega_1 \times \dots \times \Omega_N$ is defined by all possible combinations of solutions (subnetwork configurations) in the different sets. Red square: projection of Ω_H onto the two-dimensional plane (Ω_1, Ω_2). (c) A close examination of the solution set Ω_1 for the subnetwork corresponding to the first perceptron; the sketch is drawn for $N = 3$. Left: Ω_1 consists of subnetwork configurations $\{J_{1j}\}$ (with $J_{1j}^S \neq 0$) that satisfy the fixed-point conditions (5) for $i = 1$; $\mu = 1, \dots, P$ and the elliptical weight regularization $\sum_{j|J_{1j}^S \neq 0} \sigma_j^2 J_{1j}^2 = N_1$. Each memory pattern constrains the solution space to be on one side of a hyperplane (a half-space); P memories define P hyperplanes (solid blue lines) and corresponding intersection of half-spaces (light blue region). The weight regularization constrains the solution space to lie on the ellipse (dotted line). Altogether, these conditions constrain the solution space (Ω_1) to the red arc. Enforcing stability of the memories against noise corresponds to shifting each hyperplane by $\kappa\sqrt{N_1}$, thus shrinking the intersection of half-spaces (dark blue region) and the solution set (solid red line). Middle: Increasing κ can make the intersection of half-spaces (dark blue area) fall outside the elliptical region: the solution set Ω_1 vanishes and the memory capacity decreases. Right: When $\kappa = 0$, the elliptical constraint contributes to defining Ω_H (different elliptical regularizations yield different solution sets, red arcs) but cannot make the solution set vanish. Thus, the maximal capacity is unaltered by the weight regularization when $\kappa = 0$.

connected in that direction. The matrix can be asymmetric and we assume that neurons do not connect to themselves ($J_{ii}^S = 0$). We further define P patterns $\xi^\mu = (\xi_1^\mu, \dots, \xi_N^\mu)$ to memorize, such that $\xi_j^\mu = 1$ (active) with probability p_j across the patterns and $\xi_j^\mu = -1$ (inactive) with probability $1 - p_j$. Thus, p_j represents the coding level of neuron j or the expected fraction of memory patterns in which that neuron is active; unlike previous studies [7,24,25,29,60], our definition allows for different coding levels in different neurons [Fig. 2(a)].

The same activation rule used for the perceptron output, Eq. (1), defines the dynamics of Hopfield networks: a neuron becomes active if it receives a positive total input from the other cells and inactive if it receives a negative total input. There are fixed points in this dynamics: at these points, the state of every neuron has the same sign as its total input. The fixed points are memories stored in the network, which can be retrieved from partial cues—representing corrupted versions of the patterns—through the same dynamics.

Thus, to store P memory patterns $\xi^\mu = (\xi_1^\mu, \dots, \xi_N^\mu)$, the network weights $\{J_{ij}\}$ need to satisfy the following fixed-point conditions:

$$\xi_i^\mu \left(\sum_{\{j|J_{ij}^S \neq 0\}} J_{ij} \xi_j^\mu \right) > 0 \quad \text{for } i = 1, \dots, N; \mu = 1, \dots, P. \quad (5)$$

It is straightforward to see that the problem of storing memories in a Hopfield network with N neurons factorizes into N independent perceptron classification problems [Fig. 2(b)]: each neuron i in the Hopfield network can be thought of as the output unit of one perceptron, with label $\chi^\mu = \xi_i^\mu$, and all the neurons that have connections to it can be thought of as the input units for that perceptron, with pattern states $\{\xi_j^\mu | J_{ij}^S \neq 0\}$; repeating this mapping for every neuron defines N perceptrons. The inequalities (5) are precisely the classification constraints (2) applied to all N perceptrons. Thus, the solution set for the Hopfield model, that is, the set of network configurations $\{J_{ij}\}$ with structure J^S that meet all the inequalities (5), is given by all possible combinations of N independent sub-network configurations that meet the classification constraints for the different perceptrons. In mathematical language, the solution set Ω_H for the Hopfield model is the Cartesian product of the solution sets Ω_i for the N perceptrons [Fig. 2(b)]:

$$\begin{aligned} \Omega_H &= \Omega_1 \times \Omega_2 \times \dots \times \Omega_N \quad \text{with} \\ \Omega_i &\stackrel{\text{def}}{=} \left\{ (J_{ij}) \left| J_{ij}^S \neq 0, \xi_i^\mu \sum_{\{j|J_{ij}^S \neq 0\}} J_{ij} \xi_j^\mu \right. \right. \\ &\quad \left. \left. > 0 \text{ for all } \mu, i \text{ fixed} \right\}. \end{aligned} \quad (6)$$

We define the memory capacity of the Hopfield network, $C_H = \alpha_H N$, as the maximum number of patterns for which all the N perceptrons satisfy the fixed-point conditions with probability 1. Above this extreme number, there exists a set

of realizations of the patterns with nonzero measure that violate the constraints in at least one perceptron (denoted by i). For those realizations of the patterns, the volume of the solution set Ω_i , and thus the volume of Ω_H , shrinks to zero (Appendix A). It follows that the memory capacity of the network, as we defined it, coincides with that of the perceptron with the smallest maximal capacity. Thus, for a heterogeneous Hopfield model, we obtain

$$\alpha_H = \min_i \left(\frac{N_i}{N} \alpha_p(2p_i - 1) \right), \quad (7)$$

where $N_i = \sum_j J_{ij}^S$ is the number of inward connections (the in-degree) of neuron i , and $\alpha_p(2p_i - 1) = C_p/N_i$ is the normalized capacity of the perceptron with output neuron i . The latter is a function of $m_i = 2p_i - 1$, the expected state of neuron i in the memory patterns [replacing m^{out} in Eqs. (3)].

This formula generalizes classical results on the capacity of Hopfield models to heterogeneous networks, accommodating arbitrary constraints on the architecture as defined by J^S , and varying neuron coding levels as defined by the different p_i .

Similarly to the perceptron, a more restrictive condition can be enforced, requiring the memory patterns to have sufficiently wide basins of attraction. This condition ensures that the memories can be retrieved starting from a relatively wide neighborhood of initial configurations, making them stable against a proportional amount of noise. The width of the basins of attraction of the memories is controlled by a stability parameter κ [7,18]. Setting $\kappa > 0$ is equivalent to enforcing classification stability in each of the N perceptrons defined on the Hopfield network [Fig. 2(c), left]. Thus, the network capacity for $\kappa \geq 0$ is given by Eq. (7) with α_p replaced by the capacity of the perceptrons with stability κ (derived in Appendix A). Increasing κ improves memory retrievability while reducing the memory capacity of the network [Fig. 2(c), middle].

When $\kappa = 0$, the capacity result applies to any Hopfield network as defined above. In contrast, under the stronger condition $\kappa > 0$, the result holds only for a subclass of Hopfield networks that satisfy the weight regularization condition $\sum_{j|J_{ij}^S \neq 0} \sigma_j^2 J_{ij}^2 = N_i$ for every neuron i , where σ_j^2 denotes the variance of the states of unit j . This regularization—which can be interpreted as a normalization of the postsynaptic input variance, similar to the case of perceptrons—is employed in the capacity calculation for any $\kappa \geq 0$, but it does not affect the result when $\kappa = 0$ [Fig. 2(c), right, and Appendixes A and B].

Lastly, since each perceptron's solution set Ω_i is convex in the weight space $\{J_{ij}\}$ [43,61], and a convex loss function exists for this problem, such as the hinge loss function used in support vector machines (SVMs) [62], any reasonable training algorithm can find a valid solution whenever one exists. Extending from individual perceptrons to the Hopfield network does not alter the convex nature of the problem. Consequently, the memory capacity achieved by numerical algorithms will asymptotically converge to the analytical capacity as $N \rightarrow \infty$.

C. Biologically relevant case studies

In the spirit of gaining insights into the memory capacity of biological neural networks, we apply our analytical results to two biologically relevant case studies: a general class of heterogeneous networks and a class of two-layer memory networks.

1. Correlated in-degrees and coding levels maximize memory capacity in heterogeneous networks

Neurons in the brain can have different activation probabilities within memory patterns (coding levels), different numbers of inward connections (in-degrees), and different numbers of outward connections (out-degrees). Thus, memory networks in the brain are largely heterogeneous. Below, we illustrate the effects of heterogeneity on memory capacity in the cases of independent patterns, example patterns organized around prototypes or concepts, and dendritic networks.

a. Memory capacity for independent patterns in heterogeneous networks. Our analytical formula (7) demonstrates that the maximum number of patterns that can be stored as fixed points in heterogeneous networks depends only on the neuron coding levels p_i and in-degrees N_i , while out-degrees do not influence memory capacity. Here, we examine how the statistics of the two relevant parameters affect capacity for independent patterns and identify the conditions under which capacity is maximized.

We generate networks with different distributions of coding levels and in-degrees: specifically, we vary the standard deviations of coding levels and inward-connection probabilities across neurons while keeping their means constant (see Methods). Increasing the standard deviation of either parameter increases heterogeneity [Fig. 3(a)]. In heterogeneous networks, the in-degrees and coding levels of the neurons can be independent, positively correlated, or negatively correlated [Figs. 3(b)–3(d)]. We analyze below the effects of heterogeneity in these three different cases.

When in-degrees and coding levels are independent, increasing the standard deviation of either parameter decreases the memory capacity of the network [Fig. 3(e)]. This result can be understood as follows: the network capacity [normalized by the number of neurons, Eq. (7)] is the minimum of the weighted perceptron capacities $\frac{N_i}{N} \alpha_p(2p_i - 1)$; when increasing the variability of N_i or p_i across neurons, the *variance* of the weighted perceptron capacities increases, while the *mean* remains more stable. Consequently, the *minimum* of the weighted perceptron capacities becomes lower, which makes the network capacity decrease.

This capacity reduction in heterogeneous networks is mitigated when the distributions of in-degrees N_i and coding levels p_i are such that the weighted perceptron capacities are roughly constant across neurons. In the regime of low coding levels, $p_i < 0.5$ (corresponding to $m_i < 0$), the perceptron capacity decreases with increasing p_i [Fig. 1(c)]. Thus, for the weighted perceptron capacity to be roughly constant, neurons with higher p_i should have higher N_i . The effects of a positive correlation between p_i and N_i on the network memory capacity are shown in Fig. 3(f). Compared to Fig. 3(e), the capacity remains high in the heterogeneous region, when

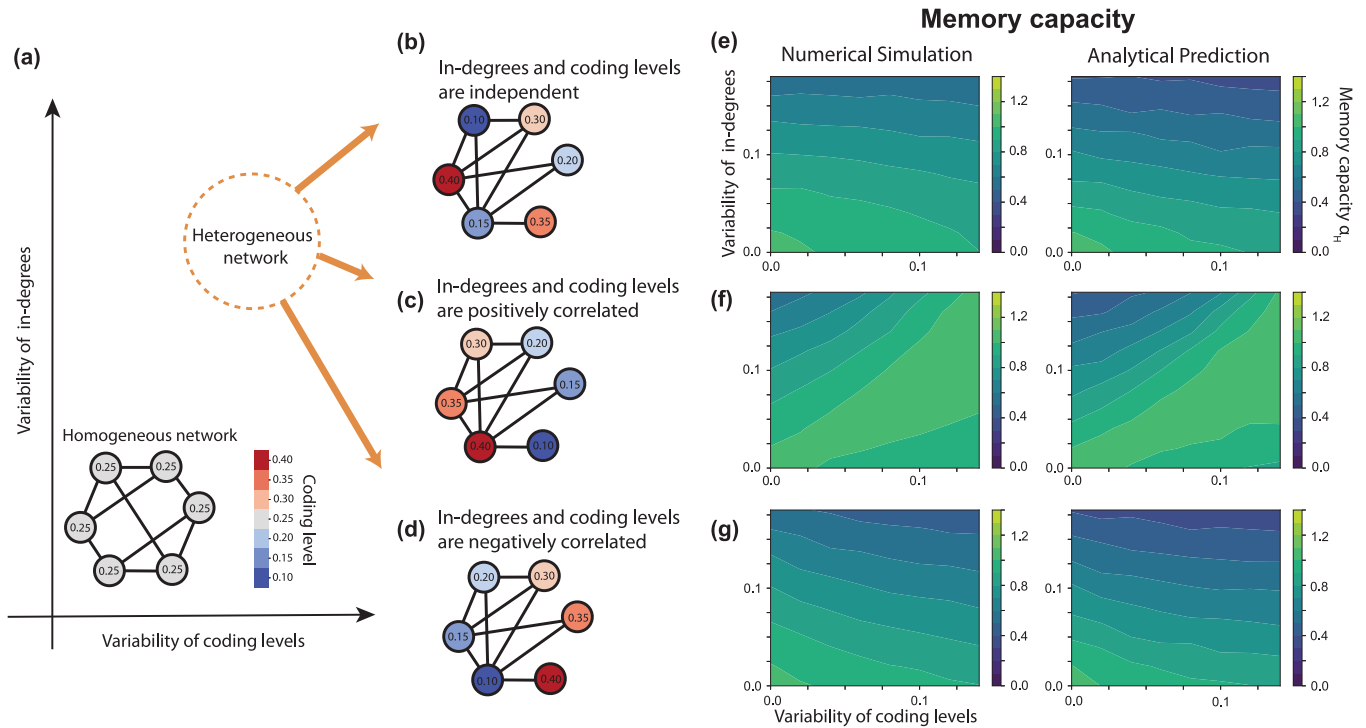


FIG. 3. Impact of heterogeneity on memory capacity for independent patterns. (a) We generate networks with different levels of heterogeneity by changing the variability of coding levels and in-degrees across neurons (specifically, the standard deviations of coding levels and inward-connection probabilities, respectively); the means over neurons of both parameters are held constant across networks (see Methods). Networks with low variability of both coding levels and in-degrees are homogeneous, whereas networks with high variability in these parameters are heterogeneous. (b)–(d) In heterogeneous networks, neuron in-degrees and coding levels can be independent (b), positively correlated (c), or negatively correlated (d). (e)–(g) Memory capacity for independent patterns as a function of the variability of in-degrees and coding levels, in the biologically plausible ($p_i < 0.5$) regime: numerical (left) and analytical (right) estimates. (e) When in-degrees and coding levels are independent, memory capacity decreases as the variability of either parameter increases. (f) When in-degrees and coding levels are positively correlated, the capacity of heterogeneous networks (with high variability of both parameters) is similar to the capacity of homogeneous networks (with low variability). (g) When in-degrees and coding levels are negatively correlated, memory capacity undergoes the steepest decrease with increasing network heterogeneity.

both in-degrees and coding levels are highly variable in the network.

This result is reversed when neurons with higher coding levels have sparser inward connections (negative-correlation case): as expected from the above argument, network capacity decreases even faster in this condition than in the independent case as heterogeneity increases [Fig. 3(g)].

As a control, we confirmed numerically that memory capacity is insensitive to neuron out-degrees, regardless of the correlation pattern between out-degrees and coding levels (Fig. 2 of the SM [59]). Furthermore, enforcing memory robustness ($\kappa > 0$) does not qualitatively change our conclusions (Fig. 3 of the SM [59]).

These results lead to the following normative prediction: to maximize memory capacity, neurons with more inward connections should have a higher probability of activation when encoding memories. How specific is this prediction? First, we note that this prediction is not equivalent to the intuitive idea that neurons receiving many excitatory inputs are expected to have high activation probabilities (i.e., coding levels). Indeed, inward connections are not exclusively excitatory; they can be excitatory or inhibitory, and our prediction applies to the total number of inward connections (in-degree).

Furthermore, in networks trained to store memories at capacity, high coding levels do not necessarily reflect a high proportion of excitatory inputs relative to the total number of inputs: the two variables are not significantly correlated, regardless of whether coding levels and in-degrees are positively correlated, independent, or negatively correlated [Fig. 4(a)]. Second, we note that in trained networks, a neuron coding level can be correlated with the total input it receives from other neurons, $\sum_{j \in \mathcal{I}_i^s, j \neq i} J_{ij} \xi_j^\mu$. However, this correlation does not imply a correlation between coding levels and in-degrees. In fact, in networks trained with the SVM algorithm, coding levels and total inputs tend to be (weakly) correlated when coding levels and in-degrees are independent or negatively correlated [Fig. 4(b)]. Thus, the requirement of a positive correlation between coding levels and in-degrees for capacity maximization differs from the trivial condition in which larger total inputs drive higher coding levels. Finally, we observe that in trained networks, coding levels always tend to increase as neuron activation thresholds decrease [Fig. 4(c)] to satisfy the fixed-point conditions [Eq. (4), for all memory patterns]. Since activation thresholds are generally independent of in-degrees, high coding levels can arise despite low in-degrees, indicating that our prediction is specific and nontrivial.

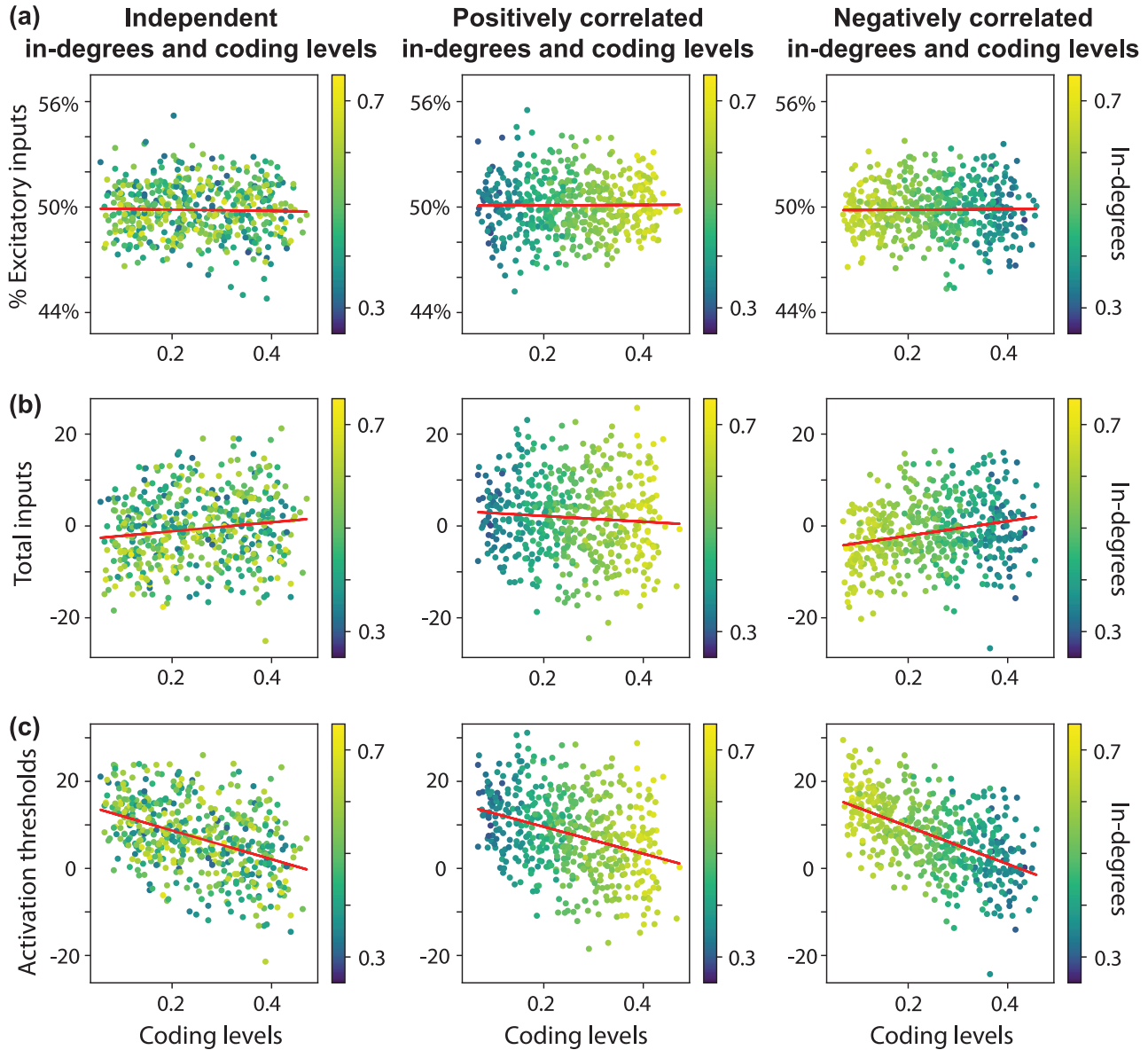


FIG. 4. Coding levels in heterogeneous networks at capacity are related to activation thresholds rather than neuron inputs. Three classes of heterogeneous networks (variability of in-degrees = variability of coding levels = 0.1), with independent, positively correlated, or negatively correlated in-degrees and coding levels, are generated as in Fig. 3 and trained to store memory patterns. We analyze the weights and neuron activation thresholds that maximize capacity in each network. Across networks, memory capacity is maximized when in-degrees and coding levels are positively correlated (middle panels). This normative condition is not trivially satisfied by any memory-storing network and does not imply trivial relationships between neuron coding levels and the sign (a) or total magnitude (b) of neuron inputs. (a)–(c) Each dot represents a neuron in a trained network, red lines represent linear fits, and r and p indicate Pearson correlation coefficients and p -values of the linear fits, respectively. (a) The percentage of excitatory inputs into each neuron i , defined as the fraction of input projections from other neurons j for which $J_{ij}\xi_j^\mu > 0$, is computed from the trained weights and neuronal states in each memory pattern and then averaged over patterns. Coding levels are not significantly correlated with the percentage of excitatory inputs, regardless of whether in-degrees and coding levels are independent, positively correlated, or negatively correlated (from left to right: $r = -0.026$, $p = 0.569$; $r = 0.007$, $p = 0.869$; $r = 0.011$, $p = 0.808$). (b) Each neuron total input, $\sum_{j|J_{ij}^\mu \neq 0} J_{ij}\xi_j^\mu$, is computed from the trained weights and neuronal states in each memory pattern and then averaged over patterns. Coding levels are weakly correlated with total inputs in networks with independent or negatively correlated—but not positively correlated—in-degrees and coding levels (from left to right: $r = 0.134$, $p = 0.003$; $r = -0.072$, $p = 0.108$; $r = 0.23$, $p = 2 \times 10^{-7}$). (c) Coding levels are anticorrelated with the neuron activation thresholds in all three classes of networks (from left to right: $r = -0.423$, $p < 10^{-7}$; $r = -0.336$, $p < 10^{-7}$; $r = -0.543$, $p < 10^{-7}$). This ensures that the fixed-point conditions are satisfied, even when coding levels are not correlated with total inputs.

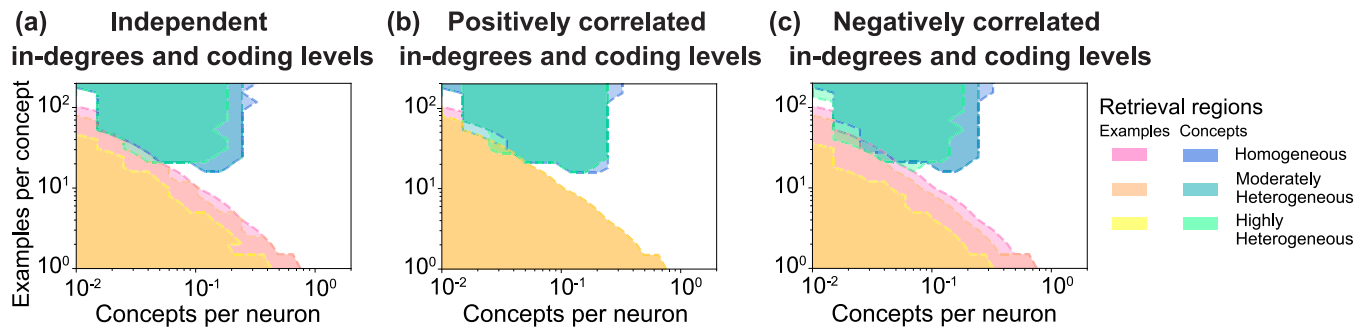


FIG. 5. Impact of heterogeneity on memory capacity in the example-concept scenario. Retrieval regions for examples (red-to-yellow) and concepts (blue-to-green) are shown in the space defined by the number of concepts per neuron (x-axis) and the number of examples per concept (y-axis). The boundaries of the retrieval regions correspond to the maximum capacities for examples and for concepts, respectively. Results are shown for three classes of networks (legend): Homogeneous (variability of in-degrees = variability of coding levels = 0, with variability defined as in Fig. 3), Moderately Heterogeneous (variability of in-degrees = 0.09, variability of coding levels = 0.07), and Highly Heterogeneous (variability of in-degrees = 0.18, variability of coding levels = 0.14). The means of in-degrees and coding levels are held constant across networks, as in Fig. 3 (see Methods). (a) When in-degrees and coding levels are independent in heterogeneous networks, the retrieval regions for both concepts and examples shrink as heterogeneity increases. (b) When in-degrees and coding levels are positively correlated in heterogeneous networks, the retrieval regions for both concepts and examples remain unchanged compared to the corresponding regions in homogeneous networks. (c) When in-degrees and coding levels are negatively correlated in heterogeneous networks, the retrieval regions for both concepts and examples shrink even more markedly than in the independent case.

In conclusion, in trained artificial networks—and possibly in biological networks as well—coding levels are primarily driven by neuron activation thresholds [Fig. 4(c)] and, to a lesser extent, by total inputs, both of which are distinct from neuron in-degrees. Our normative prediction specifies a necessary, nontrivial condition for maximizing memory capacity, which is not generally satisfied by arbitrary memory-storing networks.

We have so far focused on the regime of low coding levels ($p_i < 0.5$), which is more energy-efficient [63,64] and biologically plausible, as experimental data show that neurons tend to exhibit sparser firing rates in response to memorized objects compared to novel ones [65–68]. For completeness, we also solve the opposite (less biologically plausible) case, in which all $p_i > 0.5$. In this regime, results are reversed: network capacity is maximized when in-degrees and coding levels are negatively correlated. This reversal arises from the nonmonotonic relationship between perceptron capacity and p_i [Fig. 1(c)], whereby, for $p_i > 0.5$, perceptron capacity increases with p_i . Therefore, in this case, high coding levels must be associated with low in-degrees to maintain a roughly constant weighted perceptron capacity across neurons, ensuring a high network capacity (Fig. 4 of the SM [59]).

b. Memory capacity for correlated examples and for concepts in heterogeneous networks. While we have shown in previous sections that the perceptron capacity formula generalizes to correlated patterns, the capacity formula for Hopfield networks is derived under the assumption of independent patterns. Here, we demonstrate through numerical simulations that the network capacity remains consistent with the analytical formula even when the patterns are correlated. We generate correlated patterns or “examples” around a set of “concepts”: the concepts are independent random patterns representing prototypes, while the examples are noisy variations of their corresponding concepts and are thus correlated in clusters (see Methods). Previous research on homogeneous

Hopfield networks showed that—depending on the number of examples generated per concept and the number of concepts per neuron—these networks can learn to store either the examples or the concepts, even though they are trained only on the examples [41,42]. Here, we study how heterogeneity in in-degrees and coding levels affects memory capacity for both examples and concepts. We generate networks with three different levels of heterogeneity: fully homogeneous (uniform in-degrees and coding levels), moderately heterogeneous (intermediate variability in both parameters), and highly heterogeneous (large variability in both parameters). For each of the two nonhomogeneous classes, we generate networks with in-degrees and coding levels that are independent, positively correlated, or negatively correlated. We then train all networks to store examples. In Fig. 5, we show the retrieval regions for examples and concepts in each class of networks. The retrieval regions are the areas, in the space defined by the number of concepts per neuron and the number of examples per concept, where examples or concepts, respectively, are stored with a small error rate (see Methods). The boundaries of these regions correspond to the maximum capacities obtained from numerical simulations. Our results for homogeneous networks are qualitatively consistent with [41,42]: networks transition from storing examples to storing concepts when the number of examples per concept is large and the number of concepts per neuron is not too large. For heterogeneous networks, we find that variability in in-degrees and coding levels affects the storage of both examples and concepts in similar ways (Fig. 5). Variability shrinks both retrieval regions (i.e., reduces both capacities) when in-degrees and coding levels are independent, and even more so when they are negatively correlated. For concepts, the shrinkage occurs along the right boundary, corresponding to a reduction in the number of concepts per neuron that can be stored. In contrast, variability does not significantly affect either retrieval region when in-degrees and coding levels are positively correlated. This finding gener-

alizes the memory capacity results in Fig. 3 to the case of correlated examples and concepts.

c. Memory capacity in heterogeneous networks with dendritic structures. Lastly, we investigate the effects of heterogeneity in dendritic networks [69–71]. Here, instead of modeling neurons as simple perceptrons, we model them as treelike committee machines (TCMs) [43–49]. A TCM is a perceptron with a hidden layer between the inputs and the output. The hidden-layer units represent the neuron’s dendrites: they receive inputs from a small subset of synapses, compute intermediate values, and transmit them to the output unit. The latter integrates signals from all the dendrites and serves as the neuron’s output to the rest of the network. We define a neuron coding level as the activation probability of its corresponding TCM output unit. The connectivity structure between TCM neurons is determined by the matrix J^S , which is generated as described previously to produce heterogeneous networks (see Methods). We use backpropagation to train these networks to store memory patterns and numerically estimate their memory capacity. The results are qualitatively consistent with those shown in Fig. 3: heterogeneity in coding levels and in-degrees generally reduces memory capacity, but this reduction is mitigated when the two parameters are correlated (see Fig. 5 in the SM [59]).

2. Sparsely active units establish “quasi-indices” to optimize memory in two-layer networks

We use our analytical results to provide normative predictions on the organization of two-layer networks analogous to the CA3-DG (Cornu Ammonis 3-Dentate Gyrus) circuit in the hippocampus, known to be crucial for memory encoding [53–55,57].

a. Model definition. We consider the two-layer model in Fig. 6(a). The network structure, defined by J^S , is such that reciprocal connections are present between the two layers but not within each layer. This bipartite architecture is closely inspired by the bidirectional associative memory (BAM) model introduced by Kosko [50] and, more loosely, by restricted Boltzmann machines (RBMs) and other modern Hopfield networks used to simulate memory systems [6,8,51,52]. In our model, all units are subject to the same activation rule, Eq. (1), and the patterns are stored across both layers. The two layers differ only in the number of neurons and their coding levels, and the connections between layers do not need to be symmetric.

As suggested by theories of memory encoding and consolidation in the brain, we interpret one of the two partitions—the “feature layer”—as the storage site for the informational content of memories [53,56,72–79]. We interpret the other partition—the “memory layer”—as a memory support system that can bind feature neurons and stabilize patterns as fixed points of the dynamics. Empirical evidence suggests that, within the hippocampus, CA3 stores the informational content of experiences, while the DG acts as a teacher, providing the initial signal to bind the features in CA3 [53–55,57]. Thus, CA3 corresponds to the feature layer in our model, and DG corresponds to the memory layer. Here, we focus on the initial stage of memory encoding. Previous computational and experimental research indicates that during this stage, the strong

mossy-fiber inputs from the DG to CA3 dominate over the local recurrent-collateral inputs within CA3 [57,80,81]. This motivates the bipartite architecture of our model, where the internal CA3 connectivity is omitted.

If and how memory function is optimized in this class of networks is an area of active research. A prominent hypothesis, known as hippocampal index theory (HIT), suggests that neural units in the memory layer function as “indices” during early encoding to stabilize the memory patterns. HIT posits that each index binds all the feature neurons active in a memory so that, starting from partial cues (an incomplete set of active features), a complete memory can be retrieved by reactivating its index, a property called pattern completion [54,56,76]. HIT also posits that each index selectively binds the features of only one memory, allowing pattern separation between different memories [53,55–57,72,73]. However, several questions remain unanswered: Are neural indices with these properties required to maximize memory capacity? Are neural indices also required to maximize the information encoded in the feature layer? In a memory code that is entirely index-based, ablating an index would destroy an entire memory—so how can an indexing mechanism be compatible with a robust memory function? How do these aspects depend on neuron coding levels, as well as on the relative and absolute sizes of the two layers? The following analyses provide insights into these questions.

b. Dependence of memory and information capacities on coding levels and total network size. In the two-layer network shown in Fig. 6(a), we set the number of units in the feature and memory layers (N_1 and N_2 , respectively) to a 1/3 ratio, based on biological estimates of cell counts in CA3 and DG in rodents [82,83]. While keeping this ratio fixed, we vary the total number of units, $N = N_1 + N_2$. We also independently vary the coding levels in the two layers. As a function of coding levels, we compute the memory capacity using the analytical formula for $N \rightarrow \infty$, Eqs. (7) and (3), and numerical simulations for finite N . We also compute the information capacity, defined as the entropy per synapse of the feature-neuron states in the stored patterns (see Methods). High information capacity indicates that the feature-layer neurons exhibit a greater degree of variability and can therefore represent a wider set of distinct features, encoding more information. Convergent results from analytical and numerical estimates indicate that (a) memory capacity is maximized when the coding level is low or high in both layers [Fig. 6(b), top]; (b) information capacity is maximized when the coding level in the memory layer is low or high and the coding level in the feature layer is intermediate [Fig. 6(b), middle]; (c) memory and information capacity are more sensitive to the coding level in the memory layer compared to the feature layer; (d) as a consequence, to jointly maximize memory and information capacity (i.e., their product), the coding level must be low or high in the memory layer, whereas it is relatively unconstrained in the feature layer [Fig. 6(b), bottom]. Mathematically, the two optimal solutions are symmetric; however, biologically, they are not: maximizing memory capacity through low or high coding levels corresponds to encoding information primarily using the +1 (active) or −1 (silent) states, respectively. Therefore, only the first solution (low coding levels) is biologically meaningful, as information

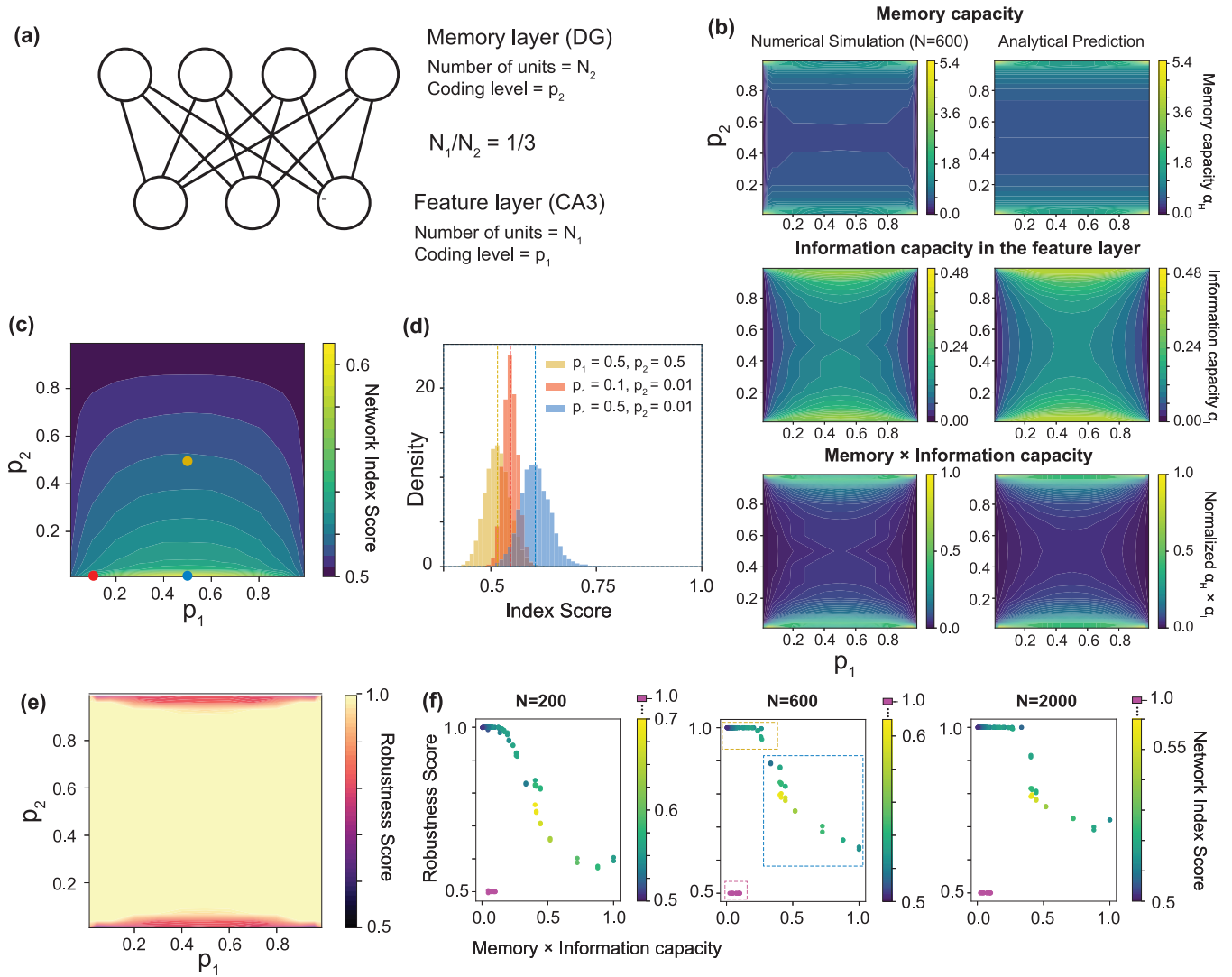


FIG. 6. Optimizing memory in two-layer networks. (a) Sketch of a two-layer network, simulating the CA3-DG circuit. The feature layer (CA3) has fewer units than the memory layer (DG). We set $N_1/N_2 = 1/3$. The coding levels in the two layers are p_1 and p_2 , respectively. (b) Estimates of memory (top), information (middle), and memory $\times$ information (bottom) capacity (the latter normalized by its maximum value) as a function of p_1 and p_2 : numerical estimates for $N = N_1 + N_2 = 600$ (left) and analytical estimates for $N \rightarrow \infty$ (right). We exclude the region where the analytical memory capacity diverges and numerical estimates become inaccurate ($p_1 < 0.01$ or $p_1 > 0.99$, and similar for p_2). Both memory and information capacities are more sensitive to p_2 , and both are maximized when p_2 is either low or high. (c) The Network Index Score varies with p_1 and p_2 . For trained networks, it reaches its maximum when p_2 is low, which aligns with the biologically meaningful condition of maximum memory and information capacity. (d) Distributions of Index Scores for active memory units in stored patterns are shown for different p_1 and p_2 (legend), corresponding to the same-colored dots in (c). Dashed lines indicate the means. When p_2 is low, most Index Scores are above 0.5 but below 1, indicating a quasi-indexing function for the memory units. (e) The Robustness Score is minimized—but remains higher than 0.5—in trained networks with low p_2 , whereas it is uniformly maximized for all other coding levels ($N = 600$). (f) Tradeoffs between joint memory $\times$ information capacity and Robustness Score, for $N = 200, 600, 2000$. Robustness increases with N , and is consistently $\gtrsim 0.7$ for $N = 2000$. Each dot represents a network with a distinct pair of coding levels ($0 < p_1 < 1, 0 < p_2 \leq 0.5$), and the color indicates the Network Index Score. The purple dots correspond to artificially generated configurations with a Network Index Score = 1 ($p_2 = 1/N_2$ while p_1 varies). Blue and yellow dotted rectangles indicate networks with $p_2 \ll 0.5$ and $p_2 \sim 0.5$, respectively. The first condition corresponds to the neighborhood of the blue dot in (c), extending toward the red dot in (c) as one moves toward the right extreme of the tradeoff. The second condition corresponds to the neighborhood of the yellow dot in (c), extending toward the blue dot in (c) as one moves toward the center of the tradeoff.

is typically encoded and transmitted by active neurons in the brain.

While these general results hold for both infinite-size and finite-size networks [Fig. 6(b), compare right and left],

differences arise in limiting cases, when the coding levels approach extreme values (0 or 1; however, since high coding levels are not biologically meaningful, we only consider the first limit hereafter). We analyze these differences both

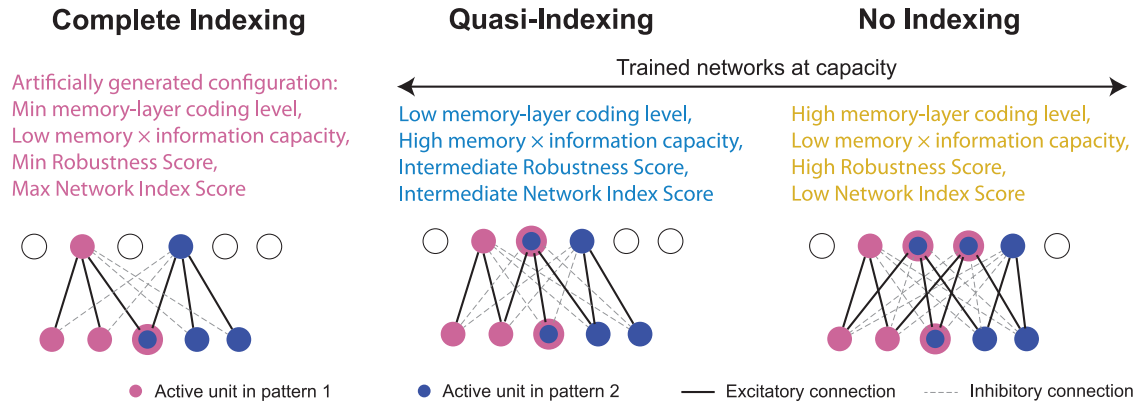


FIG. 7. Illustration of three qualitatively distinct encoding schemes. Left: complete-indexing scheme, where each memory unit binds the complete set of features of one pattern. Middle: quasi-indexing scheme, where each memory unit binds subsets of features from a small number of patterns. Right: nonindexing scheme, where each memory unit binds seemingly random features across patterns. The three schemes correspond to the conditions indicated by the purple, blue, and yellow dotted rectangles in Fig. 6(f), respectively. The schemes differ in memory-layer coding level, joint memory $\times$ information capacity, Robustness Score, and Network Index Score. The quasi-indexing scheme achieves higher capacity than both alternative schemes while maintaining robustness, particularly in large networks. We thus argue that the brain operates under a quasi-indexing scheme.

for completeness and because they are relevant for assessing the robustness of memory encoding in the meaningful high-capacity region of the coding-level space. As coding levels in both layers approach 0, the memory capacity and the joint memory $\times$ information capacity of infinite-size networks diverge to infinity, whereas they remain finite in finite-size networks, reaching a peak at small coding levels. As the coding level in the memory layer approaches 0 while the coding level in the feature layer is held constant, information capacity increases monotonically (without diverging) in infinite-size networks, whereas it reaches a peak at small but finite memory-layer coding levels in finite-size networks. The precise changes in optimal coding levels with N are illustrated in Fig. 6 of the SM [59] for each type of capacity. The fact that optimal coding levels are low—even in finite size-networks—raises the question of whether memory units form sparse indices, as posited by HIT, and whether such a sparse code can store patterns robustly. In the following sections, we quantify both the indexing and robustness of networks at capacity as coding levels vary, and we examine the tradeoffs that arise in the high-capacity coding region.

c. Index scores of different encoding schemes. To quantify indexing, we examine the network configuration (the set of J_{ij}) that stores memory patterns at capacity for each pair of coding levels. Given a network configuration, we define an “Index Score” for each active memory unit in each stored pattern. This score quantifies the extent to which the memory unit forms excitatory connections with the feature units active in the same pattern (binding them together) while forming inhibitory connections with the inactive feature units. The score is averaged over both weight directions—from feature units to the memory unit and vice versa (details in Methods)—in line with experimental evidence of bidirectional information flow between CA3 and DG [81,84]. A high index score for a memory unit in a pattern indicates that the unit tends to bind all and only the features of that pattern. A low index score indicates the opposite: the unit tends to bind subsets of features of multiple patterns and is thus not clearly designed

to perform pattern completion and separation. We define the Network Index Score as the mean index score of all active memory units in a stored pattern, averaged across patterns.

We find that, among all networks trained at capacity, those that have high or moderate Network Index Scores also have low coding levels in the memory layer [Figs. 6(c) and 6(d), blue and red dots and distributions]. This condition aligns with the biologically meaningful solution that jointly optimizes memory and information capacity [Fig. 6(b), bottom]. For higher coding levels, the Index Scores of memory units are centered around 0.5 (chance level), meaning that those units do not support any indexing function [Figs. 6(c) and 6(d), yellow dot and distribution]. We note that, even under optimal conditions of sparse coding in the memory layer, the Network Index Score—while reaching its relative maximum for trained networks—remains significantly lower than 1 (the maximum theoretical value). Thus, memory units in networks that maximize capacity function as “quasi-indices”: they have higher Index Scores than in suboptimal networks (with denser coding in the memory layer), but lower Index Scores than the ideal maximum. These results are summarized in Fig. 7, which illustrates three qualitatively distinct scenarios: (i) an artificially generated configuration with Network Index Score = 1; this configuration maximizes the indexing function of every unit but is never observed in networks trained at capacity; (ii) a configuration with high (but lower than 1) Network Index Score, corresponding to a trained network with sparse coding in the memory layer, and high joint memory and information capacity; (iii) a configuration with low Network Index Score, corresponding to a trained network with dense coding in the memory layer and low joint capacity.

The patterns of memory capacity, information capacity, and Index Scores across coding levels remain qualitatively similar in networks where the memories are required to have finite basins of attractions ($\kappa > 0$, Fig. 7 of the SM [59]).

These conclusions highlight the first limitation of an encoding mechanism that strictly maximizes indexing: in configurations with Network Index Score = 1, each unit in the

memory layer binds the features of a distinct pattern, limiting memory capacity to N_2/N (the number of memory-layer units normalized by the total number of units). This number is substantially lower than the maximum attainable capacity in networks of the same size. A second limitation of a strict indexing mechanism is that it makes memory highly sensitive to perturbations: when each memory neuron binds the features of a distinct pattern, the loss of that single neuron results in the complete loss of the corresponding memory pattern (Fig. 7, left). We quantify the robustness of different encoding schemes in the next section.

d. Robustness of different encoding schemes to neuron ablation. We quantify the stability of each encoding scheme against neuron ablation in the memory layer by computing the Robustness Score. This is defined for each network as the mean fraction of feature units that maintain the same (active or silent) state after ablating one memory unit active in a given pattern, averaged over memory units and patterns (see Methods). Not surprisingly, the score indicates that a network is less robust when the coding level in the memory layer is either extremely low or extremely high [Fig. 6(e)], whereas it is uniformly robust in all other coding-level regions. By comparing Figs. 6(e) and 6(b), bottom, we observe a tradeoff between network robustness and its joint memory and information capacity. Figure 6(f) illustrates this tradeoff across networks with varying coding levels and sizes N . The plots reveal several key insights.

First, for finite N , the joint capacity is maximized and robustness is minimized—relative to the class of trained networks—when coding levels in both layers approach a small but finite value (numerically estimated in Fig. 6 of the SM [59] for different N). This condition corresponds to the neighborhood of the red dot in Fig. 6(c). Importantly, in this condition, the Robustness Score is not minimized in an absolute sense, and it remains significantly higher than 0.5—the absolute minimum corresponding to the erasure of a memory upon ablation of its index. This absolute minimum is attained in artificially generated configurations with Network Index Score = 1 [purple dots in Fig. 6(f)], which also have very low memory $\times$ information capacity, but it is never realized in networks trained to maximize memory capacity given their coding levels. Furthermore, as N increases, the minimum Robustness Score for trained networks rises, reaching intermediate values (~ 0.7) even for networks of moderate size. Thus, we estimate that real brain networks can achieve relatively high robustness while simultaneously maximizing memory $\times$ information capacity. At this extreme point of the tradeoff, the Network Index Score is moderate, yet still higher than 0.5 (the score for random configurations). Starting from this condition, we analyze the right branch of the tradeoff in Fig. 6(f). Along this branch, the joint capacity progressively decreases but remains relatively high, while both the Robustness and Network Index Scores increase as the coding level in the feature layer approaches 0.5, with the coding level in the memory layer remaining low. At this point, corresponding to the neighborhood of the blue dot in Fig. 6(c), the Network Index Score reaches its relative maximum among trained networks. Along this branch, we also observe another tradeoff between memory and information capacity (as previously discussed): memory capacity decreases while information capacity increases, resulting in an overall decrease

in joint memory $\times$ information capacity. However, despite this tradeoff, both capacities remain relatively high for all conditions with low coding levels in the memory layer.

Finally, starting from the condition of maximum Network Index Score (for trained networks), the joint capacity continues to decrease along with the Network Index Score, while the Robustness Score rapidly increases as the coding level in the memory layer increases, in the direction of the yellow dot in Fig. 6(c). This corresponds to the left branch of the tradeoff in Fig. 6(f).

e. Overall optimal scheme for memory function. In conclusion, a sparse code in the memory layer (DG) concurrently supports high memory and information capacity, as well as intermediate levels of robustness to neuron ablation in large networks—relative to the minimum and maximum attainable levels. The network weight configurations in these conditions reflect a “quasi-indexing” function for the memory layer, where active units tend to bind subsets of features from a small number of patterns: this mechanism increases both capacity and robustness compared to maximum or complete indexing, where each memory unit binds the complete set of features of one pattern, and it increases capacity compared to no indexing, where each memory unit binds seemingly random features across patterns (Fig. 7). Thus, we argue that the brain operates under a quasi-indexing coding scheme. In the feature layer (CA3), a sparse code enhances memory capacity whereas a denser code increases information capacity. However, both capacities remain relatively high as long as the memory layer maintains low coding levels, consistent with experimental evidence of a sparse code in the DG [85,86].

f. The alignment between maximum capacity and indexing depends on the relative sizes of the two layers. An important factor in the model architecture of Fig. 6(a) is that the memory layer contains more neurons than the feature layer. If this is reversed, the memory capacity becomes more sensitive to the coding level in the feature layer, and the conditions that jointly maximize memory and information capacity no longer overlap with those of high Network Index Score (Fig. 8 of the SM [59]). This result supports the hypothesis that memory indices are stored in brain areas with more neurons than the regions storing the memory features. Thus, our result argues against an indexing scheme where the hippocampus directly indexes entire cortical (feature) modules. Instead, it supports a hierarchical scheme, where the DG stores high-order indices (i.e., indices of indices) and CA3 stores the features of those indices, which may in turn be part of a network that forms lower-order indices of features stored in cortical modules [53–55,72,74,87].

III. DISCUSSION

We have generalized the classical Gardner’s calculation to derive an analytical formula for the capacity of perceptrons with arbitrary activation probability of the output unit and arbitrary input distributions—provided the inputs have finite means and variances and are either independent or linearly correlated. We found that the perceptron capacity depends only on the output mean (m^{out}). Closely related work includes Refs. [29,34,35], which demonstrated that—in the general case of finite stability ($\kappa > 0$)—perceptron capacity is independent of the input distributions as long as the inputs

share the same covariance. These studies also showed that capacity becomes entirely independent of the input statistics, including the input covariances, in the limit of zero stability, a result that can be directly related to ours. However, these studies assumed a constant and unbiased mean activation of the output unit ($m^{\text{out}} = 0$). This limitation prevents their direct application to Hopfield networks with arbitrary coding levels. A consistent result for the upper bound, $\alpha_p = 2$, was also obtained using a different geometrical approach, based on the assumption that the patterns are in a general position, i.e., no subset of N or fewer patterns are linearly dependent [27,88]. However, this approach does not apply to cases with biased outputs ($m^{\text{out}} \neq 0$) and finite classification stability ($\kappa \neq 0$). Here, we have generalized these capacity results to simultaneously account for both heterogeneous input distributions and arbitrary output means. This generalization allowed us to derive an analytical formula for the memory capacity of Hopfield networks with heterogeneous coding levels. Furthermore, our formula applies to Hopfield networks with arbitrary architectures. This extends previous work, which primarily addressed homogeneously connected Hopfield networks—whether dense or diluted—with continuous or discrete synapses and binary or threshold-linear units [6,18–24,89]. The case of heterogeneous networks was previously addressed in [6,25,26], but only for graphs with symmetric weights (learned by symmetric Hebbian learning) and a constant coding level across all neurons.

While we considered networks with $(-1, 1)$ units, our results generalize to $(0, 1)$ units. Indeed, in a perceptron, replacing the -1 labels with 0 does not affect the linear separability of patterns, and changing the input states in a similar way only alters the input-state distributions, which do not affect the capacity. The capacity of a Hopfield model is defined as the minimum capacity of the perceptrons in the network, and is therefore also invariant to this transformation. The formula (7) also implies that in two-layer networks with only interlayer connectivity, the capacity must be linear in the number of units of one of the layers (specifically, the input layer of the perceptron with minimum capacity). This is consistent with the scaling of memory capacity found for modern Hopfield networks [8,9].

Our analytical result for Hopfield networks indicates that memory capacity depends on both the neuron coding levels and in-degrees, but not on their out-degrees.

Specifically, we found that, in general, memory capacity decreases as heterogeneity in coding levels and in-degrees increases. This result is broadly consistent with previous findings that maximizing memory capacity in networks with positive weights and homogeneous coding levels leads to lower variance in in-degrees compared to out-degrees [7]. It also aligns with the known decrease in memory capacity as correlations among patterns increase [36], since coding levels different from $p_i = 0.5$ introduce correlations between patterns. We also found that the reduction in memory capacity due to heterogeneity is mitigated under specific conditions determined by the joint statistics of coding levels and in-degrees. In particular, in a low-coding regime ($p_i < 0.5$), which is more energy-efficient and biologically plausible [63–68], coding levels and in-degrees should be positively correlated to maximize memory capacity. This result represents a nontrivial

normative prediction for real brain networks, which could be tested experimentally. While direct measurements of synaptic connectivity are generally unfeasible *in vivo*, several methods exist to infer functional or statistical interactions between cells from spike data. Some of these methods [90–92] yield sparse interaction graphs by carefully avoiding overfitting and could thus be used to estimate neuron in-degrees.

Our prediction generalizes to other biologically relevant cases. One case is the problem of storing “examples”, i.e., patterns clustered around prototypes or “concepts”. Our first numerical finding is that heterogeneous networks store examples when the number of examples per concept and the number of concepts per neuron are not too large. At high numbers of examples per concept, these networks switch to storing concepts, even when trained only on examples. This behavior is qualitatively similar to that observed in homogeneous networks [41,42]. Quantitative differences, however, arise due to three key distinctions from [41,42]: we introduced heterogeneity, we did not restrict learning to be Hebbian, and we used a more stringent definition of capacity. In particular, we required that patterns be stored with an error rate of at most 0.01, whereas Refs. [41,42] allowed error rates of approximately 0.25. Our second finding is that heterogeneity affects both the example and concept capacities, similar to the case of storing independent patterns: it generally reduces both capacities unless coding levels and in-degrees are correlated.

Another biologically inspired scenario involves storing memories in networks where neurons have treelike architectures that simulate their dendritic arbors [43–49,69,70]. Using treelike committee machines as neuron models in our networks, we numerically showed that heterogeneity affects capacity in a manner similar to classical networks, where neurons have only a single dendrite. A more detailed analytical investigation of the interactions between heterogeneity and dendritic parameters in networks of treelike committee machines is an interesting direction that we leave for future research.

Finally, we investigated the capacity of bipartite networks composed of two reciprocally connected layers that are not internally connected, similar to the BAM model [50]. We used this model to simulate interactions between CA3 and DG in the hippocampus during the initial stage of memory encoding. The two layers differ in the number of neurons and coding levels. We found that memory and information capacities are maximized when coding is sparse in the layer with more neurons (the memory layer, representing DG). Under this condition, neurons active in that layer exhibit a quasi-indexing function, binding subsets of features from a few patterns in the other layer (the feature layer, representing CA3). This result aligns broadly with the HIT and recent cognitive map theories [93–96]. At the same time, our findings highlight the limitations of a complete-indexing scheme: when each neuron in the memory layer binds the full set of features of a distinct pattern, both memory capacity and robustness drop. In contrast, a quasi-indexing mechanism maximizes capacity and maintains some robustness to neuron ablation, particularly when the number of neurons is large, as in the brain.

The DG and CA3 have properties similar to those of our model at maximal capacity. Based on biological estimates in rodents, the DG contains three to five times

more neurons than CA3 [82,83] and has sparse coding levels [97]. This correspondence suggests that the DG stores quasi-indices of memories initially encoded in CA3, supporting both pattern completion and pattern separation in the latter [54,56,76,86,98]. Our results argue against an indexing scheme in which hippocampal neurons directly bind features in the cortex. In networks where the feature layer contains more neurons than the memory layer—as suggested by a cortical-hippocampal model—we found that indexing does not maximize capacity. Instead, our results support a hierarchical indexing scheme, where the DG binds features in CA3, which may in turn be part of a broader network that binds features in the cortex [53]. Additionally, the DG indexing function may be supported by adult neurogenesis of granule cells, providing a mechanism for updating indices as new memories are continuously formed [86,98–101].

Overall, our findings advance the understanding of memory in heterogeneous networks and introduce new dimensions to the HIT of memory encoding and retrieval, while suggesting specific neural substrates for these functions.

IV. MATERIALS AND METHODS

A. Numerical estimation of perceptron capacity

For perceptrons with a finite number of input units N , we generate P random pattern-label pairs and train the perceptron to store the pairs using the “sklearn.svm.LinearSVC” function in the scikit-learn package [102], which implements the support vector machine (SVM) algorithm for pattern classification. We use the binary search algorithm to find the maximum number of pattern-label pairs P that the perceptron can store with an error rate of 0.01 or less. We then compute the mean and standard deviation of the maximum number of patterns over 10 iterations of this process. In all numerical estimates in Fig. 1, we use $N = 1000$.

The analytical capacity for $N \rightarrow \infty$ is given by solving Eqs. (3) with numerical methods provided by the “scipy.optimize.root” function [103].

B. Gaussian Markov sampling

For the perceptron with correlated inputs [Fig. 1(g) and 1(h)], we sample the input states from a Gaussian Markov chain according to the following formula:

$$\begin{aligned}\xi_1^\mu &= \epsilon_1, \\ \xi_j^\mu &= \rho \xi_{j-1}^\mu + \sqrt{1 - \rho^2} \epsilon_j \quad \text{for } j > 1,\end{aligned}$$

where each ϵ_j is drawn from a standard normal distribution.

C. Numerical estimation of Hopfield model capacity

The Hopfield model is formally equivalent to N independent perceptrons. Each perceptron i is trained to associate label ξ_i^μ to pattern $\{\xi_j^\mu | J_{ij}^S \neq 0\}$ using the “sklearn.svm.LinearSVC” function. We use the binary search algorithm to find the maximum number of memory patterns that the network can store with an error rate of 0.01 or less. The error rate is defined as the fraction of units (perceptrons) that fail to satisfy the classification constraint for a given pattern, averaged over all patterns. We then compute the mean

and standard deviation of the maximum number of patterns over 10 iterations of this process.

In simulations of the randomly connected heterogeneous networks in Fig. 3, we use $N = 500$. In simulations of the two-layer network in Fig. 6, we use $N_1 = 150$ for the feature layer and $N_2 = 450$ for the memory layer. In Fig. 6(f), we add a case with $N_1 = 50$ and $N_2 = 150$ and a case with $N_1 = 500$ and $N_2 = 1500$.

D. Generation of heterogeneous networks

To generate the structure J^S of the randomly connected heterogeneous networks shown in Fig. 3, we use the following procedure: we assign an inward-connection probability p_i^{conn} to each neuron i ; p_i^{conn} is sampled from a uniform distribution with mean 0.5; we then set J_{ij}^S to 1 with probability p_i^{conn} and to 0 with probability $1 - p_i^{\text{conn}}$ for every j . We also assign a coding level p_i to each neuron i ; p_i is sampled from another uniform distribution with mean 0.25. To generate networks with different levels of heterogeneity, we vary the standard deviations of the uniform distributions of p_i^{conn} and p_i by adjusting their widths while keeping their means constant. This adjustment allows or restricts certain coding levels depending on the width of the distribution. To enforce a positive correlation between coding levels and in-degrees, we permute the coding levels p_i so that the neuron with the lowest p_i is associated with the lowest p_i^{conn} , the neuron with the second lowest p_i is associated with the second lowest p_i^{conn} , and so on. We use the inverse procedure (associating low p_i with high p_i^{conn}) to enforce a negative correlation between coding levels and in-degrees.

E. Generation and storage of concepts and examples

To generate concepts and examples, we adapt the method used in [41,42]. Concepts ξ^μ , $\mu = 1, \dots, P_c$, are generated as random patterns, constrained only by the neuron coding levels: $\xi_i^\mu = 1$ with probability p_i , and $\xi_i^\mu = -1$ with probability $1 - p_i$, where p_i is the coding level of neuron i . Coding levels are sampled from a uniform distribution with mean 0.25 and varying standard deviations (Fig. 5). Structure matrices J^S are constructed as previously defined for all other heterogeneous networks.

We then generate a set of examples $\xi^{\mu\nu}$, $\nu = 1, \dots, P_e$, for each concept ξ^μ as follows: we initialize $\xi^{\mu\nu} = \xi^\mu$; then, with probability $p^{\text{regen}} = 0.3$, we resample the state of each neuron, consistent with its coding level as defined above. This yields

$$\xi_i^{\mu\nu} = \begin{cases} \xi_i^\mu & \text{with probability } 1 - p^{\text{regen}}, \\ 1 & \text{with probability } p^{\text{regen}} p_i, \\ -1 & \text{with probability } p^{\text{regen}} (1 - p_i). \end{cases} \quad (8)$$

For each heterogeneity level, we generate 10 networks and train them to store only the examples, using the same SVM-based algorithm described previously. We vary the number of concepts P_c and the number of examples per concept P_e to be stored.

To evaluate retrieval of both examples and concepts, we assess the stability of neurons in the corresponding patterns. If the average error rate (i.e., the fraction of unstable neurons) across examples and networks with a given heterogeneity

level is less than 0.01, we consider those networks capable of storing the examples. Similarly, if the average error rate across concepts and networks with the same heterogeneity level is less than 0.01, we consider those networks capable of storing the concepts.

F. Dendritic neural networks

To generate dendritic neural networks, we model each neuron as a treelike committee machine (TCM) rather than a simple perceptron. A TCM is a perceptron comprising one output unit (whose activation probability defines the neuron coding level), K hidden units (representing the neuron's dendrites), and N input units. Each input unit projects to exactly one hidden unit, producing a treelike structure. In the classical TCM, every hidden unit receives inputs from N/K input units. Here, to accommodate heterogeneity, we allow each hidden unit to receive inputs from a different number of input units while still preserving the treelike structure (i.e., each input unit connects to at most one hidden unit). The input units correspond to other neurons in the network. We fix the number of hidden units to $K = 3$ for all TCM neurons. The presence or absence of a connection from neuron j to neuron i is determined by the structure matrix J^S , generated as described in Sec. IV D. If neuron j projects to neuron i (i.e., $J_{ij}^S = 1$), then the output of neuron j is treated as one of the inputs to neuron i , and is assigned to one of the three hidden units (dendrites) of neuron i , chosen uniformly at random.

The weights from the hidden units to the output unit are fixed at 1 and the threshold of the output unit is fixed at 0, whereas the weights from the input units to the hidden units and the thresholds of the hidden units are trained to store the memory patterns. We train the networks to store random patterns as attractors using backpropagation with the ADAM optimizer. To facilitate gradient flow during training, we use the hyperbolic tangent as the activation function for the neurons. To estimate the memory capacity of each dendritic network, we employ the binary search algorithm to determine the maximum number of patterns that the network can store with an error rate of 0.01 or less. We use the sign activation function for the neurons to assess the stability of the patterns.

G. Information capacity in the feature layer

For the two-layer network in Fig. 6, we denote the coding level in the feature layer by p_1 and the neuron numbers in the feature and the memory layer by N_1, N_2 , respectively.

The information capacity (per synapse) is calculated as the entropy of the states of feature neurons across all memory patterns, normalized by the total number of possible synapses in a network of size $N_1 + N_2$:

$$\begin{aligned} \alpha_I &= \sum_{\mu=1}^P \sum_{i=1}^{N_1} H(p_1)/(N_1 + N_2)^2 \\ &= \alpha_H \frac{N_1}{(N_1 + N_2)} (-p_1 \log_2 p_1 - (1 - p_1) \log_2 (1 - p_1)). \end{aligned} \quad (9)$$

H. Index Score

For the two-layer network in Fig. 6, we denote the portions of each pattern stored in the feature and in the memory layer as $\xi^{1,\mu}$ and $\xi^{2,\mu}$, respectively.

We binarize the inward and outward weights of unit i in the memory layer using the formulas $\omega_{ij}^{21} = \text{sgn}(J_{ij})$, $\omega_{ji}^{12} = \text{sgn}(J_{ji})$, so that $\omega_i^{21} = (\omega_{ij}^{21}, j = 1, \dots, N_1)$ and $\omega_i^{12} = (\omega_{ji}^{12}, j = 1, \dots, N_1)$ are the binarized weight vectors from the feature layer to memory unit i and from memory unit i to the feature layer, respectively.

For each stored pattern $(\xi^{1,\mu}, \xi^{2,\mu})$ and each memory-layer unit that is active in that pattern, we compute the similarity from the feature layer to the memory layer as the fraction of connection weights toward the memory unit whose signs match the corresponding feature unit states. This fraction is computed separately for each of the two possible feature unit states—active and silent—and averaged over states:

$$s^{21}(i, \mu) = \frac{1}{2} \left(\frac{\sum_j \delta_{\omega_{ij}^{21}, 1} \delta_{\xi_j^{1,\mu}, 1}}{\sum_j \delta_{\omega_{ij}^{21}, 1}} + \frac{\sum_j \delta_{\omega_{ij}^{21}, -1} \delta_{\xi_j^{1,\mu}, -1}}{\sum_j \delta_{\omega_{ij}^{21}, -1}} \right), \quad (10)$$

where δ denotes the Kronecker delta. We compute the similarity from the memory layer to the feature layer in a similar way:

$$s^{12}(i, \mu) = \frac{1}{2} \left(\frac{\sum_j \delta_{\omega_{ji}^{12}, 1} \delta_{\xi_j^{1,\mu}, 1}}{\sum_j \delta_{\omega_{ji}^{12}, 1}} + \frac{\sum_j \delta_{\omega_{ji}^{12}, -1} \delta_{\xi_j^{1,\mu}, -1}}{\sum_j \delta_{\omega_{ji}^{12}, -1}} \right). \quad (11)$$

We define the Index Score of each active memory unit i in each stored pattern μ as the average of the similarities in the two directions:

$$\text{Index Score}(i, \mu) = \frac{s^{21}(i, \mu) + s^{12}(i, \mu)}{2}. \quad (12)$$

An Index Score of 0.5 indicates that the match between weight signs and feature states is at chance level, meaning the memory unit does not exhibit any indexing function for that pattern. An Index Score of 1 indicates maximum indexing.

We define the Network Index Score as the mean of the Index Score over active memory units in each pattern and across patterns:

$$\text{Network Index Score} = \langle \text{Index Score}(i, \mu) \rangle_{i, \mu}. \quad (13)$$

I. Robustness Score

As for Index Score, we define the portions of an intact pattern in the feature and memory layers as $(\xi^{1,\mu}, \xi^{2,\mu})$. For each pattern, we then ablate one active unit in the memory layer (setting its state to 0) and allow the network to evolve from this lesioned pattern to a retrieved pattern $(\xi'^{1,\mu}, \xi'^{2,\mu})$. If the ablation does not affect the stability of the feature units, then $\xi'^{1,\mu} = \xi^{1,\mu}$; otherwise some feature units will flip their states.

We define the robustness $R(i, \mu)$ of pattern μ against the ablation of memory unit i (active in that pattern) as the fraction of feature units that do not flip their states, conditioned on each of the two possible signs (active and silent) and averaged

over both:

$$R(i, \mu) = \frac{1}{2} \left(\frac{\sum_j \delta_{\xi_j^{1,\mu}, 1} \delta_{\xi_j^{1,\mu}, 1}}{\sum_j \delta_{\xi_j^{1,\mu}, 1}} + \frac{\sum_j \delta_{\xi_j^{1,\mu}, -1} \delta_{\xi_j^{1,\mu}, -1}}{\sum_j \delta_{\xi_j^{1,\mu}, -1}} \right). \quad (14)$$

The reason we separate active and silent states in the computation is to avoid meaningless dependencies of robustness on coding levels in the feature layer. For example, when only one unit is active in the memory layer in a given pattern (Fig. 7, left), ablating that unit completely destroys the memory, regardless of the coding level in the feature layer. Consistently, our definition yields a minimum and constant robustness of 0.5.

Finally, we define the Robustness Score for the network as the average $R(i, \mu)$ over ablated units and patterns:

$$\text{Robustness Score} = \langle R(i, \mu) \rangle_{i, \mu}. \quad (15)$$

J. Generation of artificial configurations with Network Index Score = 1 and Robustness Score = 0.5

Artificial configurations with Network Index Score = 1 [Fig. 6(f), purple dots, and Fig. 7, left] are generated as follows. First, in the memory-layer portion of each pattern, $\xi^{2,\mu}$, exactly one unit is set to be active, with different memory units active across different patterns. The states of the units in the corresponding feature-layer portion, $\xi^{1,\mu}$, are randomly generated with a coding level p_1 . The upper bound on the number of patterns that can be stored in these networks is the number of units in the memory layer (N_2). We denote the pattern in which memory unit i is active as $\{\xi^{1,i}, \xi^{2,i}\}$, where i ranges from 1 to N_2 across patterns.

To obtain the maximum Network Index Score of 1, for each unit i in the memory layer and all units j in the feature layer, we set the weights $J_{ij} = J_{ji} = \xi_j^{1,i}$. We also set the threshold of memory unit i to $\max_\mu (\sum_j J_{ij} \xi_j^{1,\mu}) - 0.5$, and the threshold of feature unit j to $\max_\mu (\sum_i J_{ji} \xi_i^{2,\mu}) - 0.5$.

It is straightforward to verify that networks designed in this way store the memory patterns $\{\xi^{1,i}, \xi^{2,i}\}$ as attractors, have Network Index Score = 1, and Robustness Score = 0.5.

ACKNOWLEDGMENTS

The authors thank Edward Han for fruitful discussions on the neural substrates of the hippocampal index theory, and Rémi Monasson, Łukasz Kuśmierz, Naoki Hiratani, and Ilya Monosov for useful comments on the manuscript. G.T. acknowledges support from the Alfred P. Sloan Foundation under Grant No. FG-2024-22163.

K.Z. and G.T. developed the ideas and conceptual framework. K.Z. performed research. K.Z. and G.T. wrote and revised the manuscript. G.T. provided supervision and funding.

DATA AVAILABILITY

The computational code that supports the findings of this article is openly available on GitHub [104].

APPENDIX A: ANALYTICAL CALCULATION OF THE PERCEPTRON CAPACITY

We present here the complete calculation of the maximal capacity of a perceptron with arbitrary (independent) distributions of input states and output labels.

1. Definition of the problem

We define P input patterns $\xi^\mu = \{\xi_1^\mu, \dots, \xi_N^\mu\}$, where $\mu = 1, \dots, P$ indexes the pattern, and each input state ξ_j^μ ($j = 1, \dots, N$) is sampled from an arbitrary distribution with finite mean m_j and variance σ_j^2 . We do not impose any constraint on these distributions insofar as the states of different neurons are statistically independent. Each pattern is associated with a label χ^μ . The labels are drawn from a Bernoulli distribution with mean m^{out} : $p(\chi^\mu = 1) = (1 + m^{\text{out}})/2$ and $p(\chi^\mu = -1) = (1 - m^{\text{out}})/2$.

We denote the normalized capacity $\alpha_p = C_p/N$, where C_p is the maximal number of pattern-label pairs that the perceptron can learn to classify, according to the classification constraints

$$\frac{\chi^\mu}{\sqrt{N}} \left(\sum_j J_j \xi_j^\mu \right) > \kappa \quad (A1)$$

for every input pattern indexed by μ ; $\kappa \geq 0$ is the stability parameter [Fig. 1(b)].

To calculate α_p , we define the following elliptical regularization on the perceptron synaptic weights $\{J_j\}$:

$$\sum_{j=1}^N \sigma_j^2 J_j^2 = N. \quad (A2)$$

This constraint can be viewed as a normalization of the post-synaptic input ranges. When all inputs have the same variance, Eq. (A2) reduces to the standard spherical regularization used by Gardner [18]. As discussed in Appendixes B and C, the difference between our elliptical and Gardner's spherical constraint is relevant only when the stability parameter κ in Eq. (A1) is strictly positive. When $\kappa = 0$, any solution that satisfies (A2) can be mapped to exactly one solution that satisfies a spherical constraint, or any other elliptical constraint with coefficients different from σ_j^2 . Thus, when $\kappa = 0$, the maximal capacity is invariant to changes in the coefficients of constraint (A2).

Following the method introduced by Gardner [18], we define the fractional volume of the weights that satisfy both conditions (A1) and (A2) in the space of weights constrained by (A2):

$$V = \frac{1}{V_0} \int \prod_j dJ_j \delta \left(\sum_j \sigma_j^2 J_j^2 - N \right) \times \prod_\mu \Theta \left(\frac{\chi^\mu}{\sqrt{N}} \sum_j J_j \xi_j^\mu - \kappa \right), \quad (A3)$$

where

$$V_0 = \int \prod_j dJ_j \delta \left(\sum_j \sigma_j^2 J_j^2 - N \right).$$

V represents the volume of the solutions for a given realization of the pattern-label pairs. As the number of patterns P increases, V diminishes. The capacity $C_p = \alpha_p N$ is defined as the number of patterns at which $\langle \log V \rangle_{\xi, \chi} \rightarrow -\infty$, where $\langle \cdot \rangle_{\xi, \chi}$ denotes the average over all possible realizations of input states ξ_j^μ and output labels χ^μ . Above this limit, for a set of realizations with nonzero measure, no $\{J_j\}$ configuration can satisfy all the constraints (A1) (and vice versa, below this limit, the perceptron can store all the patterns with probability 1).

In practice, the calculation of $\langle \log V \rangle_{\xi, \chi}$ is done using the replica method: n replicas of the perceptron are introduced, and the typical volume of the solutions for the n replicas $\langle V^n \rangle_{\xi, \chi}$ is obtained using the saddle-point method for $N \rightarrow \infty$, and in the limit $n \rightarrow 0$ (assuming analytic continuation of the volume from positive integers). Finally, the capacity α_p is determined by solving for $\langle V^{n \rightarrow 0} \rangle_{\xi, \chi} \rightarrow 0$, so that $\langle \log V \rangle_{\xi, \chi} = \lim_{n \rightarrow 0} \frac{\langle V^n \rangle_{\xi, \chi} - 1}{n} \rightarrow -\infty$. All the steps are recapitulated in detail below.

2. Calculation of the volume of the solutions for n replicas

The volume of the solutions for the n replicas (indexed by the superscript α) is defined as

$$\langle V^n \rangle_{\xi, \chi} = \frac{1}{V_0^n} \left\langle \prod_{\alpha} \int \prod_j dJ_j^{\alpha} \delta \left(\sum_j \sigma_j^2 (J_j^{\alpha})^2 - N \right) \prod_{\mu} \Theta \left(\frac{\chi^{\mu}}{\sqrt{N}} \sum_j J_j^{\alpha} \xi_j^{\mu} - \kappa \right) \right\rangle_{\xi, \chi} \quad (\text{A4})$$

with $V_0^n = \prod_{\alpha} \int \prod_j dJ_j^{\alpha} \delta(\sum_j \sigma_j^2 (J_j^{\alpha})^2 - N)$.

We substitute the Heaviside step function Θ with its integral representation:

$$\Theta(y - \kappa) = \int_{\kappa}^{\infty} \frac{d\lambda}{2\pi} \int_{-\infty}^{\infty} dx e^{ix(\lambda - y)}$$

$$\left\langle \prod_{\alpha, \mu} \Theta \left(\frac{\chi^{\mu}}{\sqrt{N}} \sum_j J_j^{\alpha} \xi_j^{\mu} - \kappa \right) \right\rangle_{\xi, \chi} = \int_{-\infty}^{\infty} \left(\prod_{\alpha, \mu} dx_{\mu}^{\alpha} \right) \int_{\kappa}^{\infty} \left(\prod_{\alpha, \mu} \frac{d\lambda_{\mu}^{\alpha}}{2\pi} \right) e^{\sum_{\alpha, \mu} ix_{\mu}^{\alpha} \lambda_{\mu}^{\alpha}} \left\langle e^{-\frac{i}{\sqrt{N}} \sum_{j, \mu} \xi_j^{\mu} \chi^{\mu} \sum_{\alpha} x_{\mu}^{\alpha} J_j^{\alpha}} \right\rangle_{\xi, \chi}. \quad (\text{A5})$$

By taking the average over the distributions of $\{\xi_j^{\mu}\}$, we obtain

$$\left\langle e^{-\frac{i}{\sqrt{N}} \sum_{j, \mu} \xi_j^{\mu} \chi^{\mu} \sum_{\alpha} x_{\mu}^{\alpha} J_j^{\alpha}} \right\rangle_{\xi, \chi} = \left\langle \exp \left[\sum_{\mu, j} \ln \varphi_{\xi_j^{\mu}} \left(-\frac{\chi^{\mu}}{\sqrt{N}} \sum_{\alpha} x_{\mu}^{\alpha} J_j^{\alpha} \right) \right] \right\rangle_{\chi}, \quad (\text{A6})$$

where $\varphi_{\xi_j^{\mu}}(\cdot)$ is the characteristic function of ξ_j^{μ} .

Expanding $\varphi_{\xi_j^{\mu}}(-\frac{\chi^{\mu}}{\sqrt{N}} \sum_{\alpha} x_{\mu}^{\alpha} J_j^{\alpha})$ and its logarithm to second order around zero and using the constraint (A2), we obtain

$$\begin{aligned} \left\langle e^{-\frac{i}{\sqrt{N}} \sum_{j, \mu} \xi_j^{\mu} \chi^{\mu} \sum_{\alpha} x_{\mu}^{\alpha} J_j^{\alpha}} \right\rangle_{\xi, \chi} &\simeq \left\langle \exp \left[\sum_{\mu, j} \left(-i m_j \frac{\chi^{\mu}}{\sqrt{N}} \left(\sum_{\alpha} x_{\mu}^{\alpha} J_j^{\alpha} \right) - \frac{\sigma_j^2}{2N} \left(\sum_{\alpha} x_{\mu}^{\alpha} J_j^{\alpha} \right)^2 \right) \right] \right\rangle_{\chi} \\ &= \left\langle \exp \left(-i \sum_{\mu, \alpha} x_{\mu}^{\alpha} \chi^{\mu} M_{\alpha} - \frac{1}{2} \left(\sum_{\mu, \alpha} (x_{\mu}^{\alpha})^2 + 2 \sum_{\mu, \alpha < \beta} x_{\mu}^{\alpha} x_{\mu}^{\beta} q_{\alpha\beta} \right) \right) \right\rangle_{\chi}, \end{aligned} \quad (\text{A7})$$

where we have introduced the variables

$$M_{\alpha} = \frac{\sum_j m_j J_j^{\alpha}}{\sqrt{N}} \quad (\text{A8})$$

and

$$q_{\alpha\beta} = \frac{\sum_j \sigma_j^2 J_j^{\alpha} J_j^{\beta}}{N}, \quad \alpha \neq \beta. \quad (\text{A9})$$

Substituting (A7) back into (A5):

$$\begin{aligned} \left\langle \prod_{\alpha, \mu} \Theta \left(\frac{\chi^{\mu}}{\sqrt{N}} \sum_j J_j^{\alpha} \xi_j^{\mu} - \kappa \right) \right\rangle_{\xi, \chi} &\simeq \left[\int_{-\infty}^{\infty} \left(\prod_{\alpha} dx^{\alpha} \right) \int_{\kappa}^{\infty} \left(\prod_{\alpha} \frac{d\lambda^{\alpha}}{2\pi} \right) \left\langle \exp \left(i \sum_{\alpha} x^{\alpha} (\lambda^{\alpha} - \chi M_{\alpha}) \right. \right. \right. \\ &\quad \left. \left. \left. - \frac{1}{2} \sum_{\alpha} (x^{\alpha})^2 - \sum_{\alpha < \beta} x^{\alpha} x^{\beta} q_{\alpha\beta} \right) \right\rangle_{\chi} \right]^P := (I_1)^P. \end{aligned} \quad (\text{A10})$$

For convenience, we have introduced the symbol I_1 to denote the integral. Using (A8) and (A9), we can rewrite $\langle V^n \rangle_{\xi, \chi}$ with M_α and $q_{\alpha\beta}$ as integral variables:

$$\begin{aligned} \langle V^n \rangle_{\xi, \chi} &\simeq \frac{1}{V_0^n} \int \left(\prod_{j, \alpha} dJ_j^\alpha \right) \left(\prod_{\alpha < \beta} dq_{\alpha\beta} \delta \left(\sum_j \sigma_j^2 J_j^\alpha J_j^\beta - N q_{\alpha\beta} \right) \right) \\ &\times \left(\prod_{\alpha} dM_\alpha \delta \left(\sum_j m_j J_j^\alpha - \sqrt{N} M_\alpha \right) \delta \left(\sum_j \sigma_j^2 (J_j^\alpha)^2 - N \right) \right) (I_1)^P. \end{aligned} \quad (\text{A11})$$

We now use the integral representation of the δ function

$$\delta \left(\sum_j \sigma_j^2 (J_j^\alpha)^2 - N \right) = \int_{-\infty}^{\infty} \frac{dE_\alpha}{2\pi} \exp \left(i E_\alpha \left(\sum_j \sigma_j^2 (J_j^\alpha)^2 - N \right) \right).$$

We do the same for the other δ functions in Eq. (A11), introducing two additional variables $F_{\alpha\beta}$ and K_α :

$$\begin{aligned} \langle V^n \rangle_{\xi, \chi} &\simeq \frac{1}{V_0^n} \int \left(\prod_{j, \alpha} dJ_j^\alpha \right) \left(\prod_{\alpha < \beta} dq_{\alpha\beta} \frac{dF_{\alpha\beta}}{2\pi} \right) \left(\prod_{\alpha} dM_\alpha \frac{dE_\alpha}{2\pi} \frac{dK_\alpha}{2\pi} \right) (I_1)^P \\ &\times \exp \left(-iN \sum_{\alpha} E_\alpha + i \sum_{\alpha} E_\alpha \sum_j \sigma_j^2 (J_j^\alpha)^2 - iN \sum_{\alpha < \beta} F_{\alpha\beta} q_{\alpha\beta} \right. \\ &\quad \left. + i \sum_{\alpha < \beta} F_{\alpha\beta} \sum_j \sigma_j^2 J_j^\alpha J_j^\beta - i\sqrt{N} \sum_{\alpha} K_\alpha M_\alpha + i \sum_{\alpha} K_\alpha \sum_j m_j J_j^\alpha \right). \end{aligned} \quad (\text{A12})$$

Equation (A12) can be expressed as

$$\langle V^n \rangle_{\xi, \chi} \simeq \frac{1}{V_0^n} \int \left(\prod_{\alpha < \beta} dq_{\alpha\beta} \frac{dF_{\alpha\beta}}{2\pi} \right) \left(\prod_{\alpha} dM_\alpha \frac{dE_\alpha}{2\pi} \frac{dK_\alpha}{2\pi} \right) e^{NG}, \quad (\text{A13})$$

where

$$G = \alpha' G_1(q_{\alpha\beta}, M_\alpha) + G_2(F_{\alpha\beta}, E_\alpha, K_\alpha) - i \sum_{\alpha} E_\alpha - i \sum_{\alpha < \beta} F_{\alpha\beta} q_{\alpha\beta}, \quad (\text{A14})$$

$$\alpha' = \frac{P}{N}, \quad (\text{A15})$$

$$G_1 = \ln I_1, \quad (\text{A16})$$

$$G_2(F_{\alpha\beta}, E_\alpha, K_\alpha) = \frac{1}{N} \sum_{j=1}^N \ln \int \left(\prod_{\alpha} dJ_j^\alpha \right) \exp \left(i \sum_{\alpha} E_\alpha \sigma_j^2 (J_j^\alpha)^2 + i \sum_{\alpha < \beta} F_{\alpha\beta} \sigma_j^2 J_j^\alpha J_j^\beta + i \sum_{\alpha} K_\alpha m_j J_j^\alpha \right). \quad (\text{A17})$$

As $N \rightarrow \infty$ and $P \rightarrow \infty$ with $\alpha' = P/N$ finite, $i\sqrt{N} \sum_{\alpha} K_\alpha M_\alpha$ in Eq. (A12) is small compared to the other terms and is discarded.

Below we estimate G_1 , G_2 , and the volume, Eq. (A13), using the saddle-point method as $N \rightarrow \infty$ (α' finite). To obtain the saddle-point equations, we take the limit $n \rightarrow 0$ (assuming an analytic continuation of $\langle V^n \rangle_{\xi, \chi}$ from positive integers) and we use the replica-symmetric ansatz [18],

$$M_\alpha = M, \quad q_{\alpha\beta} = q, \quad E_\alpha = E, \quad K_\alpha = K, \quad F_{\alpha\beta} = F. \quad (\text{A18})$$

3. Calculation of G_1

Using the replica-symmetric ansatz, I_1 simplifies to

$$I_1 = \int_{-\infty}^{\infty} \left(\prod_{\alpha} dx^\alpha \right) \int_{\kappa}^{\infty} \left(\prod_{\alpha} \frac{d\lambda^\alpha}{2\pi} \right) \left\langle \exp \left(i \sum_{\alpha} x^\alpha (\lambda^\alpha - \chi M) - \frac{1}{2} (1-q) \sum_{\alpha} (x^\alpha)^2 - \frac{1}{2} q \left(\sum_{\alpha} x^\alpha \right)^2 \right) \right\rangle_{\chi}. \quad (\text{A19})$$

The term containing $(\sum_{\alpha} x^{\alpha})^2$ is replaced by a Gaussian integral, so that I_1 factorizes over the replicas α . Defining $Dy = dy(e^{-y^2/2})/\sqrt{2\pi}$, we have

$$\begin{aligned} & \left\langle \exp \left(i \sum_{\alpha} x^{\alpha} (\lambda^{\alpha} - \chi M) - \frac{1}{2} (1-q) \sum_{\alpha} (x^{\alpha})^2 - \frac{1}{2} q \left(\sum_{\alpha} x^{\alpha} \right)^2 \right) \right\rangle_{\chi} \\ &= \left\langle \int_{-\infty}^{\infty} Dy \exp \left(i \sum_{\alpha} x^{\alpha} (\lambda^{\alpha} - \chi M) - \frac{1}{2} (1-q) \sum_{\alpha} (x^{\alpha})^2 + iy\sqrt{q} \sum_{\alpha} x^{\alpha} \right) \right\rangle_{\chi}. \end{aligned} \quad (\text{A20})$$

Thus

$$I_1 = \int_{-\infty}^{\infty} Dy \left\langle \left[\int_{\kappa}^{\infty} \frac{d\lambda}{2\pi} \int_{-\infty}^{\infty} dx \exp \left(-\frac{1}{2} (1-q)x^2 + ix(\lambda - \chi M + y\sqrt{q}) \right) \right]^n \right\rangle_{\chi}. \quad (\text{A21})$$

We introduce the following notation for convenience:

$$L(q, M, \chi, y) = \int_{\kappa}^{\infty} \frac{d\lambda}{2\pi} \int_{-\infty}^{\infty} dx \exp \left(-\frac{1}{2} (1-q)x^2 + ix(\lambda - \chi M + y\sqrt{q}) \right) \quad (\text{A22})$$

so that

$$I_1 = \int_{-\infty}^{\infty} Dy \langle L^n(q, M, \chi, y) \rangle_{\chi}. \quad (\text{A23})$$

Solving the Gaussian integral in x in Eq. (A22), we obtain

$$L(q, M, \chi, y) = \frac{1}{\sqrt{2\pi(1-q)}} \int_{\kappa}^{\infty} d\lambda \exp \left(-\frac{(\lambda - \chi M + y\sqrt{q})^2}{2(1-q)} \right) = \frac{1}{2} \text{erfc}(x_1) \quad (\text{A24})$$

with $\text{erfc}(x_1) = \frac{2}{\sqrt{\pi}} \int_{x_1}^{\infty} e^{-t^2} dt$ and $x_1 = \frac{\kappa - \chi M + y\sqrt{q}}{\sqrt{2(1-q)}}$.

We now take the limit of G_1 as $n \rightarrow 0$:

$$G_1(q, M) = \ln \int_{-\infty}^{\infty} Dy \langle L^n(q, M, \chi, y) \rangle_{\chi} \xrightarrow{n \rightarrow 0} n \left\langle \int_{-\infty}^{\infty} Dy \ln L(q, M, \chi, y) \right\rangle_{\chi}, \quad (\text{A25})$$

where we have used $L^n = e^{n \ln L} \xrightarrow{n \rightarrow 0} 1 + n \ln L$.

Finally, we write the average over the χ values explicitly:

$$G_1(q, M) \simeq n \left(\frac{1 + m^{\text{out}}}{2} \int_{-\infty}^{\infty} Dy \ln L(\chi = 1) + \frac{1 - m^{\text{out}}}{2} \int_{-\infty}^{\infty} Dy \ln L(\chi = -1) \right), \quad (\text{A26})$$

where $L(\chi = \pm 1)$ is the simplified notation for $L(q, M, \chi = \pm 1, y)$.

4. Calculation of G_2

Using the replica-symmetric ansatz, Eqs. (A18), G_2 simplifies to

$$G_2(F, E, K) = \frac{1}{N} \sum_{j=1}^N \ln \int \left(\prod_{\alpha} dJ_j^{\alpha} \right) \exp \left(i \sum_{\alpha} E \sigma_j^2 (J_j^{\alpha})^2 + i \sum_{\alpha < \beta} F \sigma_j^2 J_j^{\alpha} J_j^{\beta} + i \sum_{\alpha} K m_j J_j^{\alpha} \right). \quad (\text{A27})$$

We define two statistics $s_1 = \frac{1}{2N} \sum_j \ln(\sigma_j)^2$ and $s_2 = \frac{1}{N} \sum_j \frac{m_j^2}{\sigma_j^2}$ related to the distributions of input states, and we solve the multidimensional Gaussian integral in Eq. (A27):

$$G_2 = c - \frac{1}{2} \ln \left(E + \frac{n-1}{2} F \right) - \frac{n-1}{2} \ln \left(E - \frac{F}{2} \right) - n s_1 - n s_2 \frac{i K^2}{4E + 2(n-1)F}, \quad (\text{A28})$$

where c is a constant, independent of the relevant variables.

We take the limit $n \rightarrow 0$:

$$G_2 \xrightarrow{n \rightarrow 0} c - n \left[\frac{1}{2} \ln \left(E - \frac{F}{2} \right) + \frac{F}{4E - 2F} + s_1 + s_2 \frac{i K^2}{4E - 2F} \right]. \quad (\text{A29})$$

5. Derivation of the saddle-point equations

We substitute the results obtained in the limits $N \rightarrow \infty$, $n \rightarrow 0$ into Eq. (A14):

$$G/n = \alpha' \left\langle \int_{-\infty}^{\infty} Dy \ln L(q, M, \chi, y) \right\rangle_{\chi} - \frac{1}{2} \ln \left(E - \frac{F}{2} \right) - \frac{F}{4E - 2F} - s_1 - s_2 \frac{i K^2}{4E - 2F} - iE + \frac{i}{2} Fq + c. \quad (\text{A30})$$

By the method of steepest descent, the volume is estimated at the saddle-point:

$$\begin{aligned} \frac{\partial(G/n)}{\partial K} &= 0, & \frac{\partial(G/n)}{\partial F} &= 0, & \frac{\partial(G/n)}{\partial E} &= 0, \\ \frac{\partial(G/n)}{\partial q} &= 0, & \frac{\partial(G/n)}{\partial M} &= 0. \end{aligned} \quad (\text{A31})$$

The first three equations in (A31) give K, F, E as functions of q :

$$K = 0, \quad F = -\frac{i q}{(1 - q)^2}, \quad E = \frac{i(1 - 2q)}{2(1 - q)^2}.$$

Substituting them back into (A30), we have

$$G/n = \alpha' \left\langle \int_{-\infty}^{\infty} Dy \ln L(q, M, \chi, y) \right\rangle_{\chi} + \frac{1}{2} \ln(1 - q) + \frac{1}{2(1 - q)} - s_1 + c. \quad (\text{A32})$$

As α' increases, the typical volume $\langle V^{n \rightarrow 0} \rangle_{\xi, \chi}$ decreases and the solutions $\{J_j^\alpha\}$ become more correlated. Approaching the maximal capacity ($\alpha' \rightarrow \alpha_p$), the typical volume shrinks to 0 and the solutions $\{J_j^\alpha\}$ for all the replicas converge to the same configuration. It follows that $q = q_{\alpha\beta} = \frac{\sum_j \sigma_j^\alpha J_j^\alpha J_j^\beta}{N} \rightarrow 1$ for all α, β , by constraint (A2).

Thus, to estimate the maximal capacity α_p , it is sufficient to solve the two remaining saddle-point equations in (A31) for $q \rightarrow 1$:

$$0 = \frac{\partial G(\alpha' \rightarrow \alpha_p)/n}{\partial q} \Big|_{q \rightarrow 1} \quad (\text{A33})$$

and

$$0 = \frac{\partial G(\alpha' \rightarrow \alpha_p)/n}{\partial M} \Big|_{q \rightarrow 1}. \quad (\text{A34})$$

6. Solving Eqs. (A33) and (A34)

$$\begin{aligned} \frac{\partial G(\alpha' \rightarrow \alpha_p)/n}{\partial q} &= \alpha_p \frac{1 + m^{\text{out}}}{2} \int_{-\infty}^{\infty} Dy \frac{1}{L(\chi = 1)} \frac{\partial L(\chi = 1)}{\partial q} \\ &+ \alpha_p \frac{1 - m^{\text{out}}}{2} \int_{-\infty}^{\infty} Dy \frac{1}{L(\chi = -1)} \\ &\times \frac{\partial L(\chi = -1)}{\partial q} + \frac{q}{2(1 - q)^2}. \end{aligned} \quad (\text{A35})$$

For the first term on the right-hand side of Eq. (A35), substituting the expression for $L(\chi = 1)$ from (A24), we obtain

$$\text{1st term} = -\alpha_p \frac{1 + m^{\text{out}}}{2} \int_{-\infty}^{\infty} Dy \frac{2}{\sqrt{\pi}} \frac{e^{-x_1^2}}{\text{erfc}(x_1)} \frac{\partial x_1}{\partial q} \quad (\text{A36})$$

with $x_1 = \frac{\kappa - M + y\sqrt{q}}{\sqrt{2(1 - q)}}$.

Approximating the complementary error function $\text{erfc}(x_1) \simeq \frac{e^{-x_1^2}}{\sqrt{\pi} x_1}$ for $x_1 \gg 0$ [105], and taking the limit $q \rightarrow 1$, the integration (A36) simplifies to

$$\begin{aligned} \text{1st term} |_{q \rightarrow 1} &\simeq -\alpha_p \frac{1 + m^{\text{out}}}{2} \frac{1}{2(1 - q)^2} \\ &\times \int_{M - \kappa}^{\infty} Dy (\kappa - M + y)^2. \end{aligned} \quad (\text{A37})$$

For the second term on the right-hand side of Eq. (A35), a similar approach yields

$$\begin{aligned} \text{2nd term} |_{q \rightarrow 1} &\simeq -\alpha_p \frac{1 - m^{\text{out}}}{2} \frac{1}{2(1 - q)^2} \\ &\times \int_{-M - \kappa}^{\infty} Dy (\kappa + M + y)^2. \end{aligned} \quad (\text{A38})$$

Substituting into Eq. (A33), we obtain

$$\begin{aligned} \alpha_p &= \left[\frac{1 + m^{\text{out}}}{2} \int_{M - \kappa}^{\infty} Dy (\kappa - M + y)^2 \right. \\ &\left. + \frac{1 - m^{\text{out}}}{2} \int_{-M - \kappa}^{\infty} Dy (\kappa + M + y)^2 \right]^{-1} \end{aligned} \quad (\text{A39})$$

Following the same procedure for Eq. (A34), we obtain the equation that gives the value of M :

$$\begin{aligned} &\frac{1 + m^{\text{out}}}{2} \int_{M - \kappa}^{\infty} Dy (\kappa - M + y) \\ &= \frac{1 - m^{\text{out}}}{2} \int_{-M - \kappa}^{\infty} Dy (\kappa + M + y). \end{aligned} \quad (\text{A40})$$

Equations (A39) and (A40) together provide the analytical solution for the maximal capacity of the perceptron when $m_j \neq 0$ for at least one input unit. When $m_j = 0$ for all input units, $M = 0$ by definition. Thus, Eq. (A34) is not defined and the maximal capacity is given by Eq. (A39) with $M = 0$:

$$\alpha_p = \left[\int_{-\kappa}^{\infty} Dy (\kappa + y)^2 \right]^{-1}. \quad (\text{A41})$$

7. Special case: $\kappa = 0$

When the stability parameter $\kappa = 0$ [such that condition (A1) is sufficient for the input patterns to be stored as fixed points in the network but not sufficient for the basins of attraction to be finite], the formula for the maximal capacity further simplifies. Integrating by parts in Eq. (A39), solving the integrals in Eq. (A40), and substituting the resulting equality into Eq. (A39), we obtain

$$\alpha_p = 2 \left[1 - m^{\text{out}} \int_{-M}^M Dy \right]^{-1}. \quad (\text{A42})$$

If $m_j = 0$ for all input units, then $M = 0$ and $\alpha_p = 2$; otherwise, M is determined by

$$\frac{1 + m^{\text{out}}}{2} \int_M^\infty Dy (y - M) = \frac{1 - m^{\text{out}}}{2} \int_{-M}^\infty Dy (y + M). \quad (\text{A43})$$

We prove below that when $\kappa = 0$, the maximal capacity is independent of the elliptical regularization (A2).

APPENDIX B: INDEPENDENCE OF THE CAPACITY FROM THE ELLIPTICAL REGULARIZATION WHEN $\kappa = 0$

We show that there exists a bijective relation between the sets of solutions $\{J_{ij}\}$ that satisfy (A1) with $\kappa = 0$ under two arbitrary (different) elliptical constraints: constraint A, $\sum_j a_j J_j^2 = N$, and constraint B, $\sum_j b_j J_j^2 = N$.

Given inputs ξ_j^μ and their corresponding labels χ^μ , for every set $\{J_j^A\}$ that satisfies constraint A and the classification conditions

$$\chi^\mu \left(\sum_j J_j^A \xi_j^\mu \right) > 0 \quad \text{for all } \mu, \quad (\text{B1})$$

there exists a set $\{J_j^B\}$ defined as

$$\{J_j^B\} = \{c J_j^A\} \quad \text{with } c = \sqrt{\frac{N}{\sum_j b_j (J_j^A)^2}} \quad (\text{B2})$$

that satisfies constraint B, $\sum_j b_j (J_j^B)^2 = N$, and the classification conditions:

$$\begin{aligned} \chi^\mu \left(\sum_j J_j^B \xi_j^\mu \right) &= \chi^\mu \left(\sum_j c J_j^A \xi_j^\mu \right) \\ &= c \chi^\mu \left(\sum_j J_j^A \xi_j^\mu \right) > 0. \end{aligned} \quad (\text{B3})$$

Similarly, for every set $\{J_j^B\}$ that is a solution under constraint B, there exists a set $\{J_j^A = \frac{1}{c} J_j^B\}$ that is a solution under constraint A. In conclusion, there exists a one-to-one mapping between the solution spaces under constraints A and B. Since A and B can be chosen arbitrarily, this bijection implies that the maximal capacity, for $\kappa = 0$, is not affected by the specific elliptical constraint used in the calculation. This conclusion is directly related to a result previously presented in [35], where the authors demonstrated that the perceptron capacity is independent of the input covariance in the limit of $\kappa \rightarrow 0$.

APPENDIX C: COMPARISON WITH GARDNER'S SOLUTION

In Gardner's calculation [18,19], all units have binary, independent, identically distributed states, with $m^{\text{out}} = m_j = m$ and $\sigma_j^2 = 1 - m^2$. This condition corresponds to a special case within our framework.

To address the case of arbitrary $\{m_j\}$ and $\{\sigma_j^2\}$, we implemented some modifications in our calculation. These modifications make use of an elliptical constraint on the weights:

$$\sum_{j=1}^N \sigma_j^2 J_j^2 = N. \quad (\text{C1})$$

This is in general different from Gardner's spherical constraint $\sum_j J_j^2 = N$.

In the special case of binary, independent, identically distributed states with mean m and variance $1 - m^2$, our solution for the maximal capacity reduces to Gardner's solution upon a rescaling of the stability parameter κ . In particular, the elliptical constraint simplifies to a spherical form, $\sum_j J_j^2 = \frac{1}{1-m^2} N = c N$, which differs from Gardner's constraint $\sum_j J_j^2 = N$ by the coefficient c . As previously shown in [29], the effect of the coefficient $c = 1/(1 - m^2)$ is equivalent to a reduction in the stability κ of the patterns by a factor $1/\sqrt{c}$. Thus, expressing the capacity α_p under the spherical constraint $\sum_j J_j^2 = c N$ as $\alpha_p(\kappa, m, c)$, the following equivalence holds [29]:

$$\alpha_p(\kappa, m, c) = \alpha_p((1/\sqrt{c})\kappa, m, 1). \quad (\text{C2})$$

Indeed, substituting $\kappa' = (1/\sqrt{c})\kappa$ into Eqs. (A39) and (A40), and defining $M' = \frac{1/\sqrt{c}}{m} M$, we retrieve Gardner's solution for the maximal capacity:

$$\begin{aligned} \alpha_p(\kappa, m, c) &= \left[\frac{1+m}{2} \int_{\frac{mM'-\kappa'}{\sqrt{1-m^2}}}^\infty Dy \left(\frac{\kappa' - mM'}{\sqrt{1-m^2}} + y \right)^2 \right. \\ &\quad \left. + \frac{1-m}{2} \int_{-\frac{mM'+\kappa'}{\sqrt{1-m^2}}}^\infty Dy \left(\frac{\kappa' + mM'}{\sqrt{1-m^2}} + y \right)^2 \right]^{-1} \\ &\equiv \alpha_{\text{Gardner}} = \alpha_p(\kappa', m, 1), \end{aligned} \quad (\text{C3})$$

where M' is given by

$$\begin{aligned} &\frac{1+m}{2} \int_{\frac{mM'-\kappa'}{\sqrt{1-m^2}}}^\infty Dy \left(\frac{\kappa' - mM'}{\sqrt{1-m^2}} + y \right) \\ &= \frac{1-m}{2} \int_{-\frac{mM'+\kappa'}{\sqrt{1-m^2}}}^\infty Dy \left(\frac{\kappa' + mM'}{\sqrt{1-m^2}} + y \right). \end{aligned} \quad (\text{C4})$$

- [1] J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, *Proc. Natl. Acad. Sci. USA* **79**, 2554 (1982).
- [2] W. A. Little, The existence of persistent states in the brain, *Math. Biosci.* **19**, 101 (1974).

- [3] S. I. Amari, Learning patterns and pattern sequences by self-organizing nets of threshold elements, *IEEE Trans. Comput.* **C-21**, 1197 (1972).
- [4] S. G. Brush, History of the Lenz-Ising model, *Rev. Mod. Phys.* **39**, 883 (1967).

- [5] D. J. Amit, *Modeling Brain Function: The World of Attractor Neural Networks* (Cambridge University Press, Cambridge, 1989).
- [6] C. Marullo and E. Agliari, Boltzmann machines as generalized Hopfield networks: A review of recent results and outlooks, *Entropy* **23**, 34 (2020).
- [7] N. Brunel, Is cortical connectivity optimized for storing information?, *Nat. Neurosci.* **19**, 749 (2016).
- [8] D. Krotov and J. J. Hopfield, Large associative memory problem in neurobiology and machine learning, [arXiv:2008.06996](https://arxiv.org/abs/2008.06996).
- [9] D. Krotov, Hierarchical associative memory, [arXiv:2107.06446](https://arxiv.org/abs/2107.06446).
- [10] D. Krotov and J. J. Hopfield, Dense associative memory for pattern recognition, in *Advances in Neural Information Processing Systems*, edited by D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (Curran Associates, Red Hook, NY, 2016), Vol. 29.
- [11] M. Gabri , S. Ganguli, C. Lucibello, and R. Zecchina, Neural networks: From the perceptron to deep nets, in *Spin Glass Theory and Far Beyond: Replica Symmetry Breaking After 40 Years* (World Scientific, Singapore, 2023), pp. 477–497.
- [12] S. Cocco, R. Monasson, L. Posani, S. Rosay, and J. Tubiana, Statistical physics and representations in real and artificial neural networks, *Physica A* **504**, 45 (2018).
- [13] S. Fusi, Memory capacity of neural network models, [arXiv:2108.07839](https://arxiv.org/abs/2108.07839).
- [14] M. K. Benna and S. Fusi, Place cells may simply be memory cells: Memory compression leads to spatial tuning and history dependence, *Proc. Natl. Acad. Sci. USA* **118**, e2018422118 (2021).
- [15] D. O. Hebb, *The Organization of Behavior: A Neuropsychological Theory* (Psychology Press, London, UK, 2005).
- [16] 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).
- [17] D. J. Amit, H. Gutfreund, and H. Sompolinsky, Statistical mechanics of neural networks near saturation, *Ann. Phys.* **173**, 30 (1987).
- [18] E. Gardner, The space of interactions in neural network models, *J. Phys. A* **21**, 257 (1988).
- [19] E. Gardner, Maximum storage capacity in neural networks, *Europhys. Lett.* **4**, 481 (1987).
- [20] E. Gardner and B. Derrida, Optimal storage properties of neural network models, *J. Phys. A* **21**, 271 (1988).
- [21] H. Sompolinsky, Neural networks with nonlinear synapses and a static noise, *Phys. Rev. A* **34**, 2571 (1986).
- [22] B. Wemmenhove and A. C. C. Coolen, Finite connectivity attractor neural networks, *J. Phys. A* **36**, 9617 (2003).
- [23] D. J. Amit, M. R. Evans, H. Horner, and K. Y. M. Wong, Retrieval phase diagrams for attractor neural networks with optimal interactions, *J. Phys. A* **23**, 3361 (1990).
- [24] Y. Feng and N. Brunel, Storage capacity of networks with discrete synapses and sparsely encoded memories, *Phys. Rev. E* **105**, 054408 (2022).
- [25] P. Sollich, D. Tantari, A. Annibale, and A. Barra, Extensive parallel processing on scale-free networks, *Phys. Rev. Lett.* **113**, 238106 (2014).
- [26] I. P. Castillo, B. Wemmenhove, J. P. L. Hatchett, A. C. C. Coolen, N. S. Skantzos, and T. Nikolettopoulos, Analytic solution of attractor neural networks on scale-free graphs, *J. Phys. A* **37**, 8789 (2004).
- [27] T. M. Cover, Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition, *IEEE Trans. Electron. Comput.* **EC-14**, 326 (1965).
- [28] P. Del Giudice, S. Franz, and M. A. Virasoro, Perceptron beyond the limit of capacity, *J. Phys.* **50**, 121 (1989).
- [29] R. Monasson, Properties of neural networks storing spatially correlated patterns, *J. Phys. A* **25**, 3701 (1992).
- [30] N. Brunel, J.-P. Nadal, and G. Toulouse, Information capacity of a perceptron, *J. Phys. A* **25**, 5017 (1992).
- [31] S. Y. Chung, D. D. Lee, and H. Sompolinsky, Linear readout of object manifolds, *Phys. Rev. E* **93**, 060301(R) (2016).
- [32] S. Y. Chung, D. D. Lee, and H. Sompolinsky, Classification and geometry of general perceptual manifolds, *Phys. Rev. X* **8**, 031003 (2018).
- [33] U. Cohen, S. Y. Chung, D. D. Lee, and H. Sompolinsky, Separability and geometry of object manifolds in deep neural networks, *Nat. Commun.* **11**, 746 (2020).
- [34] T. Shinzato and Y. Kabashima, Perceptron capacity revisited: classification ability for correlated patterns, *J. Phys. A* **41**, 324013 (2008).
- [35] F. Gerace, F. Krzakala, B. Loureiro, L. Stephan, and L. Zdeborov , Gaussian universality of perceptrons with random labels, *Phys. Rev. E* **109**, 034305 (2024).
- [36] M. Negri, C. Lauditi, G. Perugini, C. Lucibello, and E. Malatesta, Storage and learning phase transitions in the random-features Hopfield model, *Phys. Rev. Lett.* **131**, 257301 (2023).
- [37] S. Kalaj, C. Lauditi, G. Perugini, C. Lucibello, E. M. Malatesta, and M. Negri, Random features Hopfield networks generalize retrieval to previously unseen examples, [arXiv:2407.05658](https://arxiv.org/abs/2407.05658).
- [38] A. Battista and R. Monasson, Capacity-resolution trade-off in the optimal learning of multiple low-dimensional manifolds by attractor neural networks, *Phys. Rev. Lett.* **124**, 048302 (2020).
- [39] B. Lopez, M. Schroder, and M. Opper, Storage of correlated patterns in a perceptron, *J. Phys. A* **28**, L447 (1995).
- [40] M. Schr der, W. Kinzel, and I. Kanter, Training a perceptron by a bit sequence: Storage capacity, *J. Phys. A* **29**, 7965 (1996).
- [41] L. Kang and T. Toyozumi, Hopfield-like network with complementary encodings of memories, *Phys. Rev. E* **108**, 054410 (2023).
- [42] L. Kang and T. Toyozumi, Distinguishing examples while building concepts in hippocampal and artificial networks, *Nat. Commun.* **15**, 647 (2024).
- [43] E. Barkai, D. Hansel, and H. Sompolinsky, Broken symmetries in multilayered perceptrons, *Phys. Rev. A* **45**, 4146 (1992).
- [44] A. Engel, H. M. K hler, F. Tschepke, H. Vollmayr, and A. Zippelius, Storage capacity and learning algorithms for two-layer neural networks, *Phys. Rev. A* **45**, 7590 (1992).
- [45] B. L. Annesi, E. M. Malatesta, and F. Zamponi, Exact full-RSB SAT/UNSAT transition in infinitely wide two-layer neural networks, *SciPost Phys.* **18**, 118 (2025).
- [46] C. Baldassi, E. M. Malatesta, and R. Zecchina, Properties of the geometry of solutions and capacity of multilayer neural networks with rectified linear unit activations, *Phys. Rev. Lett.* **123**, 170602 (2019).

- [47] J. A. Zavatone-Veth and C. Pehlevan, Activation function dependence of the storage capacity of treelike neural networks, *Phys. Rev. E* **103**, L020301 (2021).
- [48] J. A. Zavatone-Veth, *Statistical Mechanics of Bayesian Inference and Learning in Neural Networks* (Harvard University, Cambridge, MA, 2024).
- [49] M. Stojnic, Fixed width treelike neural networks capacity analysis—generic activations, [arXiv:2402.05696](https://arxiv.org/abs/2402.05696).
- [50] B. Kosko, Bidirectional associative memories, *IEEE Trans. Syst. Man. Cybern.* **18**, 49 (1988).
- [51] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, A learning algorithm for Boltzmann machines, *Cogn. Sci.* **9**, 147 (1985).
- [52] G. E. Hinton, A practical guide to training restricted Boltzmann machines, in *Neural Networks: Tricks of the Trade*, 2nd ed. (Springer, Berlin, 2012), pp. 599–619.
- [53] T. D. Goode, K. Z. Tanaka, A. Sahay, and T. J. McHugh, An integrated index: Engrams, place cells, and hippocampal memory, *Neuron* **107**, 805 (2020).
- [54] K. Nakazawa, M. C. Quirk, R. A. Chitwood, M. Watanabe, M. F. Yeckel, L. D. Sun, A. Kato, C. A. Carr, D. Johnston, M. A. Wilson *et al.*, Requirement for hippocampal CA3 NMDA receptors in associative memory recall, *Science* **297**, 211 (2002).
- [55] X. Liu, S. Ramirez, P. T. Pang, C. B. Puryear, A. Govindarajan, K. Deisseroth, and S. Tonegawa, Optogenetic stimulation of a hippocampal engram activates fear memory recall, *Nature (London)* **484**, 381 (2012).
- [56] T. J. Teyler and J. W. Rudy, The hippocampal indexing theory and episodic memory: Updating the index, *Hippocampus* **17**, 1158 (2007).
- [57] A. Treves and E. T. Rolls, Computational constraints suggest the need for two distinct input systems to the hippocampal CA3 network, *Hippocampus* **2**, 189 (1992).
- [58] H. Nishimori, *Statistical Physics of Spin Glasses and Information Processing: An Introduction* (Clarendon Press, Oxford, UK, 2001), p. 111.
- [59] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/qb9h-qxkx> for additional figures showing the capacity of networks with finite stability, high coding levels, and dendritic structures; the insensitivity of capacity to neuron out-degrees; the optimal coding levels in two-layer networks of finite size; and the capacity of networks with a reversed neuron ratio between the two layers.
- [60] M. V. Tsodyks and M. V. Feigel'man, The enhanced storage capacity in neural networks with low activity level, *Europhys. Lett.* **6**, 101 (1988).
- [61] M. Mézard, The space of interactions in neural networks: Gardner's computation with the cavity method, *J. Phys. A* **22**, 2181 (1989).
- [62] C. Cortes and V. Vapnik, Support-vector networks, *Mach. Learn.* **20**, 273 (1995).
- [63] M. Sorbaro, Q. Liu, M. Bortone, and S. Sheik, Optimizing the energy consumption of spiking neural networks for neuromorphic applications, *Front. Neurosci.* **14**, 662 (2020).
- [64] Y. Cao, Y. Chen, and D. Khosla, Spiking deep convolutional neural networks for energy-efficient object recognition, *Int. J. Comput. Vision* **113**, 54 (2015).
- [65] V. Mehrpour, T. Meyer, E. P. Simoncelli, and N. C. Rust, Pinpointing the neural signatures of single-exposure visual recognition memory, *Proc. Natl. Acad. Sci. USA* **118**, e2021660118 (2021).
- [66] K. Zhang, E. S. Bromberg-Martin, F. Sogukpinar, K. Kocher, and I. E. Monosov, Surprise and recency in novelty detection in the primate brain, *Curr. Biol.* **32**, 2160 (2022).
- [67] J.-Z. Xiang and M. W. Brown, Differential neuronal encoding of novelty, familiarity and recency in regions of the anterior temporal lobe, *Neuropharmacology* **37**, 657 (1998).
- [68] C. Ranganath and G. Rainer, Neural mechanisms for detecting and remembering novel events, *Nat. Rev. Neurosci.* **4**, 193 (2003).
- [69] S. Sardi, R. Vardi, A. Sheinin, A. Goldental, and I. Kanter, New types of experiments reveal that a neuron functions as multiple independent threshold units, *Sci. Rep.* **7**, 18036 (2017).
- [70] A. Polsky, B. W. Mel, and J. Schiller, Computational subunits in thin dendrites of pyramidal cells, *Nat. Neurosci.* **7**, 621 (2004).
- [71] B. Mel, The clusteron: Toward a simple abstraction for a complex neuron, in *Advances in Neural Information Processing Systems*, edited by J. Moody, S. Hanson, and R. Lippmann (Morgan-Kaufmann, San Francisco, CA, 1991), Vol. 4.
- [72] P. Rajasethupathy, S. Sankaran, J. H. Marshel, C. K. Kim, E. Ferenczi, S. Y. Lee, A. Berndt, C. Ramakrishnan, A. Jaffe, M. Lo, C. Liston, and K. Deisseroth, Projections from neocortex mediate top-down control of memory retrieval, *Nature (London)* **526**, 653 (2015).
- [73] B. J. Wiltgen and K. Z. Tanaka, Systems consolidation and the content of memory, *Neurobiol. Learn. Mem.* **106**, 365 (2013).
- [74] P. A. Lewis and S. J. Durrant, Overlapping memory replay during sleep builds cognitive schemata, *Trends Cogn. Sci.* **15**, 343 (2011).
- [75] T. J. Teyler and P. DiScenna, The hippocampal memory indexing theory, *Behav. Neurosci.* **100**, 147 (1986).
- [76] J. W. Rudy, Context representations, context functions, and the parahippocampal–hippocampal system, *Learn. Mem.* **16**, 573 (2009).
- [77] L. Nadel and M. Moscovitch, Memory consolidation, retrograde amnesia and the hippocampal complex, *Curr. Opin. Neurobiol.* **7**, 217 (1997).
- [78] L. R. Squire and P. Alvarez, Retrograde amnesia and memory consolidation: A neurobiological perspective, *Curr. Opin. Neurobiol.* **5**, 169 (1995).
- [79] J. L. McClelland, B. L. McNaughton, and R. C. O'Reilly, Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory, *Psychol. Rev.* **102**, 419 (1995).
- [80] A. Treves and E. T. Rolls, Computational analysis of the role of the hippocampus in memory, *Hippocampus* **4**, 374 (1994).
- [81] R. P. Kesner, A process analysis of the CA3 subregion of the hippocampus, *Front. Cell. Neurosci.* **7**, 78 (2013).
- [82] S. M. Attili, K. Moradi, D. W. Wheeler, and G. A. Ascoli, Quantification of neuron types in the rodent hippocampal formation by data mining and numerical optimization, *Eur. J. Neurosci.* **55**, 1724 (2022).

- [83] E. T. Rolls, The storage and recall of memories in the hippocampo-cortical system, *Cell Tissue Res.* **373**, 577 (2018).
- [84] H. E. Scharfman, The CA3 “backprojection” to the dentate gyrus, *Prog. Brain Res.* **163**, 627 (2007).
- [85] M. W. Jung and B. L. McNaughton, Spatial selectivity of unit activity in the hippocampal granular layer, *Hippocampus* **3**, 165 (1993).
- [86] J. B. Aimone, W. Deng, and F. H. Gage, Resolving new memories: A critical look at the dentate gyrus, adult neurogenesis, and pattern separation, *Neuron* **70**, 589 (2011).
- [87] K. K. Cowansage, T. Shuman, B. C. Dillingham, A. Chang, P. Golshani, and M. Mayford, Direct reactivation of a coherent neocortical memory of context, *Neuron* **84**, 432 (2014).
- [88] A. Engel and C. Broeck, *Statistical Mechanics of Learning* (Cambridge University Press, Cambridge, 2001).
- [89] F. Schönsberg, Y. Roudi, and A. Treves, Efficiency of local learning rules in threshold-linear associative networks, *Phys. Rev. Lett.* **126**, 018301 (2021).
- [90] G. Tavoni, U. Ferrari, F. P. Battaglia, S. Cocco, and R. Monasson, Functional coupling networks inferred from prefrontal cortex activity show experience-related effective plasticity, *Netw. Neurosci.* **1**, 275 (2017).
- [91] G. Tavoni, S. Cocco, and R. Monasson, Neural assemblies revealed by inferred connectivity-based models of prefrontal cortex recordings, *J. Comput. Neurosci.* **41**, 269 (2016).
- [92] S. Cocco, R. Monasson, L. Posani, and G. Tavoni, Functional networks from inverse modeling of neural population activity, *Curr. Opin. Syst. Biol.* **3**, 103 (2017).
- [93] O. Miry, J. Li, and L. Chen, The quest for the hippocampal memory engram: From theories to experimental evidence, *Front. Behav. Neurosci.* **14**, 632019 (2021).
- [94] T. E. J. Behrens, T. H. Muller, J. C. R. Whittington, S. Mark, A. B. Baram, K. L. Stachenfeld, and Z. Kurth-Nelson, What is a cognitive map? Organizing knowledge for flexible behavior, *Neuron* **100**, 490 (2018).
- [95] R. M. Tavares, A. Mendelsohn, Y. Grossman, C. H. Williams, M. Shapiro, Y. Trope, and D. Schiller, A map for social navigation in the human brain, *Neuron* **87**, 231 (2015).
- [96] D. Sun, N. E. K. Shaik, R. R. Unnithan, and C. French, Hippocampal cognitive and relational map paradigms explored by multisensory encoding recording with wide-field calcium imaging, *iScience* **27**, 108603 (2023).
- [97] M. Diamantaki, M. Frey, P. Berens, P. Preston-Ferrer, and A. Burgalossi, Sparse activity of identified dentate granule cells during spatial exploration, *eLife* **5**, e20252 (2016).
- [98] A. Sahay, D. A. Wilson, and R. Hen, Pattern separation: A common function for new neurons in hippocampus and olfactory bulb, *Neuron* **70**, 582 (2011).
- [99] S. B. McHugh, V. Lopes-dos Santos, G. P. Gava, K. Hartwich, S. K. E. Tam, D. M. Bannerman, and D. Dupret, Adult-born dentate granule cells promote hippocampal population sparsity, *Nat. Neurosci.* **25**, 1481 (2022).
- [100] T. Kitamura, Y. Saitoh, N. Takashima, A. Murayama, Y. Niibori, H. Ageta, M. Sekiguchi, H. Sugiyama, and K. Inokuchi, Adult neurogenesis modulates the hippocampus-dependent period of associative fear memory, *Cell* **139**, 814 (2009).
- [101] J. B. Aimone, J. Wiles, and F. H. Gage, Potential role for adult neurogenesis in the encoding of time in new memories, *Nat. Neurosci.* **9**, 723 (2006).
- [102] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, Scikit-learn: Machine learning in Python, *J. Mach. Learn. Res.* **12**, 2825 (2011).
- [103] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright *et al.*, Scipy 1.0: Fundamental algorithms for scientific computing in Python, *Nat. Meth.* **17**, 261 (2020).
- [104] https://github.com/kalfon/heterogeneous_hopfield_network_capacity.git.
- [105] C. Ren and A. R. MacKenzie, Closed-form approximations to the error and complementary error functions and their applications in atmospheric science, *Atm. Sci. Lett.* **8**, 70 (2007).