RESEARCH   Open Access

CrossMark

Robust Exponential Memory in Hopfield Networks

Christopher J. Hillar[1] · Ngoc M. Tran[2]

Abstract The Hopfield recurrent neural network is a classical auto-associative model
of memory, in which collections of symmetrically coupled McCulloch–Pitts binary
neurons interact to perform emergent computation. Although previous researchers
have explored the potential of this network to solve combinatorial optimization prob-
lems or store reoccurring activity patterns as attractors of its deterministic dynamics,
a basic open problem is to design a family of Hopfield networks with a number of
noise-tolerant memories that grows exponentially with neural population size. Here,
we discover such networks by minimizing probability flow, a recently proposed ob-
jective for estimating parameters in discrete maximum entropy models. By descend-
ing the gradient of the convex probability flow, our networks adapt synaptic weights
to achieve robust exponential storage, even when presented with vanishingly small
numbers of training patterns. In addition to providing a new set of low-density error-
correcting codes that achieve Shannon's noisy channel bound, these networks also
efficiently solve a variant of the hidden clique problem in computer science, opening
new avenues for real-world applications of computational models originating from
biology.

Christopher J. Hillar and Ngoc M. Tran contributed equally to this work.

✉ C.J. Hillar
chillar@msri.org

N.M. Tran
ntran@math.utexas.edu

[1] Redwood Center for Theoretical Neuroscience, Berkeley, CA, USA

[2] University of Texas, Austin, Austin, TX, USA

Springer

Abbreviations
OPR outer-product rule
MPF minimum probability flow

1 Introduction

Discovered first by Pastur and Figotin [1] as a simplified spin glass [2] in statistical physics, the Hopfield model [3] is a recurrent network of n linear threshold McCulloch–Pitts [4] neurons that can store $n/(4 \ln n)$ binary patterns [5] as distributed "memories" in the form of auto-associative fixed-point attractors. While several aspects of these networks appeared earlier (see, e.g., [6] for dynamics and learning), the approach nonetheless introduced ideas from physics into the theoretical study of neural computation. The Hopfield model and its variants have been studied intensely in theoretical neuroscience and statistical physics [7], but investigations into its utility for memory and coding have mainly focused on storing collections of patterns X using a "one-shot" *outer-product rule* (OPR) for learning, which essentially assigns abstract synaptic weights between neurons to be their correlation, an early idea in neuroscience [8, 9]. Independent of learning, at most $2n$ randomly generated dense patterns can be simultaneously stored in networks with n neurons [10].

Despite this restriction, super-linear capacity in Hopfield networks is possible for special pattern classes and connectivity structures. For instance, if patterns to memorize contain many zeros, it is possible to store nearly a quadratic number [11]. Other examples are random networks, which have $\approx 1.22^n$ attractors asymptotically [12], and networks storing all permutations [13]. In both examples of exponential storage, however, memories have vanishingly small basins of attraction, making them ill-suited for noise-tolerant pattern storage. Interestingly, the situation is even worse for networks storing permutations: any Hopfield network storing permutations will not recover the derangements (more than a third of all permutations) from asymptotically vanishing noise (see Theorem 4, proved in Sect. 5).

In this note, we design a family of sparsely connected n-node Hopfield networks with (asymptotically, as $n \to \infty$)

$$\sim \frac{2^{\sqrt{2n}+\frac{1}{4}}}{n^{1/4}\sqrt{\pi}} \tag{1}$$

robustly stored fixed-point attractors by minimizing "probability flow" [14, 15]. To our knowledge, this is the first rigorous demonstration of super-polynomial noise-tolerant storage in recurrent networks of simple linear threshold elements. The approach also provides a normative, convex, biologically plausible learning mechanism for discovering these networks from small amounts of data and reveals new connections between binary McCulloch–Pitts neural networks, efficient error-correcting codes, and computational graph theory.

2 Background

The underlying probabilistic model of data in the Hopfield network is the non-ferromagnetic *Lenz–Ising model* [16] from statistical physics, more generally called

Fig. 1 Energy landscape and discrete dynamics in a Hopfield network having robust storage of all 4-cliques in graphs on 8 vertices. The deterministic network dynamics sends three corrupted cliques to graphs with smaller energy, converging on the underlying 4-clique attractors
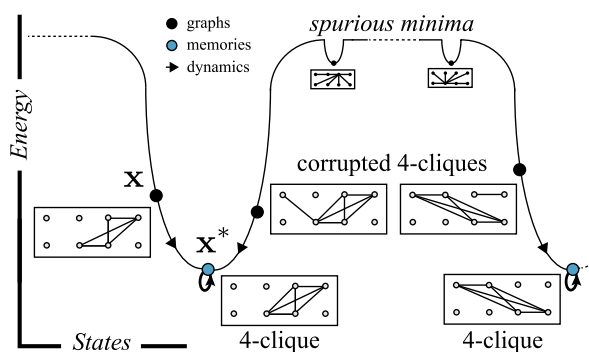
a Markov random field in the literature, and the model distribution in a fully observable Boltzmann machine [17] from artificial intelligence. The states of this discrete distribution are length n binary column vectors $\mathbf{x} = (x_1, \ldots, x_n) \in \{0, 1\}^n$ each having probability $p_{\mathbf{x}} := \frac{1}{Z} \exp(-E_{\mathbf{x}})$, in which $E_{\mathbf{x}} := -\frac{1}{2}\mathbf{x}^\top \mathbf{W}\mathbf{x} + \theta^\top \mathbf{x}$ is the *energy* of a state, $\mathbf{W}$ is an n-by-n real symmetric matrix with zero diagonal (the *weight matrix*), the vector $\theta \in \mathbb{R}^n$ is a *threshold* term, and $Z := \sum_{\mathbf{x}} \exp(-E_{\mathbf{x}})$ is the *partition function*, the normalizing factor ensuring that $p_{\mathbf{x}}$ represents a probability distribution. In theoretical neuroscience, rows $\mathbf{W}_e$ of the matrix $\mathbf{W}$ are interpreted as abstract "synaptic" weights W_{ef} connecting neuron e to other neurons f.

The pair $(\mathbf{W}, \theta)$ determines an asynchronous deterministic ("zero-temperature") *dynamics* on states $\mathbf{x}$ by replacing each x_e in $\mathbf{x}$ with the value:

$$x_e = \begin{cases} 1 & \text{if } \sum_{f \neq e} W_{ef} x_f > \theta_e, \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

in a (usually initialized randomly) fixed order through all neurons $e = 1, \ldots, n$. The quantity $I_e := \langle \mathbf{W}_e, \mathbf{x} \rangle$ in (2) is often called the *feedforward input* to neuron e and may be computed by linearly combining input signals from neurons with connections to e. Let ΔE_e (resp. $\Delta x_e = \pm 1, 0$) be the energy (resp. bit) change when applying (2) at neuron e. The relationship

$$\Delta E_e = -\Delta x_e (I_e - \theta_e) \tag{3}$$

guarantees that network dynamics does not increase energy. Thus, each initial state $\mathbf{x}$ will converge in a finite number of steps to its *attractor* $\mathbf{x}^*$ (also called in the literature *fixed-point*, *memory*, or *metastable state*); e.g., see Fig. 1. The biological plausibility and potential computational power [18] of the dynamics update (2) inspired both early computer [19] and neural network architectures [4, 20].

We next formalize the notion of robust fixed-point attractor storage for families of Hopfield networks. For $p \in [0, \frac{1}{2}]$, the *p-corruption* of $\mathbf{x}$ is the random pattern $\mathbf{x}_p$ obtained by replacing each x_e by $1 - x_e$ with probability p, independently. The *p*-corruption of a state differs from the original by pn bit flips on average so that for larger p it is more difficult to recover the original binary pattern; in particular, $\mathbf{x}_{\frac{1}{2}}$ is the uniform distribution on $\{0, 1\}^n$ (and thus independent of $\mathbf{x}$). Given a Hopfield network, the attractor $\mathbf{x}^*$ has $(1 - \varepsilon)$-*tolerance* for a *p*-corruption if the dynamics can

recover $\mathbf{x}^*$ from $(\mathbf{x}^*)_p$ with probability at least $1 - \varepsilon$. The α-*robustness* $\alpha(X, \varepsilon)$ for a set of states X is the most p-corruption every state $(1 - \varepsilon)$-tolerates.

At last, we say that a sequence of Hopfield networks $\mathcal{H}_n$ *robustly stores* states X_n with robustness index $\alpha > 0$ if the following limit exists and equals the number α:

$$\lim_{\varepsilon \to 0^+} \lim_{n \to \infty} \inf\{\alpha(X_n, \varepsilon), \alpha(X_{n+1}, \varepsilon), \ldots\} = \alpha. \tag{4}$$

If α is the robustness index of a family of networks, then the chance that dynamics does not recover an α-corrupted memory can be made as small as desired by devoting more neurons. (Note that by definition, we always have $\alpha \leq 1/2$.)

To determine parameters $(\mathbf{W}, \theta)$ in our networks from a set of training patterns $X \subseteq \{0, 1\}^n$, we minimize the following *probability flow* objective function [14, 15]:

$$\frac{1}{|X|} \sum_{\mathbf{x} \in X} \sum_{\mathbf{x}' \in \mathcal{N}(\mathbf{x})} \exp\left(\frac{E_{\mathbf{x}} - E_{\mathbf{x}'}}{2}\right), \tag{5}$$

in which $\mathcal{N}(\mathbf{x})$ are those neighboring states $\mathbf{x}'$ differing from $\mathbf{x}$ by a single flipped bit. It is elementary that a Hopfield network has attractors X if and only if the probability flow (5) can be arbitrarily close to zero, motivating the application of minimizing (5) to find such networks [15]. Importantly, the probability flow is a convex function of the parameters, consists of a number of terms linear in n and the size of X, and avoids the exponentially large partition function Z. We remark that the factor of $\frac{1}{2}$ inside of the exponential in (5) will turn out to be unimportant for our analysis; however, we keep it to be consistent with the previous literature on interpreting (5) as a probability density estimation objective.

Let v be a positive integer and set $n = \frac{v(v-1)}{2}$. A state $\mathbf{x}$ in a Hopfield network on n nodes represents a simple undirected graph G on v vertices by interpreting a binary entry x_e in $\mathbf{x}$ as indicating whether edge e is in G ($x_e = 1$) or not ($x_e = 0$). A *k-clique* $\mathbf{x}$ is one of the $\binom{v}{k} = \frac{v \cdot (v-1) \cdots (v-k+1)}{k \cdot (k-1) \cdots 2 \cdot 1}$ graphs consisting of k fully connected nodes and $v - k$ other isolated nodes. Below, in Sect. 3, we will design Hopfield networks that have all k-cliques on $2k$ (or $2k - 2$) vertices as robustly stored memories. For large n, the count $\binom{2k}{k}$ approaches (1) by Stirling's approximation. Figure 1 depicts a network with $n = 28$ neurons storing 4-cliques in graphs on $v = 8$ vertices.

3 Results

Our first result is that numerical minimization of probability flow over a vanishingly small critical number of training cliques determines linear threshold networks with exponential attractor memory. We fit all-to-all connected networks on $n = 3160, 2016, 1128$ neurons ($v = 80, 64, 48$; $k = 40, 32, 24$) with increasing numbers of randomly generated k-cliques as training data X by minimizing (5) with the limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm [21] (implemented in the programming language Python's package SciPy). In Fig. 2, we plot the percentage of 1000 random new k-cliques that are fixed-points in these networks after training as a function of the ratio of training set size to total number of k-cliques.
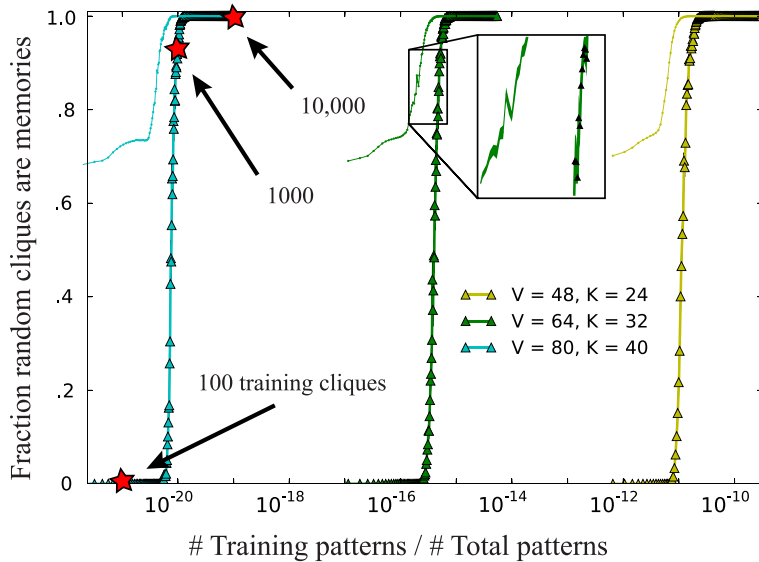
Fig. 2 Learning critical networks with exponential memory by minimizing probability flow on few training patterns. For numbers of vertices $v = 80, 64, 48$ ($k = 40, 32, 24$) with 50 trials each, the average percent of 1000 randomly drawn cliques that are memories vs. the fraction of training samples to total number of k-cliques. Inset displays enlarged version of the region demarcated by *black square*; *filled regions* indicate standard deviation errors over these 50 trials. *Dotted lines* are average percentage of correct bits after converging dynamics
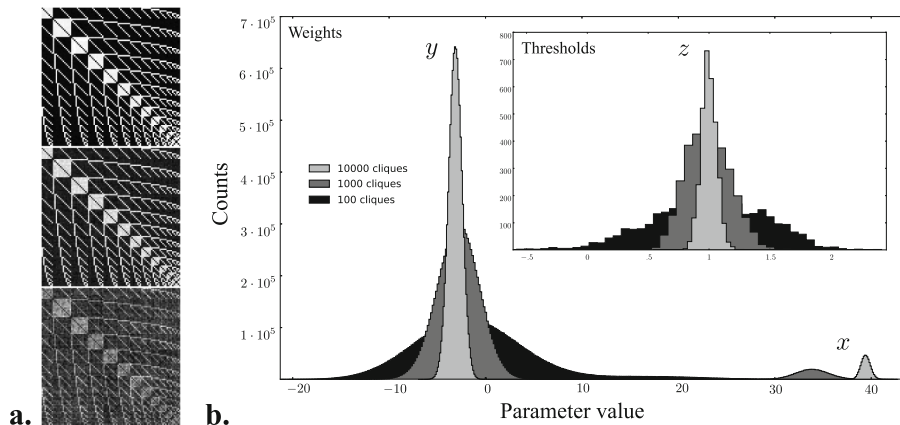

Fig. 3 Distribution of network parameters learned by minimizing probability flow (MPF) sharpens around three critical values. **(a)** Portion of network weights **W** after minimizing (5) given 100 (*bottom*), 1000 (*middle*), or 10,000 (*top*) random 40-cliques X (of about 10^{23} in total) on $v = 80$ vertices. These networks represent the marked points in Fig. 2. **(b)** Histograms of weight and threshold parameters for networks in (a) (histogram of thresholds θ in inset). Network parameters are scaled so that thresholds have mean 1 (this does not affect the dynamics). Groups of similar network weights and thresholds are labeled with corresponding parameter x, y, z

Each triangle in the figure represents the average of this fraction over 50 networks, each given the same number of randomly generated (but different) training data. The finding is that a critical number of training samples allows for storage of all k-cliques. Moreover, this count is significantly smaller than the total number of patterns to be learned.

In Fig. 3(a), we display a portion of the weight matrix with minimum probability flow representing a $v = 80$ network (4,994,380 weight and threshold parameters) given 100 ($\approx$1e−21% of all 40-cliques), 1000 (1e−20%), or 10,000 (1e−19%) randomly generated 40-cliques as training data; these are the three special starred points in Fig. 2. In Fig. 3(b), we also plot histograms of learned parameters from networks trained on data with these three sample sizes. The finding is that weights and thresholds become highly peaked and symmetric about three limiting quantities as sample size increases.

We next analytically minimize probability flow to determine explicit networks achieving robust exponential storage. To simplify matters, we first observe by a symmetrizing argument (see Sect. 5) that there is a network storing all k-cliques if and only if there is one with constant threshold $\theta = (z, \ldots, z) \in \mathbb{R}^n$ and satisfying for each pair $e \neq f$, ether $W_{ef} = x$ (whenever e and f share one vertex) or $W_{ef} = y$ (when e and f are disjoint). Weight matrices approximating this symmetry can be seen in Fig. 3(a). (Note that this symmetry structure on the weights is independent of clique size k.) In this case, the energy of a graph G with $\#E(G)$ edges is the following linear function of $(x, y, z) \in \mathbb{R}^3$:

$$E_G(x, y, z) = -x \cdot S_1(G) - y \cdot S_0(G) + z \cdot \#E(G), \tag{6}$$

in which $S_1(G)$ and $S_0(G)$ are the number of edge pairs in the graph G with exactly one or zero shared vertices, respectively.

Consider the minimization of (5) over a training set X consisting of all $\binom{v}{k}$ k-cliques on $v = 2k - 2$ vertices (this simplifies the mathematics), restricting networks to our 3-parameter family (x, y, z). When $y = 0$, these networks are sparsely connected, having a vanishing number of connections between neurons relative to total population size. Using single variable calculus and Eq. (6), one can check that, for any fixed positive threshold z, the minimum value of (5) is achieved uniquely at the parameter setting $(x, 0, z)$, where

$$x = \frac{2z}{3k - 5}. \tag{7}$$

This elementary calculation gives our first main theoretical contribution.

Theorem 1 *McCulloch–Pitts attractor networks minimizing probability flow can achieve robust exponential pattern storage.*

We prove Theorem 1 using the following large deviation theory argument; this approach also allows us to design networks achieving optimal robustness index $\alpha = 1/2$ (Theorem 2). Fix $v = 2k$ (or $v = 2k - 2$) and consider a p-corrupted clique. Using Bernstein's concentration inequality for sums of Bernoulli binary random variables

[22] ("coin flips"), it can be shown that with high probability (i.e., approaching 1 as $v \to \infty$) an edge in the clique has $2k$ neighboring edges at least, on average (see Corollary 1).

This gives the fixed-point requirement from (2):

$$2kx + o(x\sqrt{k}\ln k) > z.$$

On the other hand, a non-clique edge sharing a vertex with the clique has $k(1 + 2p)$ neighbors at most, on average. Therefore, for a k-clique to be a robust fixed-point, this forces again from (2):

$$k(1 + 2p)x + o(x\sqrt{k}\ln k) \le z,$$

and any other edges will disappear when this holds. ($o(\cdot)$ is "little-o" notation.)

It follows that the optimal setting (7) for x minimizing probability flow gives robust storage (with a single parallel dynamics update) of all k-cliques for $p < 1/4$. This proves Theorem 1 (see Sect. 5 for the full mathematical details).

It is possible to do better than robustness index $\alpha = 1/4$ by setting $x = \frac{1}{2}[\frac{z}{2k} + \frac{z}{k(1+2p)}] = \frac{z(3+2p)}{4k(1+2p)}$, which satisfies the above fixed-point requirements with probability approaching 1 for any fixed $p < 1/2$ and increasing k. We have thus also demonstrated:

Theorem 2 *There is a family of Hopfield networks on $n = \binom{2k}{2}$ nodes that robustly store $\binom{2k}{k} \sim \frac{2^{\sqrt{2n}+\frac{1}{4}}}{n^{1/4}\sqrt{\pi}}$ binary patterns with maximal robustness index $\alpha = 1/2$.*

In Fig. 4, we show robust storage of the ($\approx 10^{37}$) 64-cliques in graphs on 128 vertices using three (x, y, z) parameter specializations designed here.

A natural question is whether we can store a range of cliques using the same architecture. In fact, we show here that there is a network storing nearly all cliques.

Theorem 3 *For large v, there is a Hopfield network on $n = \binom{v}{2}$ nodes that stores all $\sim 2^v(1 - e^{-Cv})$ cliques of size k as fixed-points, where k is in the range:*

$$m = \frac{1}{D}v \le k \le v = M,$$

for constants $C \approx 0.43$, $D \approx 13.93$. Moreover, this is the largest possible range of k for any such Hopfield network.

Our next result demonstrates that even robustness to vanishingly small amounts of noise is nontrivial (see Sect. 5.5 for the proof).

Theorem 4 *Hopfield–Platt networks storing all permutations will not robustly store derangements (permutations without fixed-points).*

As a final application to biologically plausible learning theory, we derive a synaptic update rule for adapting weights and thresholds in these networks. Given a training pattern **x**, the *minimum probability flow* (MPF) learning rule moves weights and
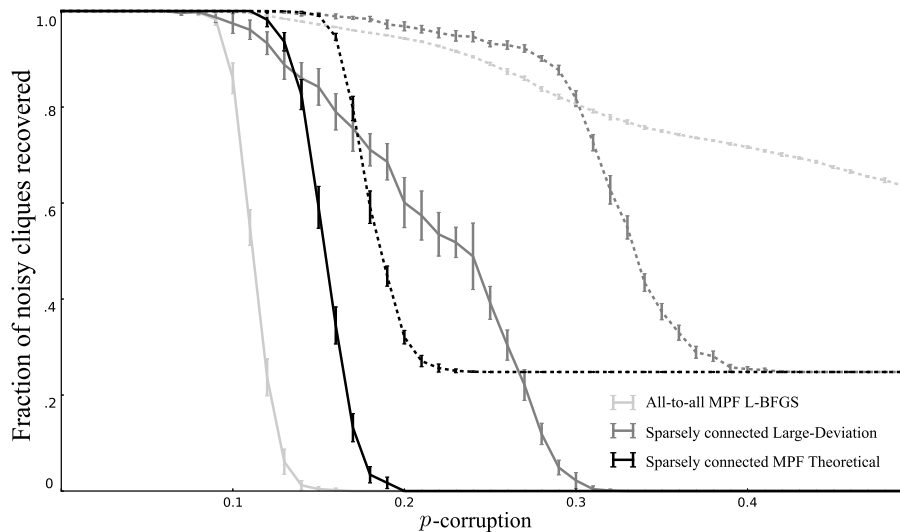
Fig. 4 Robust exponential storage in networks of McCulloch–Pitts neurons. Error-correction performance of Hopfield networks storing all 64-cliques in $v = 128$ vertex graphs using a fully connected 8128-bit network minimizing probability flow (5) on 50,000 random 64-cliques (*light gray line*), a sparsely connected $(x, 0, 1)$ network with large deviation setting $x = \frac{3+2p}{4k(1+2p)}$ and $p = 1/4$ (*gray*), or a sparsely connected MPF theoretical optimum (7) (*black*). Over 10 trials, 100 64-cliques chosen uniformly at random were p-corrupted for different p and then dynamics were converged initialized at noisy cliques. The plot shows the fraction of cliques completely recovered vs. pattern corruption p (standard deviation error bars). *Dotted lines* are average number of bits in a pattern retrieved correctly after converging network dynamics

thresholds in the direction of steepest descent of the probability flow objective function (5) evaluated at $X = \{\mathbf{x}\}$. Specifically, for $e \neq f$ the rule takes the form:

$$\Delta W_{ef} \propto -x_f \Delta x_e \exp(-\Delta E_e/2),$$
$$\Delta \theta_e \propto \Delta x_e \exp(-\Delta E_e/2). \tag{8}$$

After learning, the weights between neurons e and f are symmetrized to $\frac{1}{2}(W_{ef} + W_{fe})$, which preserves the energy function and guarantees that dynamics terminates in fixed-point attractors. As update directions (8) descend the gradient of an infinitely differentiable convex function, learning rules based on them have good convergence rates [23].

Let us examine the (symmetrized) learning rule (8) more closely. Suppose first that $x_e = 0$ so that $\Delta x_e = 0$ or 1 (depending on the sign of $I_e - \theta_e$). When $\Delta x_e = 0$, weight W_{ef} does not change; on the other hand, when $\Delta x_e = 1$, the weight decreases if $x_f = 1$ (and stays the same, otherwise). If instead $x_e = 1$, then W_{ef} changes only if $\Delta x_e = -1$ or $\Delta x_f = -1$, in which case the update is positive when at least one of x_e, x_f is 1 (and zero, otherwise). In particular, either (i) weights do not change (when the pattern is memorized or there is no neural activity) or (ii) when neurons e and f are both active in (8), weights increase, while when they are different, they decrease, consistent with Hebb's postulate [9], a basic hypothesis about neural synaptic plasticity. In fact, approximating the exponential function with unity in (8) gives a variant

of classical outer-product rule (OPR) learning. Note also that adaptation (8) is *local* in that updating weights between 2 neurons only requires their current state/threshold and feedforward input from nearby active neurons.

4 Discussion

The biologically inspired networks introduced in this work constitute a new nonlinear error-correcting scheme that is simple to implement, parallelizable, and achieves the most asymptotic error tolerance possible [24] for low-density codes over a binary symmetric channel ($\alpha = 1/2$ in definition (4)). There have been several other approaches to optimal error-correcting codes derived from a statistical physics perspective; for a comprehensive account, we refer the reader to [25]. See also [26–29] for related work on neural architectures with large memory. Additionally, for a recent review of memory principles in computational neuroscience theory more broadly, we refer the reader to the extensive high level summary [30].

Although we have focused on minimizing probability flow to learn parameters in our discrete neural networks, several other strategies exist. For instance, one could maximize the (Bayesian) likelihood of cliques given network parameters, though any strategy involving a partition function over graphs might run into challenging algorithmic complexity issues [31]. Contrastive divergence [17] is another popular method to estimate parameters in discrete maximum entropy models. While this approach avoids the partition function, it requires a nontrivial sampling procedure that precludes exact determination of optimal parameters.

Early work in the theory of neural computation put forward a framework for neurally plausible computation of (combinatorial) optimization tasks [32]. Here, we add another task to this list by interpreting error-correction by a recurrent neural network in the language of computational graph theory. A basic challenge in this field is to design efficient algorithms that recover structures imperfectly hidden inside of others; in the case of finding fully connected subgraphs, this is called the "Hidden clique problem" [33]. The essential goal of this task is to find a single clique that has been planted in a graph by adding (or removing) edges at random.

Phrased in this language, we have discovered discrete recurrent neural networks that learn to use their cooperative McCulloch–Pitts dynamics to solve hidden clique problems efficiently. For example, in Fig. 5 we show the adjacency matrices of three corrupted 64-cliques on $v = 128$ vertices returning to their original configuration by one iteration of the network dynamics through all neurons. As a practical matter, it is possible to use networks robustly storing k-cliques for detecting highly connected subgraphs with about k neighbors in large graphs. In this case, error-correction serves as a synchrony finder with free parameter k, similar to how "K-means" is a standard unsupervised approach to decompose data into K clusters.

In the direction of applications to basic neuroscience, we comment that it has been proposed that co-activation of groups of neurons—that is, synchronizing them—is a design principle in the brain (see, e.g., [34–36]). If this were true, then perhaps the networks designed here can help discover this phenomenon from spike data. Moreover, our networks also then provide an abstract model for how such coordination might be implemented, sustained, and error-corrected in nervous tissue.
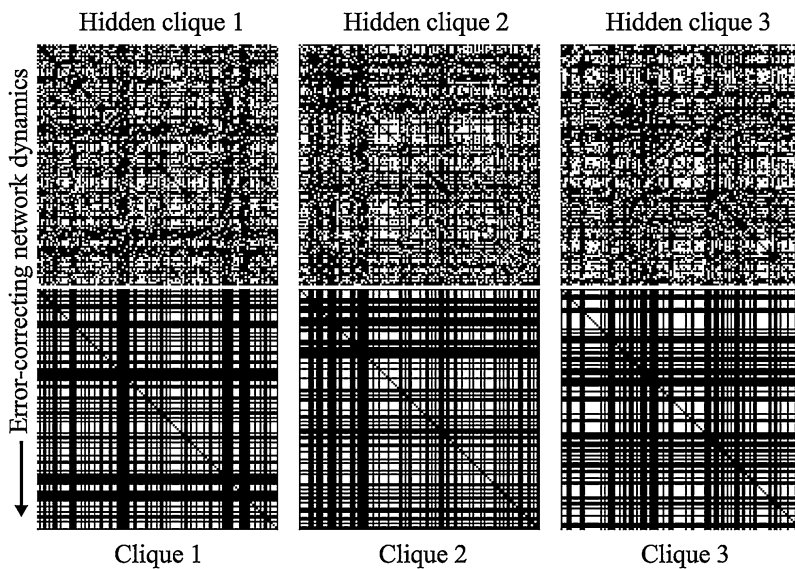
Fig. 5 Learning to solve $\approx 10^{37}$ "Hidden clique" problems. (*Bottom*) Adjacency matrices of three 64-cliques on $v = 128$ vertices. (*Top*) Adjacency matrices of noisy versions of the cliques having, on average, 1219 bits corrupted out of $n = 8128$ from the original. Converging dynamics of a symmetric 3-parameter network $(x, y, z) = (0.0107, 0, 1)$ with minimum probability flow initialized at these noisy cliques uncovers the originals

As a final technical remark about our networks, note that our synapses are actually discrete since the probability flow is minimized at a synaptic ratio equaling a rational number. Thus, our work adds to the literature on the capacity of neural networks with discrete synapses (see, e.g., [26, 37–40]), all of which build upon early classical work with associative memory systems (see, e.g., [20, 41]).

5 Mathematical Details

We provide the remaining details for the proofs of mathematical statements appearing earlier in the text.

5.1 Symmetric 3-Parameter (x, y, z) Networks

The first step of our construction is to exploit symmetry in the following set of linear inequalities:

$$E_{\mathbf{c}} - E_{\mathbf{c}'} < 0, \tag{9}$$

where $\mathbf{c}$ runs over k-cliques and $\mathbf{c}'$ over vectors differing from $\mathbf{c}$ by a single bit flip. The space of solutions to (9) is the convex polyhedral cone of networks having each clique as a strict local minimum of the energy function, and thus a fixed-point of the dynamics.

The permutations $P \in P_V$ of the vertices V act on a network by permuting the rows/columns of the weight matrix ($\mathbf{W} \mapsto P\mathbf{W}P^{\top}$) and thresholds ($\theta \mapsto P\theta$), and this action on a network satisfying property (9) preserves that property. Consider the average $(\overline{\mathbf{W}}, \bar{\theta})$ of a network over the group P_V: $\overline{\mathbf{W}} := \frac{1}{v!} \sum_{P \in P_V} P\mathbf{W}P^{\top}, \bar{\theta} := \frac{1}{v!} \sum_{P \in P_V} P\theta$, and note that if $(\mathbf{W}, \theta)$ satisfies (9) then so does the highly symmetric object $(\overline{\mathbf{W}}, \bar{\theta})$. To characterize $(\overline{\mathbf{W}}, \bar{\theta})$, observe that $P\overline{\mathbf{W}}P^{\top} = \overline{\mathbf{W}}$ and $P\bar{\theta} = \bar{\theta}$ for all $P \in P_V$.

These strong symmetries imply there are x, y, z such that $\bar{\theta} = (z, \ldots, z) \in \mathbb{R}^n$ and for each pair $e \neq f$ of all possible edges:

$$\overline{W}_{ef} = \begin{cases} x & \text{if } |e \cap f| = 1, \\ y & \text{if } |e \cap f| = 0, \end{cases}$$

where $|e \cap f|$ is the number of vertices that e and f share.

Our next demonstration is an exact setting for weights in these Hopfield networks.

5.2 Exponential Storage

For an integer $r \geq 0$, we say that state $\mathbf{x}^*$ is r-*stable* if it is an attractor for all states with Hamming distance at most r from $\mathbf{x}^*$. Thus, if a state $\mathbf{x}^*$ is r-stably stored, the network is guaranteed to converge to $\mathbf{x}^*$ when exposed to any corrupted version not more than r bit flips away.

For positive integers k and r, is there a Hopfield network on $n = \binom{2k}{2}$ nodes storing all k-cliques r-stably? We necessarily have $r \leq \lfloor k/2 \rfloor$, since $2(\lfloor k/2 \rfloor + 1)$ is greater than or equal to the Hamming distance between two k-cliques that share a $(k-1)$-subclique. In fact, for any $k > 3$, this upper bound is achievable by a sparsely connected three-parameter network.

Lemma 1 *There exists a family of three-parameter Hopfield networks with $z = 1$, $y = 0$ storing all k-cliques as $\lfloor k/2 \rfloor$-stable states.*

The proof relies on the following lemma, which gives the precise condition for the three-parameter Hopfield network to store k-cliques as r-stable states for fixed r.

Lemma 2 *Fix $k > 3$ and $0 \leq r < k$. The Hopfield network $(\mathbf{W}(x, y), \theta(z))$ stores all k-cliques as r-stable states if and only if the parameters $x, y, z \in \mathbb{R}$ satisfy*

$$M \cdot \begin{bmatrix} x \\ y \end{bmatrix} < \begin{bmatrix} -2 \\ -2 \\ 2 \\ 2 \end{bmatrix} z,$$

where

$$M = \begin{bmatrix} 4(2-k)+2r & (2-k)(k-3) \\ 4(2-k) & (2-k)(k-3)-2r \\ 2(k-1)+2r & (k-1)(k-2) \\ 2(k-1) & (k-1)(k-2)-2r \end{bmatrix}.$$

Furthermore, *a pattern within Hamming distance r of a k-clique converges after one iteration of the dynamics.*

Proof For fixed r and k-clique $\mathbf{x}$, there are 2^r possible patterns within Hamming distance r of $\mathbf{x}$. Each of these patterns defines a pair of linear inequalities on the parameters x, y, z. However, only the inequalities from the following two extreme cases are active constraints. All the other inequalities are convex combinations of these.

1. r edges in the clique with a common node i are removed.
2. r edges are added to a node i not in the clique.

In the first case, there are two types of edges at risk of being mislabeled. The first are those of the form ij for all nodes j in the clique. Such an edge has $2(k-2) - r$ neighbors and $\binom{k-2}{2}$ non-neighbors. Thus, each such edge will correctly be labeled 1 after one network update if and only if x, y, and z satisfy

$$2(2k - r - 4)x + (k-2)(k-3)y > 2z. \tag{10}$$

The other type are those of the form $\bar{i}j$ for all nodes $\bar{i} \neq i$ in the clique, and j not in the clique. Assuming $r < k - 1$, such an edge has at most $k - 1$ neighbors and $\binom{k-1}{2} - r$ non-neighbors. Thus, each such edge will be correctly labeled 0 if and only if

$$2(k-1)x + \big((k-1)(k-2) - 2r\big)y < 2z. \tag{11}$$

Rearranging Eqs. (10) and (11) yield the first two rows of the matrix in the lemma. A similar argument applies for the second case, giving the last two inequalities.

From the derivation, it follows that if a pattern is within Hamming distance r of a k-clique, then all spurious edges are immediately deleted by case 1, all missing edges are immediately added by case 2, and thus the clique is recovered in precisely one iteration of the network dynamics. $\qquad\square$

Proof of Lemma 1 The matrix inequalities in Lemma 2 define a cone in $\mathbb{R}^3$, and the cases $z = 1$ or $z = 0$ correspond to two separate components of this cone. For the proof of Theorem 1 in the main article, we use the cone with $z = 1$. We further assume $y = 0$ to achieve a sparsely connected matrix $\mathbf{W}$. In this case, the second and fourth constraints are dominated by the first and third. Thus, we need x that solves

$$\frac{1}{2(k-1) - r} < x < \frac{1}{k - 1 + r}.$$

There exists such a solution if and only if

$$2(k-1) - r > k - 1 + r \quad \Leftrightarrow \quad k > 2r + 1. \tag{12}$$

The above equation is feasible if and only if $r \leq \lfloor k/2 \rfloor$. $\qquad\square$

5.3 Proofs of Theorems 1, 2

Fix $y = 0$ and $z = 1$. We now tune x such that asymptotically the α-robustness of our set of Hopfield networks storing k-cliques tends to $1/2$ as $n \to \infty$. By symmetry, it is sufficient to prove robustness for one fixed k-clique $\mathbf{x}$; for instance, the one with vertices $\{1, \dots, k\}$. For $0 < p < \frac{1}{2}$, let $\mathbf{x}_p$ be the p-corruption of $\mathbf{x}$. For each node $i \in \{1, \dots, 2k\}$, let $i_{\text{in}}, i_{\text{out}}$ denote the number of edges from i to other clique and non-clique nodes, respectively. With an abuse of notation, we write $i \in \mathbf{x}$ to mean a vertex i in the clique; that is, $i \in \{1, \dots, k\}$. We need the following inequality originally due to Bernstein from 1924.

Proposition 1 (Bernstein's inequality [22]) *Let S_i be independent Bernoulli random variables taking values $+1$ and -1, each with probability $1/2$. For any $\varepsilon > 0$, the following holds*:

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} S_i > \varepsilon\right) \leq \exp\left(-\frac{n\varepsilon^2}{2 + 2\varepsilon/3}\right).$$

The following fact is a fairly direct consequence of Proposition 1.

Lemma 3 *Let Y be an $n \times n$ symmetric matrix with zero diagonal, $Y_{ij} \overset{\text{i.i.d.}}{\sim}$ Bernoulli(p). For each $i = 1, \dots, n$, let $Y_i = \sum_j Y_{ij}$ be the ith row sum. Let $M_n = \max_{1 \leq i \leq n} Y_i$, and $m_n = \min_{1 \leq i \leq n} Y_i$. Then, for any constant $c > 0$, as $n \to \infty$, we have*

$$\mathbb{P}\left(|m_n - np| > c\sqrt{n}\ln n\right) \to 0$$

and

$$\mathbb{P}\left(|M_n - np| > c\sqrt{n}\ln n\right) \to 0.$$

In particular, $|m_n - np|, |M_n - np| = o(\sqrt{n}\ln n)$.

Proof Fix $c > 0$. As a direct corollary of Bernstein's inequality, for each i and for any $\varepsilon > 0$, we have

$$\mathbb{P}\left(Y_i - np > n\varepsilon - (p + \varepsilon)\right) \leq \exp\left(-\frac{(n-1)\varepsilon^2}{2 + 2\varepsilon/3}\right).$$

It follows that

$$\mathbb{P}(Y_i - np > n\varepsilon) \leq \exp\left(-\frac{n\varepsilon^2}{4 + 4\varepsilon/3}\right),$$

and thus from a union bound with $\varepsilon = \frac{c\ln n}{\sqrt{n}}$, we have

$$\mathbb{P}\left(\max_i Y_i - np > c\sqrt{n}\ln n\right) \leq \exp\left(-\frac{n\varepsilon^2}{4 + 4\varepsilon/3} + \ln n\right)$$

$$\leq \exp\left(-\frac{c^2\ln^2 n}{4 + 4c} + \ln n\right).$$

Since this last bound converges to 0 with $n \to \infty$, we have proved the claim for M_n. Since Y_i is symmetric about np, a similar inequality holds for m_n. ☐

Corollary 1 *Let $M_{\text{in}} = \max_{i \in \mathbf{x}} i_{\text{in}}$, $m_{\text{in}} = \min_{i \in \mathbf{x}} i_{\text{in}}$, $M_{\text{out}} = \max_{i \notin \mathbf{x}} i_{\text{out}}$, $m_{\text{out}} = \min_{i \notin \mathbf{x}} i_{\text{out}}$, and $M_{\text{between}} = \max_{i \notin \mathbf{x}} i_{\text{in}}$. Then $M_{\text{in}} - k(1 - p)$, $m_{\text{in}} - k(1 - p)$, $M_{\text{out}} - kp$, $m_{\text{out}} - kp$, and $M_{\text{between}} - kp$ are all of order $o(\sqrt{k} \ln k)$ as $k \to \infty$ almost surely.*

Proofs of Theorems 1, 2 (robustness) Let $N(e)$ be the number of neighbors of edge e. For each e in the clique:

$$N(e) \geq 2m_{\text{in}} + 2m_{\text{out}} \sim 2k + o(\sqrt{k} \ln k), \quad \text{w.h.p. (with high probability).}$$

To guarantee that all edges e in the clique are labeled 1 after one dynamics update, we need $x > \frac{1}{N(e)}$; that is,

$$x > \frac{1}{2k + o(\sqrt{k} \ln k)}. \tag{13}$$

If f is an edge with exactly one clique vertex, then we have

$$N(f) \leq M_{\text{in}} + M_{\text{out}} + 2M_{\text{between}}$$

$$\sim k(1 + 2p) + o(\sqrt{k} \ln k), \quad \text{w.h.p.}$$

To guarantee that $\mathbf{x}_f = 0$ for all such edges f after one iteration of the dynamics, we need $x < \frac{1}{N(f)}$; that is,

$$x < \frac{1}{k(1 + 2p) + o(\sqrt{k} \ln k)}. \tag{14}$$

In particular, if $p = p(k) \sim \frac{1}{2} - k^{\delta - 1/2}$ for some small $\delta \in (0, 1/2)$, then taking $x = x(k) = \frac{1}{2}[\frac{1}{2k} + \frac{1}{k(1+2p)}]$ would guarantee that for large k the two inequalities (13) and (14) are simultaneous satisfied. In this case, $\lim_{k \to \infty} p(k) = 1/2$, and thus the family of two-parameter Hopfield networks with $x(k)$, $y = 0$, $z = 1$ has robustness index $\alpha = 1/2$. ☐

5.4 Clique Range Storage

In this section, we give precise conditions for the existence of a Hopfield network on $\binom{v}{2}$ nodes that stores all k-cliques for k in an interval $[m, M]$, $m \leq M \leq v$. We do not address the issue of robustness as the qualitative trade-off is clear: the more memories the network is required to store, the less robust it is. This trade-off can be analyzed by large deviation principles as in Theorem 2.

Lemma 4 *Fix m such that $3 \leq m < v$. For $M \geq m$, there exists a Hopfield network on $\binom{v}{2}$ nodes that stores all k-cliques in the range $[m, M]$ if and only if M solves the*

implicit equation $x_M - x_m < 0$, *where*

$$x_m = \frac{-(4m - \sqrt{12m^2 - 52m + 57} - 7)}{2(m^2 - m - 2)},$$

$$x_M = \frac{-(4M + \sqrt{12M^2 - 52M + 57} - 7)}{2(M^2 - M - 2)}.$$

Proof Fix $z = 1/2$ and $r = 0$ in Lemma 1. (We do not impose the constraint $y = 0$.) Then the cone defined by the inequalities in Lemma 1 is in bijection with the polyhedron $\mathcal{I}_k \subseteq \mathbb{R}^2$ cut out by inequalities:

$$4(k - 2)x + (k - 2)(k - 3)y - 1 > 0,$$

$$2(k - 1)x + (k - 1)(k - 2)y - 1 < 0.$$

Let R_k be the line $4(k - 2)x + (k - 2)(k - 3)y - 1 = 0$, and B_k be the line $2(k - 1)x + (k - 1)(k - 2)y - 1 = 0$. By symmetry, there exists a Hopfield network that stores all k-cliques in the range $[m, M]$ if and only if $\bigcap_{k=m}^{M} \mathcal{I}_k \neq \emptyset$. For a point $P \in \mathbb{R}^2$, write $x(P)$ for its x-coordinate. Note that, for $k \geq 3$, the points $B_k \cap B_{k+1}$ lie on the following curve Q implicitly parametrized by k:

$$Q := \left\{ \left(\frac{1}{k - 1}, \frac{-1}{(k - 1)(k - 2)} \right) : k \geq 3 \right\}.$$

When the polytope $\bigcap_{k=m}^{M} \mathcal{I}_k$ is nonempty, its vertices are the following points: $R_M \cap R_m$, $R_M \cap B_m$, $B_k \cap B_{k+1}$ for $m \leq k \leq M - 1$, and the points $B_M \cap R_m$. This defines a nonempty convex polytope if and only if

$$x_M := x(Q \cap R_M) < x_m := x(Q \cap R_m).$$

Direct computation gives the formulas for x_m, x_M in the lemma statement. See Fig. 6 for a visualization of the constraints of the feasible region. □

Fixing the number of nodes and optimizing the range $M - m$ in Lemma 4, we obtain Theorem 3 from Sect. 3.

Proof of Theorem 3 From Lemma 4, for large m, M, and v, we have the approximations $x_m \approx \frac{\sqrt{12} - 4}{2m}$, $x_M \approx \frac{-\sqrt{12} - 4}{2M}$. Hence $x_M - x_m < 0$ when $M \lesssim \frac{2 + \sqrt{3}}{2 - \sqrt{3}} m = Dm$. Asymptotically for large v, the most cliques are stored when $M = Dm$ and $[m, M]$ contains $v/2$. Consider $m = \beta v$ so that $v \geq M = D\beta v \geq v/2$, and thus $1/D \geq \beta \geq 1/(2D)$. Next, set $u = v/2 - m = v(1/2 - \beta)$ and $w = M - v/2 = v(D\beta - 1/2)$ so that storing the most cliques becomes the problem of maximizing over admissible β the quantity:

$$\max\{u, w\} = \max\{v(1/2 - \beta), v(D\beta - 1/2)\}.$$

One can now check that $\beta = 1/D$ gives the best value, producing the range in the statement of the theorem.
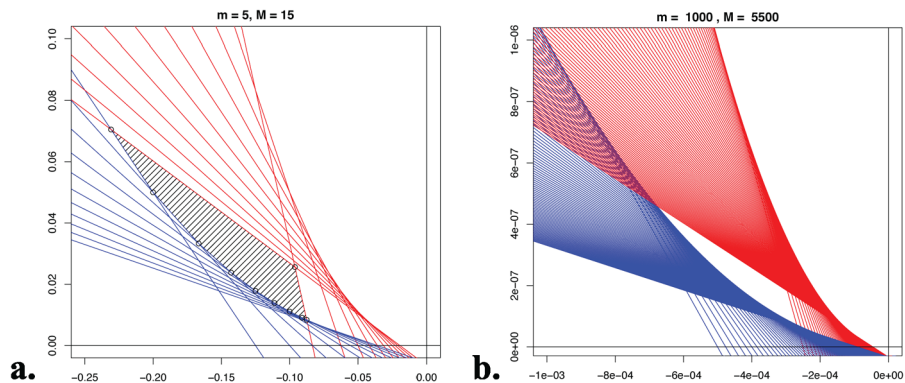
Fig. 6 Feasible exponential storage. **(a)** The shaded region is the feasible polytope for network parameters giving clique storage for $5 \leq k \leq 15$. *Black points* are its vertices, *the red R_k and blue B_k lines* are linear constraints. **(b)** Lines R_k (*red*) and B_k (*blue*) for $1000 \leq k \leq 5500$. Note the appearance of the smooth curve Q enveloping the family B_k in the figure

Next, note that $\binom{v}{k} 2^{-v}$ is the fraction of k-cliques in all cliques on v vertices, which is also the probability of a $\text{Binom}(v, 1/2)$ variable equaling k. For large v, approximating this variable with a normal distribution and then using Mill's ratio to bound its tail c.d.f. Φ, we see that the proportion of cliques storable tends to

$$1 - \Phi\left(\frac{D-1}{D}\sqrt{v}\right) \approx 1 - \exp(-Cv),$$

for some constant $C \approx \frac{(D-1)^2}{2D^2} \approx 0.43$.   □

5.5 Hopfield–Platt Networks

We prove the claim in the main text that Hopfield–Platt networks [13] storing all permutations on $\{1, \ldots, k\}$ will not robustly store *derangements* (permutations without fixed-points). For large k, the fraction of permutations that are derangements is known to be $e^{-1} \approx 0.36$.

Proof of Theorem 4 Fix a derangement σ on $\{1, \ldots, k\}$, represented as a binary vector $\mathbf{x}$ in $\{0, 1\}^n$ for $n = k(k-1)$. For each ordered pair (i, j), $i \neq j$, $j \neq \sigma(i)$, we construct a pattern $\mathbf{y}_{ij}$ that differs from $\mathbf{x}$ by exactly two bit flips:

1. Add the edge ij.
2. Remove the edge $i\sigma(i)$.

There are $k(k-2)$ such pairs (i, j), and thus $k(k-2)$ different patterns $\mathbf{y}_{ij}$. For each such pattern, we flip two more bits to obtain a new permutation $\mathbf{x}^{ij}$ as follows:

1. Remove the edge $\sigma^{-1}(j)j$.
2. Add the edge $\sigma^{-1}(j)\sigma(i)$.

It is easy to see that $\mathbf{x}^{ij}$ is a permutation on k letters with exactly two cycles determined by (i, j). Call the set of edges modified the *critical edges* of the pair (i, j). Note that $\mathbf{x}^{ij}$ are all distinct and have disjoint critical edges.

Each $\mathbf{y}_{ij}$ is exactly two bit flips away from $\mathbf{x}$ and $\mathbf{x}^{ij}$, both permutations on k letters. Starting from $\mathbf{y}_{ij}$, there is no binary Hopfield network storing all permutations that always correctly recovers the original state. In other words, for a binary Hopfield network, $\mathbf{y}_{ij}$ is an *indistinguishable* realization of a corrupted version of $\mathbf{x}$ and $\mathbf{x}^{ij}$.

We now prove that, for each derangement $\mathbf{x}$, with probability at least $1 - (1 - 4p^2)^{n/2}$, its p-corruption $\mathbf{x}_p$ is indistinguishable from the p-corruption of some other permutation. This implies the statement in the theorem.

For each pair (i, j) as above, recall that $\mathbf{x}_p$ and $\mathbf{x}_p^{ij}$ are two random variables in $\{0, 1\}^n$ obtained by flipping each edge of $\mathbf{x}$ (resp. $\mathbf{x}^{ij}$) independently with probability p. We construct a coupling between them as follows. Define the random variable $\mathbf{x}_p'$ via:

- For each non-critical edge, flip this edge on $\mathbf{x}_p'$ and $\mathbf{x}^{ij}$ with the same Bernoulli(p).
- For each critical edge, flip them on $\mathbf{x}_p'$ and $\mathbf{x}^{ij}$ with independent Bernoulli(p).

Then $\mathbf{x}_p' \stackrel{d}{=} \mathbf{x}_p$ have the same distribution, and $\mathbf{x}_p'$ and $\mathbf{x}_p^{ij}$ only differ in distribution on the four critical edges. Their marginal distributions on these four edges are two discrete variables on 2^4 states, with total variation distance $1 - 4(1 - p)^2 p^2$. Thus, there exists a random variable $\mathbf{x}_p''$ such that $\mathbf{x}_p'' \stackrel{d}{=} \mathbf{x}_p' \stackrel{d}{=} \mathbf{x}_p$, and

$$\mathbb{P}\big(\mathbf{x}_p'' = \mathbf{x}_p^{ij}\big) = 4(1 - p)^2 p^2.$$

In other words, given a realization of $\mathbf{x}_p^{ij}$, with probability $4(1 - p)^2 p^2$, this is equal to a realization from the distribution of $\mathbf{x}_p$, and therefore no binary Hopfield network storing both $\mathbf{x}^{ij}$ and $\mathbf{x}$ can correctly recover the original state from such an input. An indistinguishable realization occurs when two of the four critical edges are flipped in a certain combination. For fixed $\mathbf{x}$, there are $k(k - 2)$ such $\mathbf{x}^{ij}$ where the critical edges are disjoint. Thus, the probability of $\mathbf{x}_p$ being an indistinguishable realization from a realization of one of the $\mathbf{x}^{ij}$ is at least

$$1 - \big(1 - 4(1 - p)^2 p^2\big)^{k(k-2)} > 1 - \big(1 - 4p^2\big)^{n/2},$$

completing the proof of Theorem 4. □

5.6 Examples of Clique Storage

To illustrate the effect of two different noise levels on hidden clique finding performance of the networks from Fig. 4, we present examples in Fig. 7 of multiple networks acting with their dynamics on the same two noisy inputs. Notice that non-clique fixed-points appear, and it is natural to ask whether a complete characterization of the fixed-point landscape is possible. Intuitively, our network performs a local, weighted degree count at each edge of the underlying graph and attempts to remove edges with too few neighbors, while adding in edges that connect nodes with high
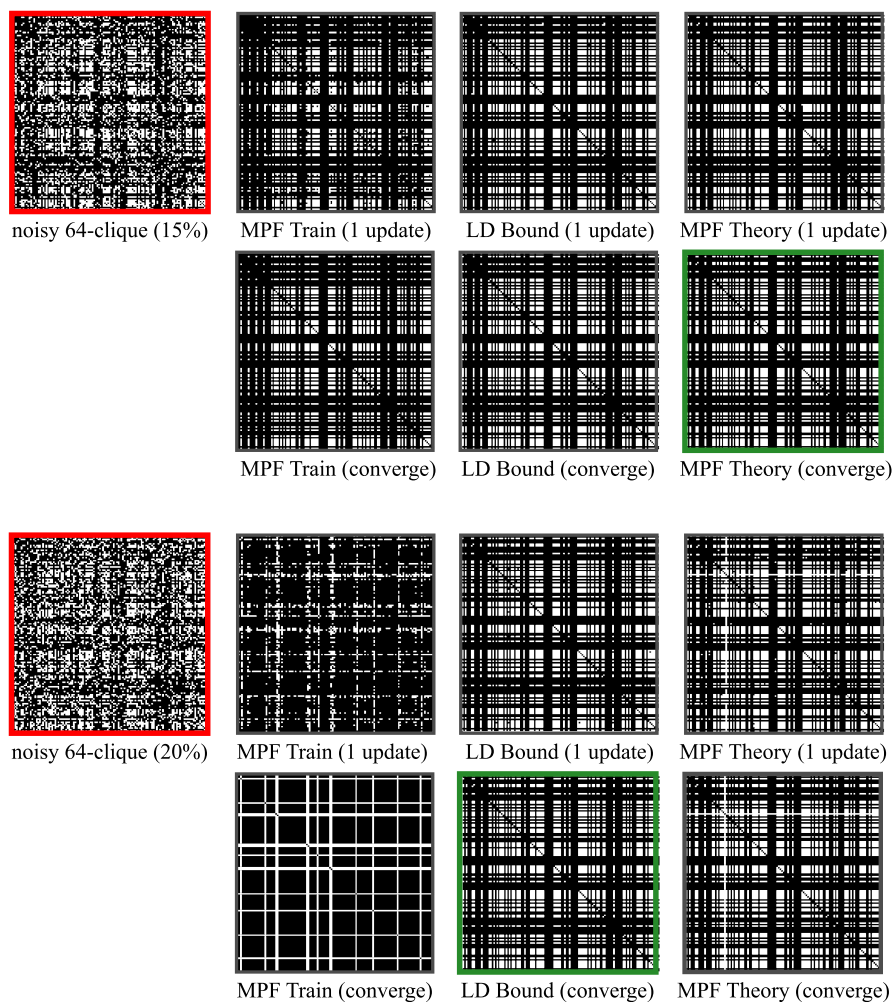
noisy 64-clique (15%)	MPF Train (1 update)	LD Bound (1 update)	MPF Theory (1 update)
	MPF Train (converge)	LD Bound (converge)	MPF Theory (converge)
noisy 64-clique (20%)	MPF Train (1 update)	LD Bound (1 update)	MPF Theory (1 update)
	MPF Train (converge)	LD Bound (converge)	MPF Theory (converge)

Fig. 7 Examples of robustness for networks in Fig. 4 of main text with $v = 128$, $k = 64$, $n = 8128$. Adjacency matrices of noisy cliques (in *red*) have 1219 (*top*) or 1625 (*bottom*) bits corrupted out of 8128 ($p = 0.15/0.2$) from the original 64-clique (in *green*). Images show the result of dynamics applied to these noisy patterns using networks with all-to-all MPF parameters after L-BFGS training on 50,000 64-cliques ($\approx 2e{-}31\%$ of all 64-cliques), Large deviation parameters $(x, y, z) = (0.0091, 0, 1)$, or MPF Theory parameters $(x, y, z) = (0.0107, 0, 1)$ from Eq. (7) in the main text

degrees. Thus, resulting fixed-points (of the dynamics) end up being graphs such as cliques and stars. Beyond this intuition, however, we do not have a way to characterize all fixed-points of our network in general.

In fact, this is a very difficult problem in discrete geometry, and except for toy networks, we believe that this has never been done. Geometrically, the set of all states of a binary Hopfield network with n neurons is the n-hypercube $\{0, 1\}^n$. Being a fixed-point can be characterized by the energy function becoming larger when one bit is flipped. As the energy function is quadratic, for each of the n bits flipped, this creates

a quadratic inequality. Thus, the set of all fixed-point attractors in a binary Hopfield network is the n-hypercube intersected with n quadratic inequalities in n variables. In theory, one could enumerate such sets for small n; however, characterizing them all is challenging, even for the highly symmetric family of weight matrices that we propose here.

Availability of data and materials   Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Ethics approval and consent to participate   Not applicable.

Competing interests   The authors declare that they have no competing interests.

Consent for publication   Not applicable.

Authors' contributions   CJH and NMT contributed equally. All authors read and approved the final manuscript.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Pastur L, Figotin A. Exactly soluble model of a spin glass. Sov J Low Temp Phys. 1977;3:378–83.
2. Edwards S, Anderson P. Theory of spin glasses. J Phys F, Met Phys. 1975;5(5):965.
3. Hopfield J. Neural networks and physical systems with emergent collective computational abilities. Proc Natl Acad Sci USA. 1982;79(8):2554.
4. McCulloch W, Pitts W. A logical calculus of the ideas immanent in nervous activity. Bull Math Biol. 1943;5(4):115–33.
5. McEliece R, Posner E, Rodemich E, Venkatesh S. The capacity of the Hopfield associative memory. IEEE Trans Inf Theory. 1987;33(4):461–82.
6. Amari S-I. Learning patterns and pattern sequences by self-organizing nets of threshold elements. IEEE Trans Comput. 1972;100(11):1197–206.
7. Talagrand M. Spin glasses: a challenge for mathematicians. vol. 46. Berlin: Springer; 2003.
8. Lorente de Nó R. Vestibulo-ocular reflex arc. Arch Neurol Psychiatry. 1933;30(2):245–91.
9. Hebb D. The organization of behavior. New York: Wiley; 1949.
10. Cover T. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. IEEE Trans Comput. 1965;3:326–34.
11. Amari S-I. Characteristics of sparsely encoded associative memory. Neural Netw. 1989;2(6):451–7.
12. Tanaka F, Edwards S. Analytic theory of the ground state properties of a spin glass. I. Ising spin glass. J Phys F, Met Phys. 1980;10:2769.

13. Platt J, Hopfield J. Analog decoding using neural networks. In: Neural networks for computing. vol. 151. Melville: AIP Publishing; 1986. p. 364–9.
14. Sohl-Dickstein J, Battaglino P, DeWeese M. New method for parameter estimation in probabilistic models: minimum probability flow. Phys Rev Lett. 2011;107(22):220601.
15. Hillar C, Sohl-Dickstein J, Koepsell K. Efficient and optimal binary Hopfield associative memory storage using minimum probability flow. In: 4th neural information processing systems (NIPS) workshop on discrete optimization in machine learning (DISCML): structure and scalability. 2012. p. 1–6.
16. Ising E. Beitrag zur theorie des ferromagnetismus. Z Phys. 1925;31:253–8.
17. Ackley D, Hinton G, Sejnowski T. A learning algorithm for Boltzmann machines. Cogn Sci. 1985;9(1):147–69.
18. Turing A. On computable numbers, with an application to the Entscheidungsproblem. Proc Lond Math Soc. 1937;2(1):230–65.
19. Von Neumann J. First draft of a report on the EDVAC. IEEE Ann Hist Comput. 1993;15(4):27–75.
20. Rosenblatt F. Principles of neurodynamics: perceptrons and the theory of brain mechanisms. Washington, DC: Spartan Books; 1961.
21. Nocedal J. Updating quasi-Newton matrices with limited storage. Math Comput. 1980;35(151):773–82.
22. Bernstein S. On a modification of Chebyshev's inequality and of the error formula of Laplace. Ann Sci Inst Sav Ukr, Sect Math. 1924;1(4):38–49.
23. Hazan E, Agarwal A, Kale S. Logarithmic regret algorithms for online convex optimization. Mach Learn. 2007;69(2–3):169–92.
24. Shannon C. A mathematical theory of communication. Bell Syst Tech J. 1948;27:379–423.
25. Vicente R, Saad D, Kabashima Y. Low-density parity-check codes—a statistical physics perspective. Adv Imaging Electron Phys. 2002;125:232–355.
26. Gripon V, Berrou C. Sparse neural networks with large learning diversity. IEEE Trans Neural Netw. 2011;22(7):1087–96.
27. Kumar K, Salavati A, Shokrollahi A. Exponential pattern retrieval capacity with non-binary associative memory. In: Information theory workshop (ITW). New York: IEEE Press; 2011. p. 80–4.
28. Curto C, Itskov V, Morrison K, Roth Z, Walker J. Combinatorial neural codes from a mathematical coding theory perspective. Neural Comput. 2013;25(7):1891–925.
29. Karbasi A, Salavati A, Shokrollahi A, Varshney L. Noise facilitation in associative memories of exponential capacity. Neural Comput. 2014;16(11):2493–526.
30. Chaudhuri R, Fiete I. Computational principles of memory. Nat Neurosci. 2016;19(3):394–403.
31. Jerrum M, Sinclair A. Polynomial-time approximation algorithms for the Ising model. SIAM J Comput. 1993;22(5):1087–116.
32. Hopfield J, Tank D. Computing with neural circuits: a model. Science. 1986;233(4764):625–33.
33. Alon N, Krivelevich M, Sudakov B. Finding a large hidden clique in a random graph. Random Struct Algorithms. 1998;13(3–4):457–66.
34. Singer W. Synchronization of cortical activity and its putative role in information processing and learning. Annu Rev Physiol. 1993;55(1):349–74.
35. Singer W. Neuronal synchrony: a versatile code for the definition of relations? Neuron. 1999;24(1):49–65.
36. Womelsdorf T, Schoffelen J-M, Oostenveld R, Singer W, Desimone R, Engel A, Fries P. Modulation of neuronal interactions through neuronal synchronization. Science. 2007;316(5831):1609–12.
37. Gutfreund H, Stein Y. Capacity of neural networks with discrete synaptic couplings. J Phys A. 1990;23(12):2613–30.
38. Kocher I, Monasson R. On the capacity of neural networks with binary weights. J Phys A. 1992;25(2):367–80.
39. Knoblauch A. Efficient associative computation with discrete synapses. Neural Comput. 2015;28(1):118–86.
40. Alemi A, Baldassi C, Brunel N, Zecchina R. A three-threshold learning rule approaches the maximal capacity of recurrent neural networks. PLoS Comput Biol. 2015;11(8):1004439.
41. Willshaw D, Buneman O, Longuet-Higgins H. Non-holographic associative memory. Nature. 1969;222(5197):960–2.