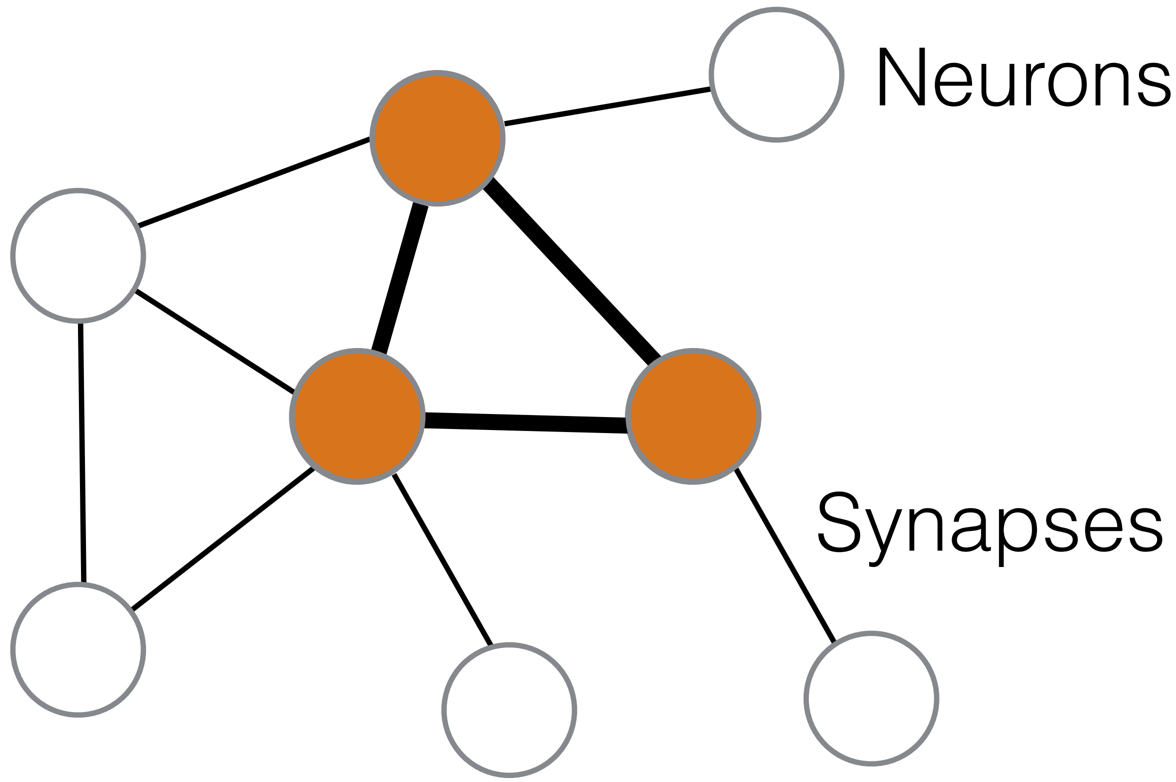
Bipartite expander Hopfield networks as self-decoding high-capacity error correcting codes

Rishidev Chaudhuri, Ila Fiete

Strong connections allow networks of neurons to store memories



Patterns of activity stabilized by strong synaptic weights (Hebb, 1949)

Attractors: self-sustained stable patterns of neural activity

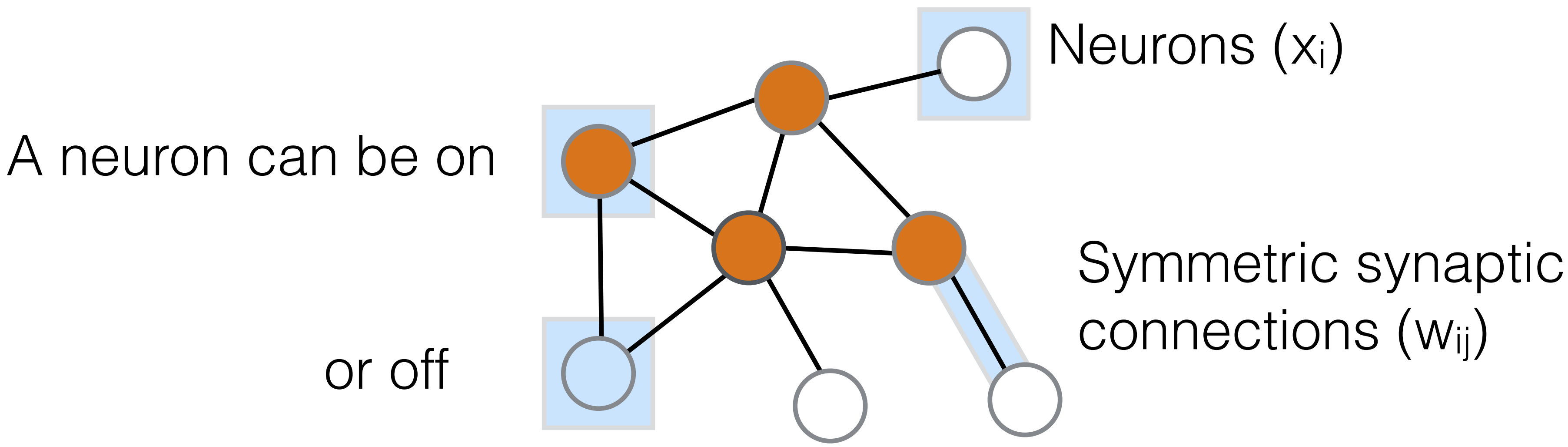
High-capacity attractor networks

Current models either contain large numbers of attractors or show good noise tolerance

Can we construct an attractor network that is both **high-capacity** and **robust**?

Use *expander graphs* to construct a robust high-capacity *Hopfield network*

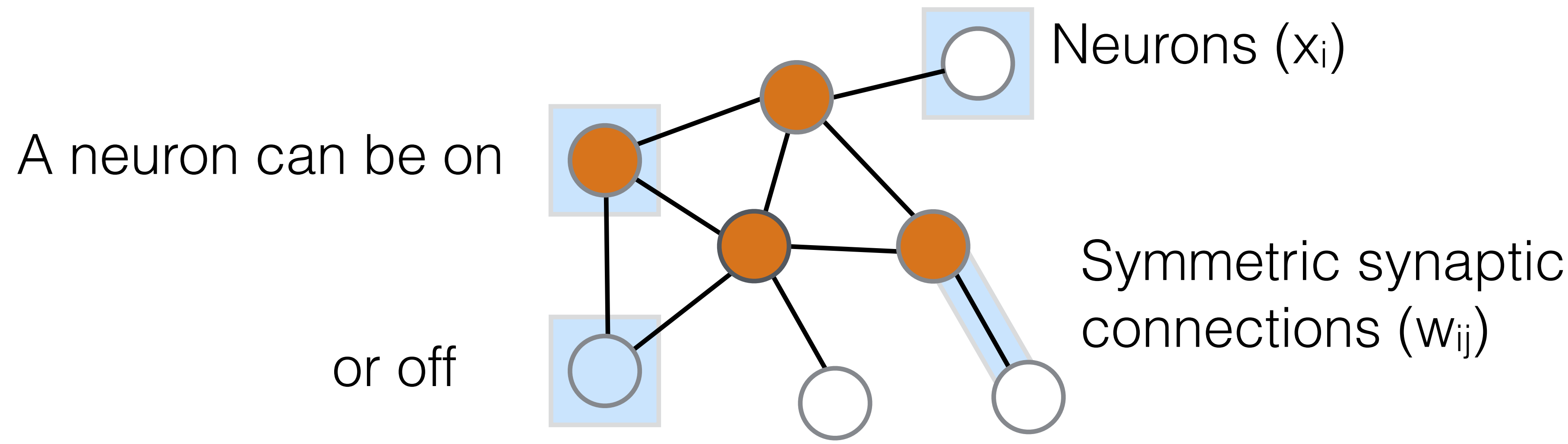
Hopfield networks: canonical models of neural memory



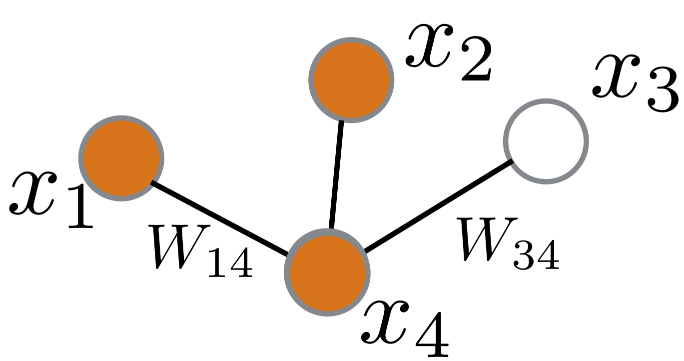
also known as Ising models. Very similar to undirected graphical models

Hopfield 1982, Hinton & Sejnowski 1985

Hopfield networks: canonical models of neural memory



Network dynamics



Input $\sum_i W_{i4} x_i$

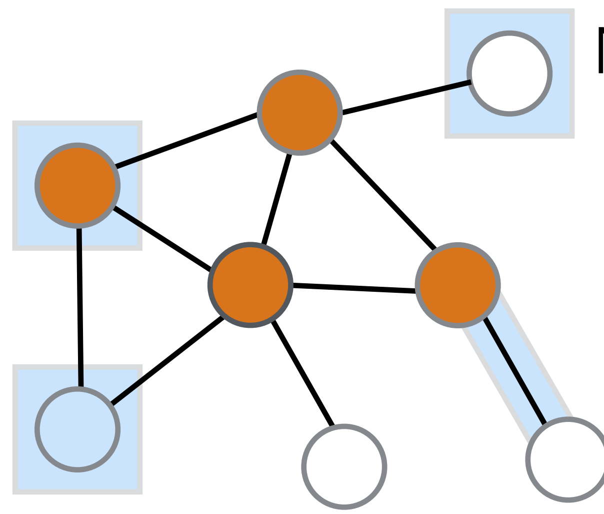
$x_4 = 0$ if input < threshold

$x_4 = 1$ if input > threshold

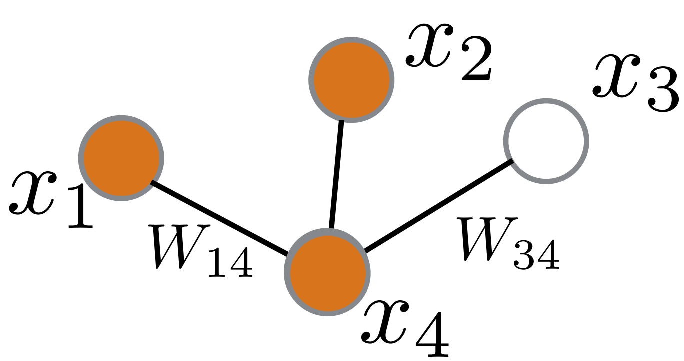
also known as Ising models. Very similar to undirected graphical models

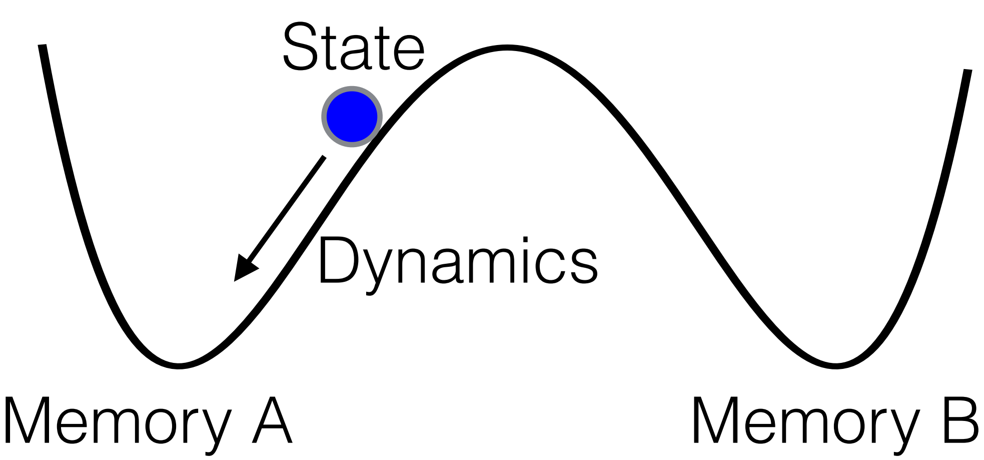
Hopfield 1982, Hinton & Sejnowski 1985

Hopfield networks: canonical models of neural memory



Network dynamics





Input $\sum_i W_{i4} x_i$

$x_4 = 0$ if input < threshold

$x_4 = 1$ if input > threshold

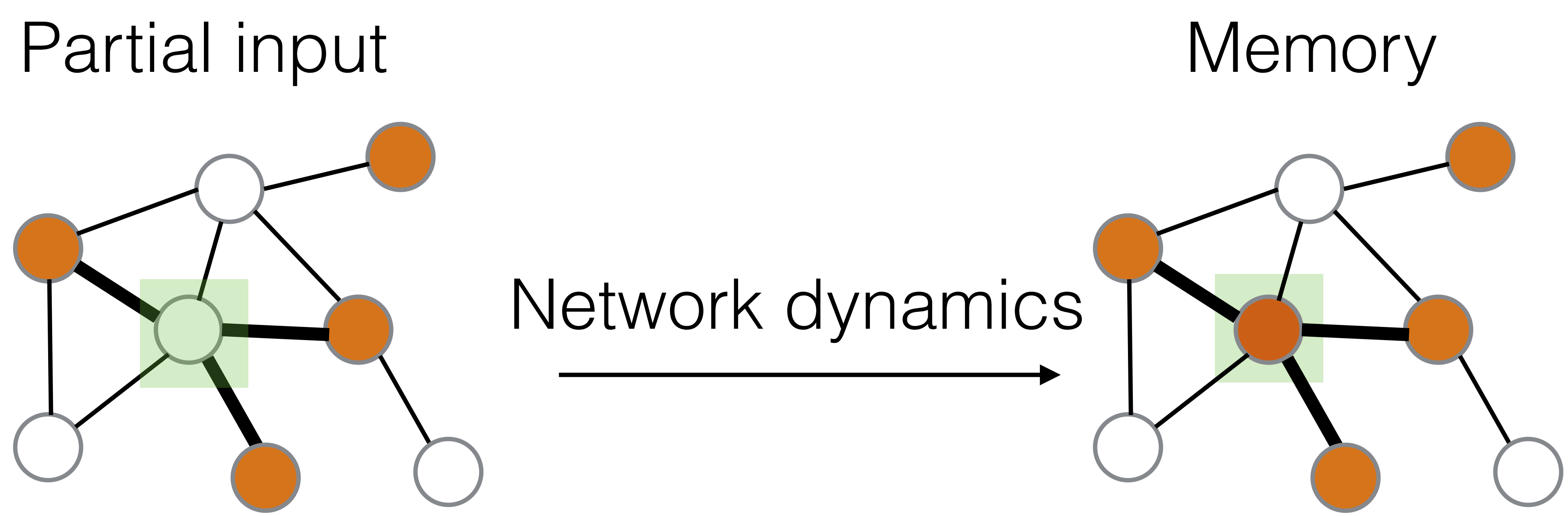
$$E(\boldsymbol{x}|W) = -\frac{1}{2}\boldsymbol{x}^T W \boldsymbol{x}$$

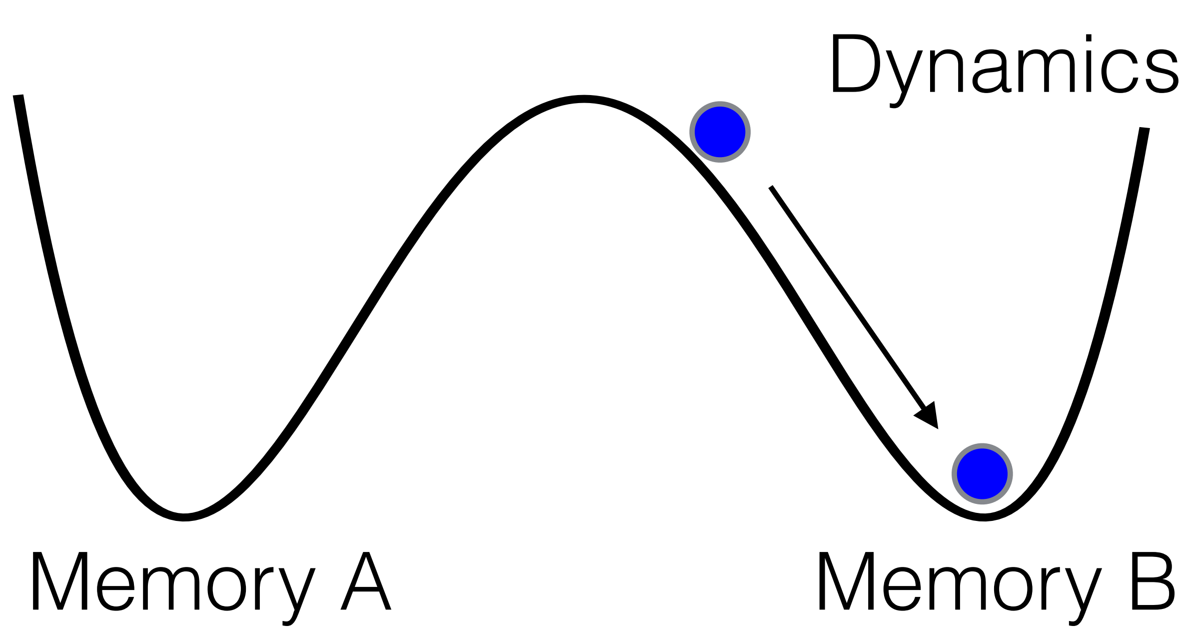
Fixed points of the dynamics can be used to store memories

also known as Ising models. Very similar to undirected graphical models

Hopfield 1982, Hinton & Sejnowski 1985

Stored states recovered from noisy or partial input





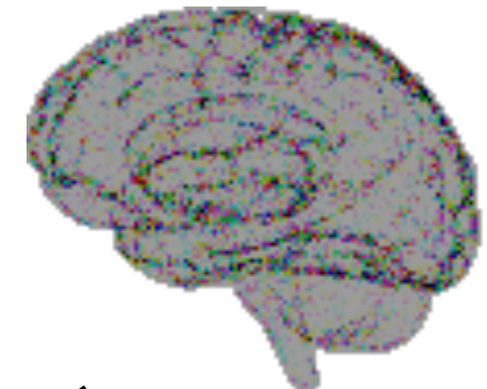
Noise tolerant computation

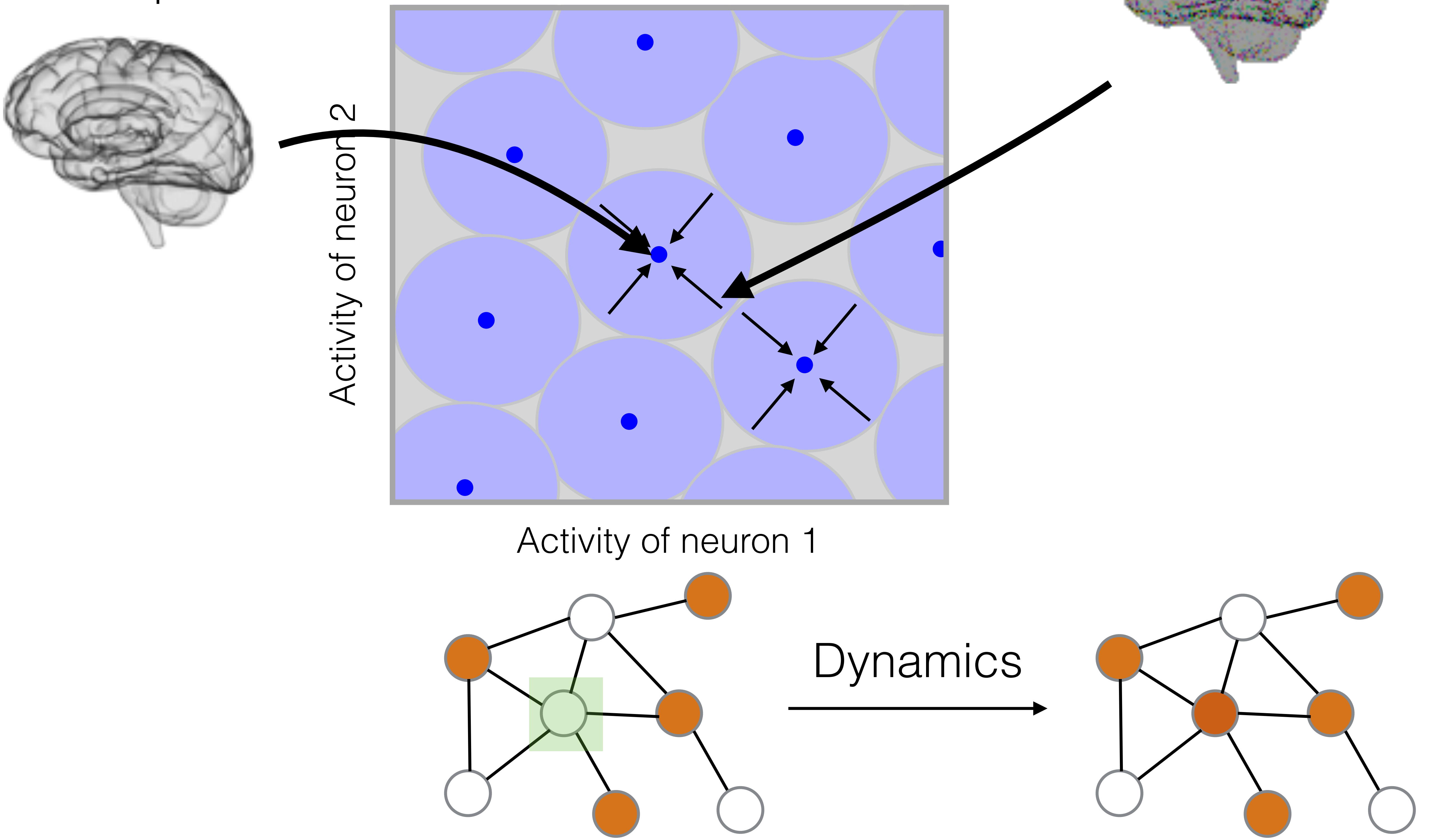
Associative recall

e.g., Fermat's last ________

Autoassociative (Hopfield) memory networks

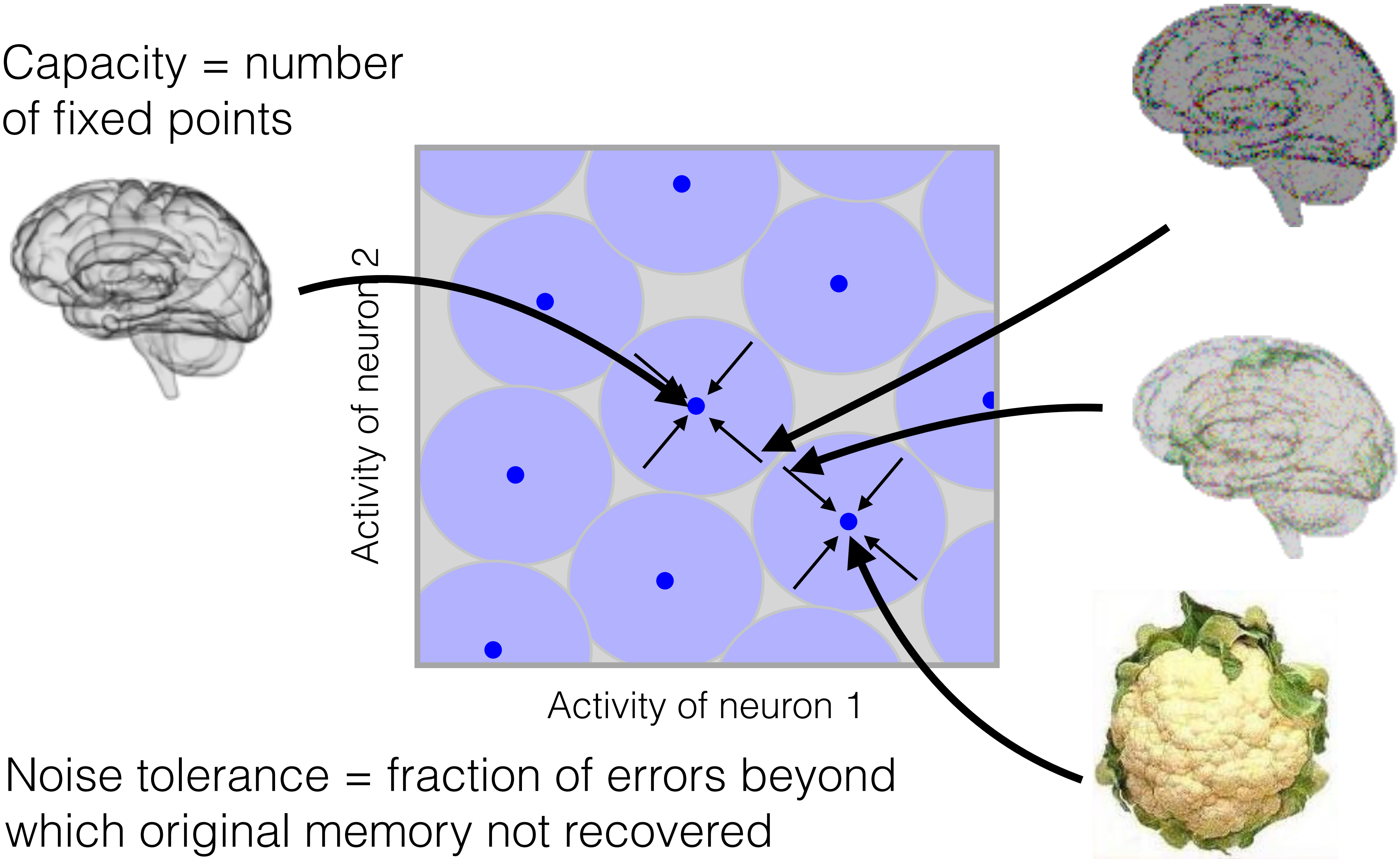
Capacity = number of fixed points





Autoassociative (Hopfield) memory networks

Capacity = number of fixed points

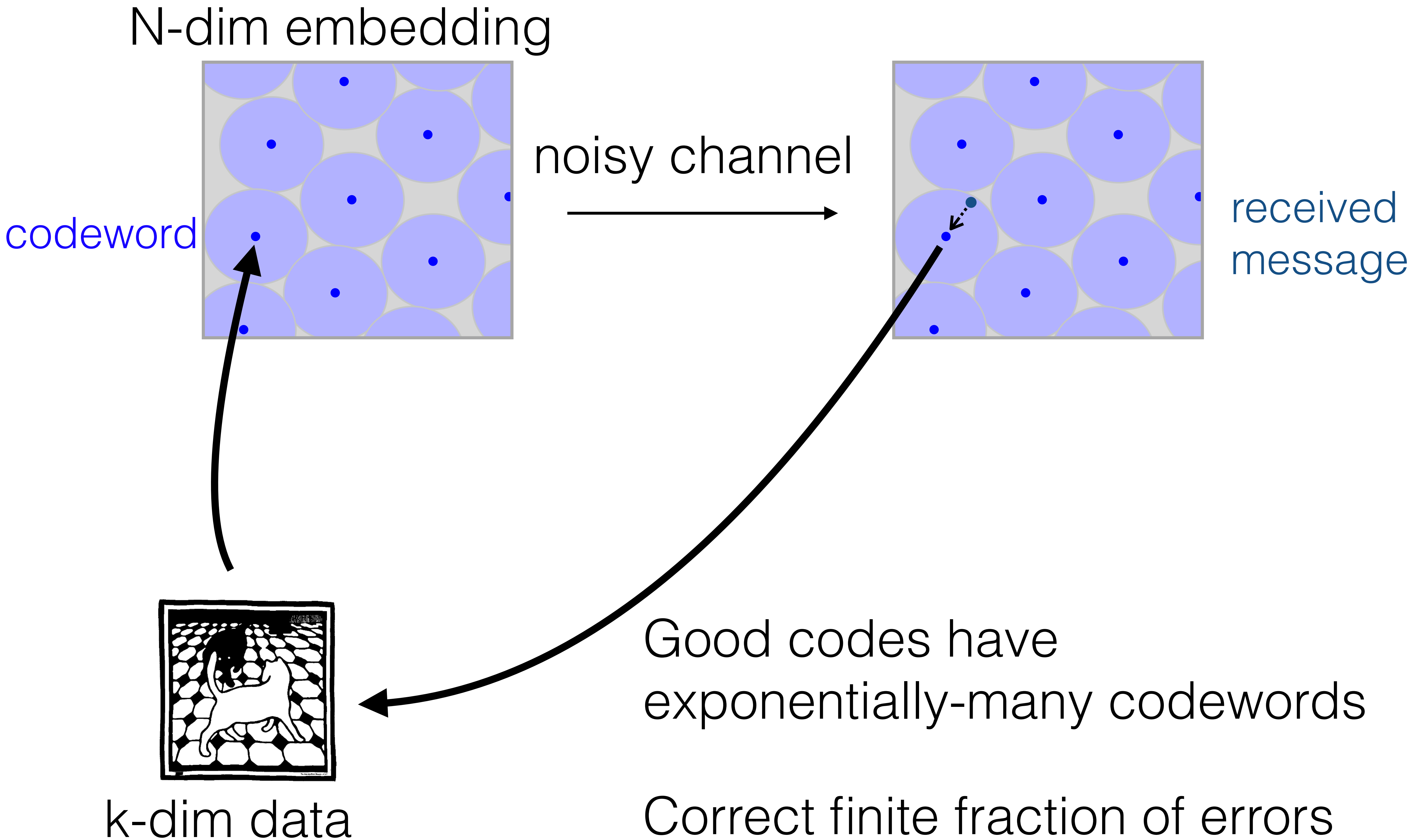


Noise tolerance = fraction of errors beyond which original memory not recovered

Can we combine high capacity and robustness?

Number of stable states	Robustness	
$\sim 2^N$	~ 1	Spin glasses Tanaka 1980; Baldi & Venkatesh 1987; Anderson 1989
$\sim N$	$\sim N$	Random patterns Amit 1985; Abu-Mostafa 1985; McEliece 1987; Gardner 1988; Sompolinsky 1988;
$\sim 2^{\sqrt{N}}$, $\sim 2^{N/\log(N)}$	$\sim N$	Structured patterns Hillar & Tran, 2018; Fiete et al, 2014
$\sim 2^N$	$\sim N$	?

Inspiration from error-correcting codes



Good codes have
exponentially-many codewords

Correct finite fraction of errors

Codewords of a parity check code on N bits can be mapped to the energy minima of Hopfield networks on O(N) variables

ECC	Hopfield network
$x_i \in \{0, 1\}$	$s_i \in \{-1, +1\}$
For all C_i $\sum_{i \in C_i} x_i = 0$	$E(\vec{s}) = -\sum_{C_i} \prod_{i \in C_i} s_i$
	higher-order Hopfield network, we prove that two-layer Hopfield network can do the same

Hopfield networks of N nodes can have exponentially-many well-separated energy minima

Also see Sourlas, 1989

Problem: ECCs are difficult to decode

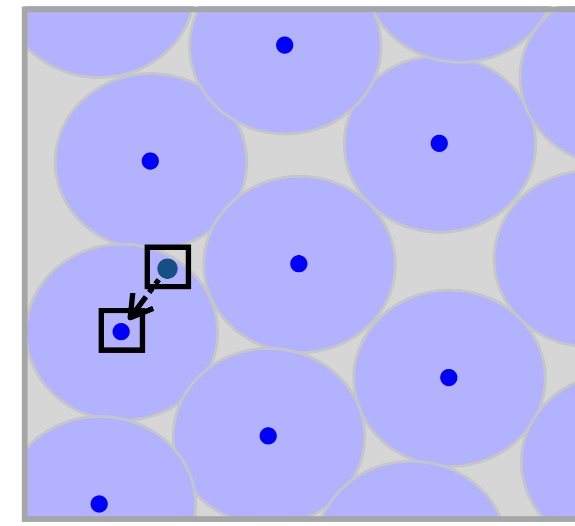
Complex, non-local decoding

Noisy message → Codeword

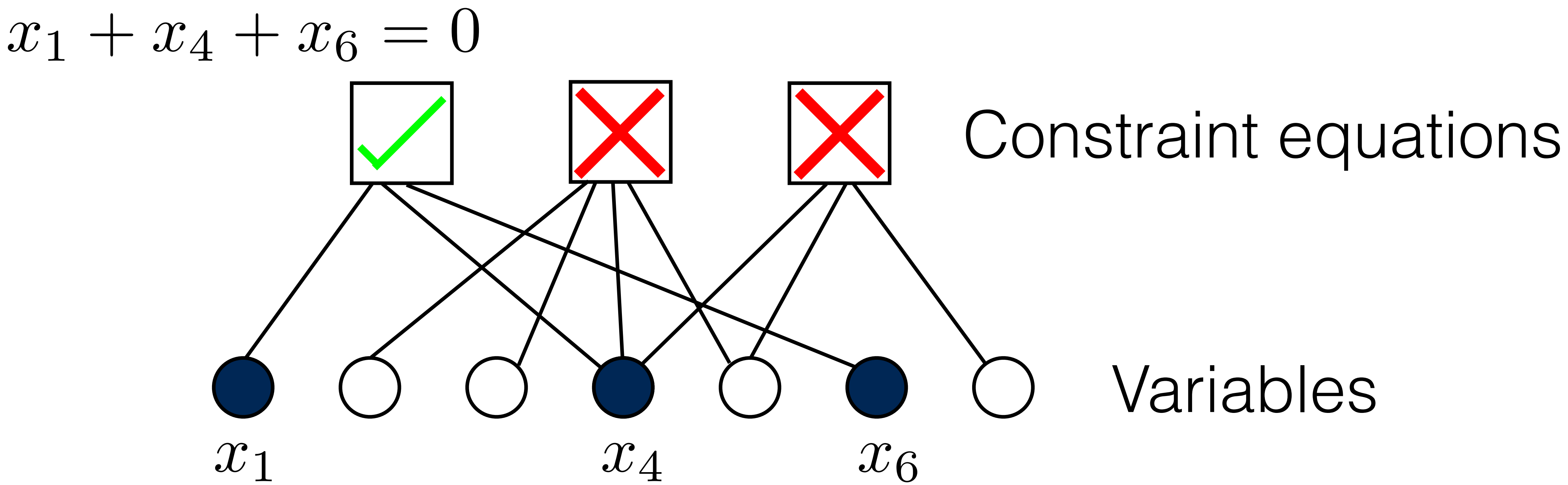
X

Input state → Stable state

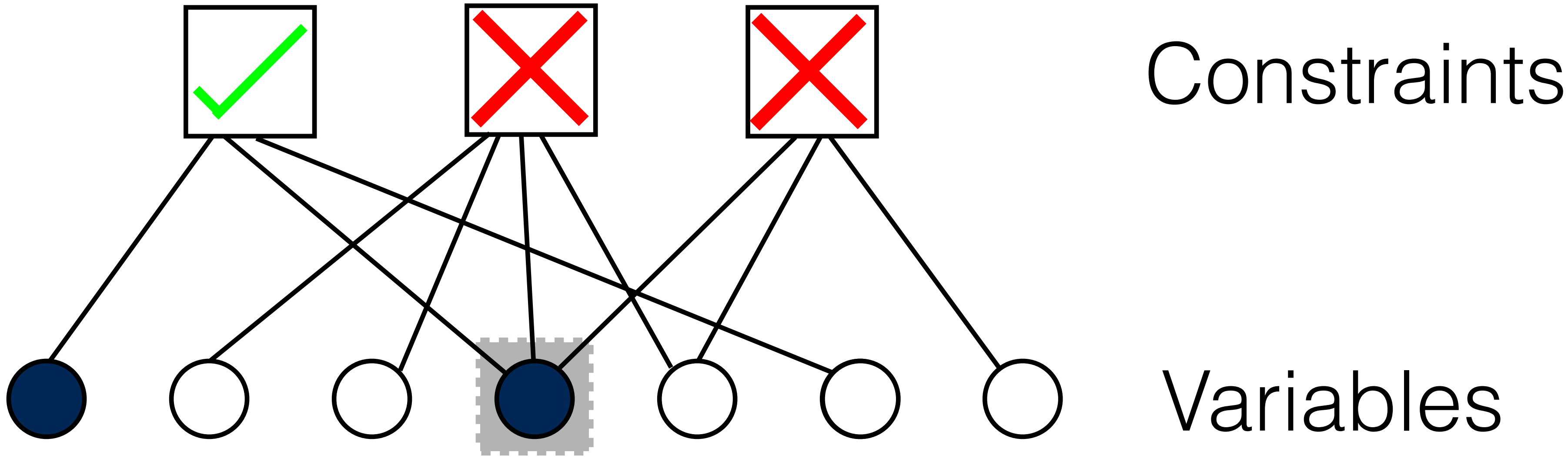
Greedy, local network dynamics



ECCs can be represented by a bipartite graph

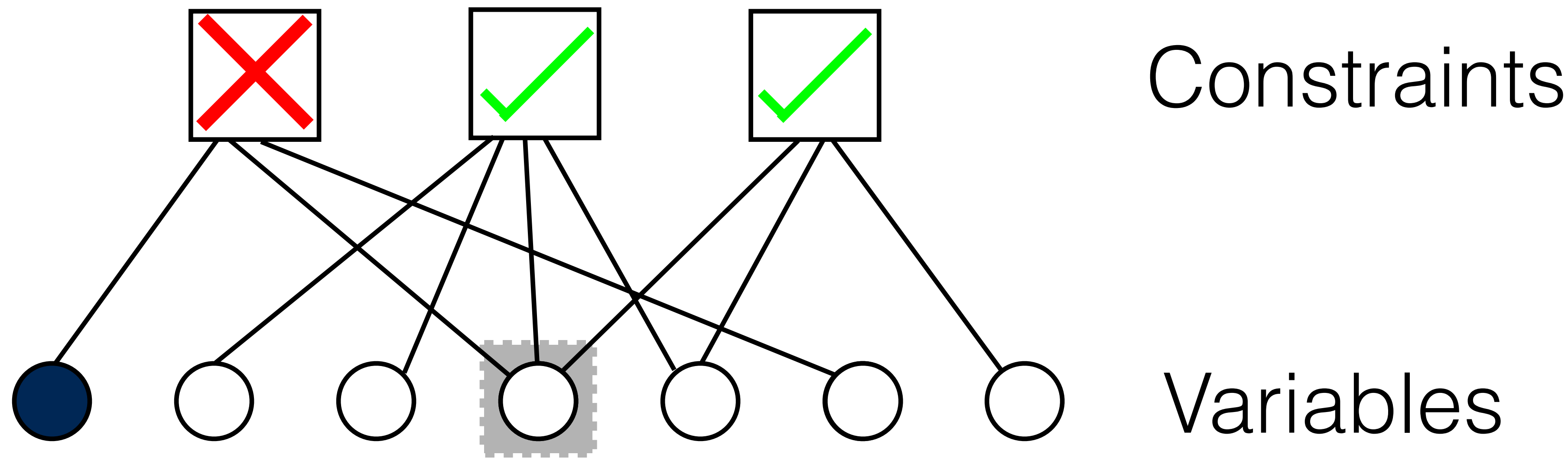


What would a local decoding rule look like?



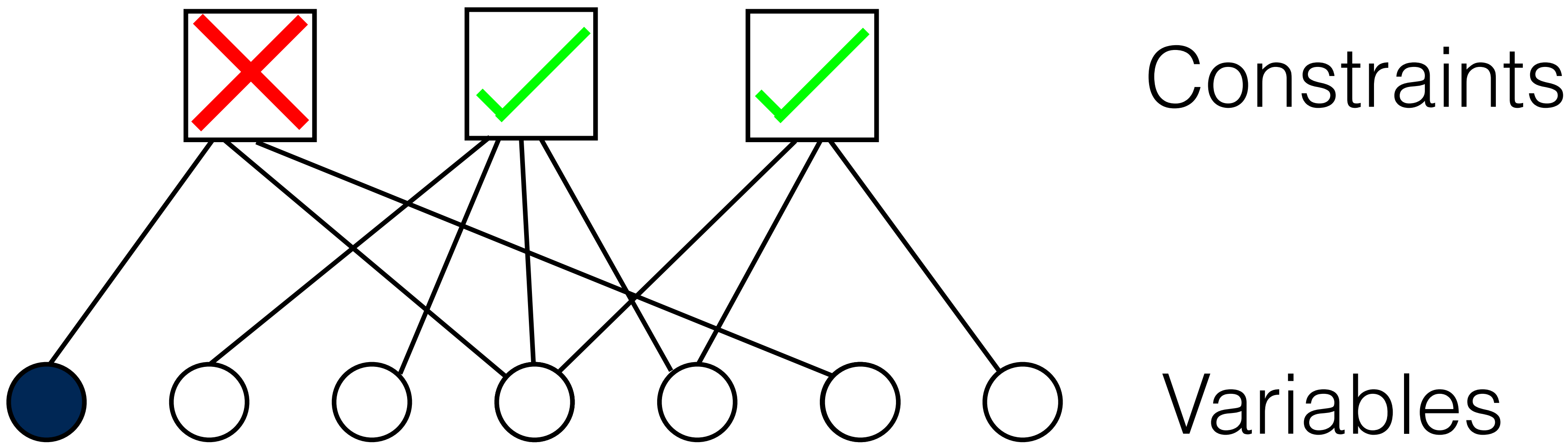
Change state of variable to reduce numbers of violated constraints

What would a local decoding rule look like?



Change state of variable to reduce numbers of violated constraints

What would a local decoding rule look like?



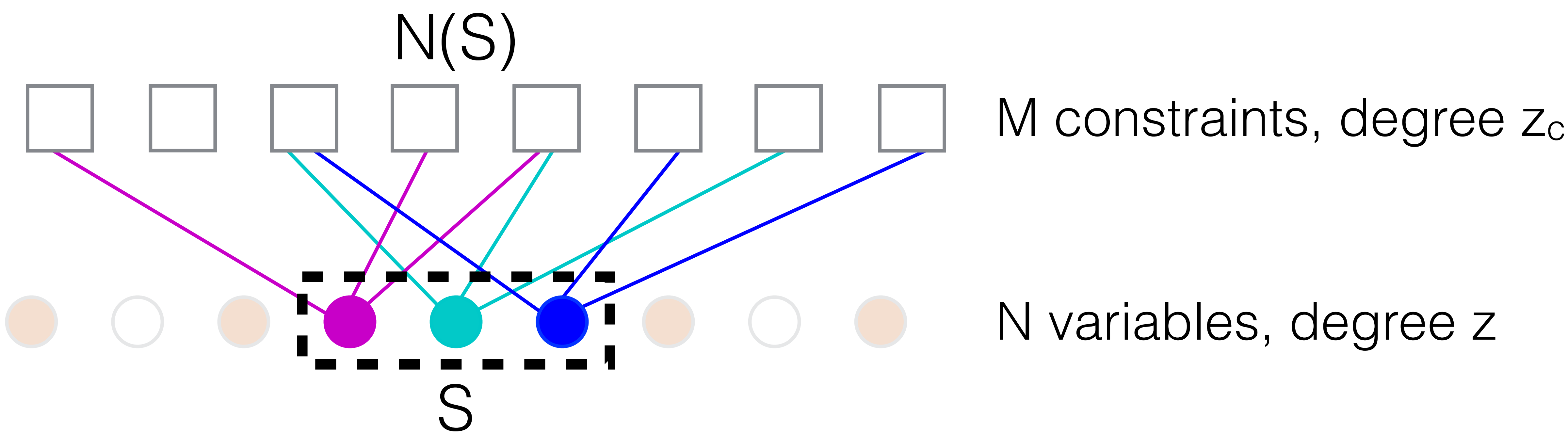
Change state of variable to reduce numbers of violated constraints

Does not work for general codes (we show for Hamming codes fails with probability $\longrightarrow 1$)

Local decoding if graph is a sparse **expander**

Expander codes: Sipser & Spielman, 1996

Bipartite expander graphs



$(\gamma, (1-\epsilon))$ expander: all sets of variables S s.t. $|S| \leq \gamma N$ have neighbors N(s) s.t. $|N(S)| > (1-\epsilon) z |S|$

"small" sets of nodes have "large" numbers of neighbors

"small" sets of nodes do not share many neighbors

Expander graphs can be constructed randomly for large N

Pinsker 1973, Alon & Spencer 2011

Some evidence that Kolmogorov introduced expanders to model the brain!

Gromov & Guth, 2011

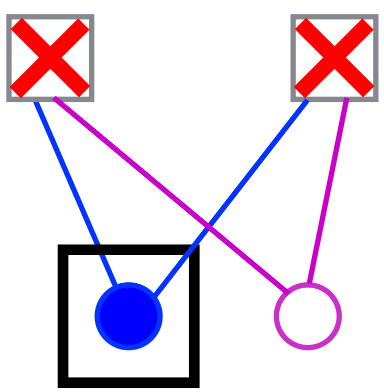
Rich source of distributed algorithms

Why does expansion help with error correction?

Code must identify and correct errors using
weak constraints

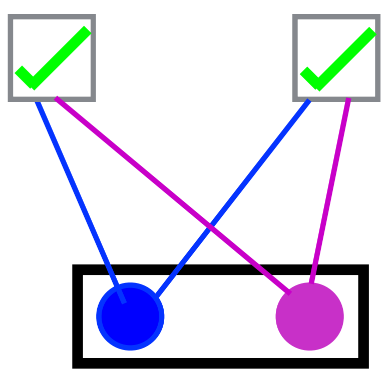
Can fail when variables share constraints in common

One error



Who caused it?

Two errors

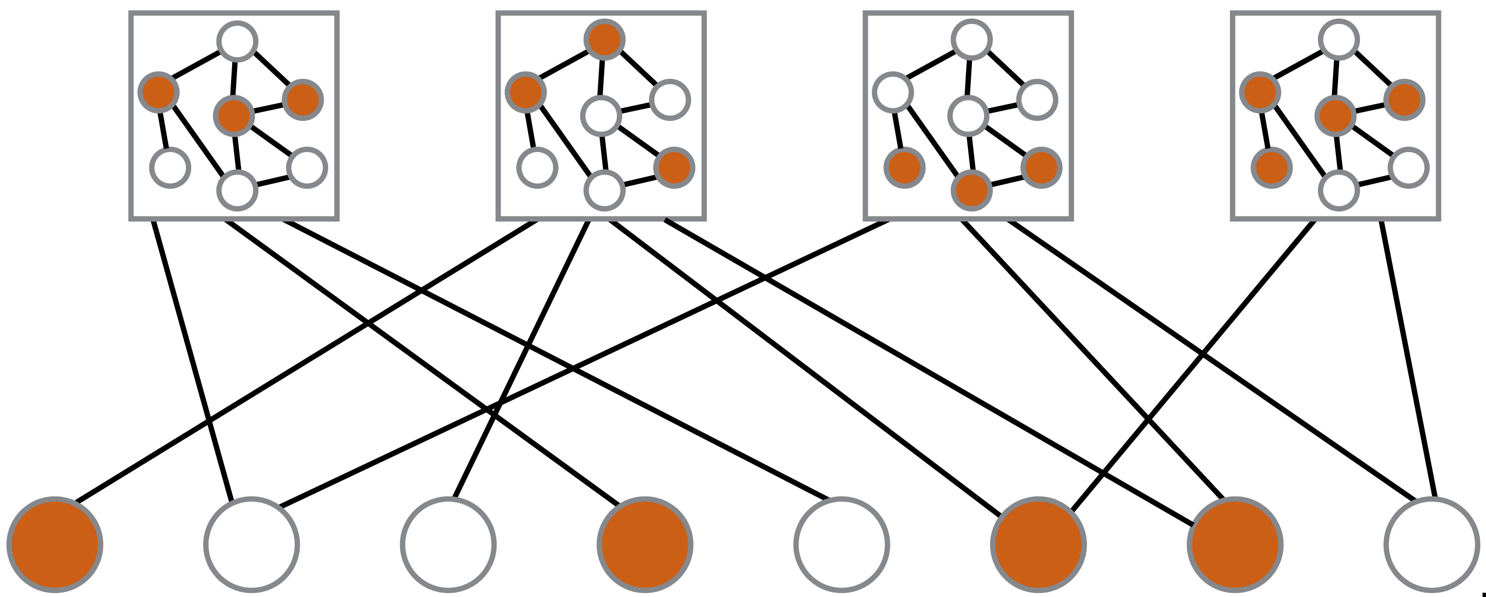


Constraints overwhelmed

Expansion guarantees that this does not happen often

Sipser & Spielman, 1996

Construct Hopfield network where stable states are determined by sparse constraints with expander structure



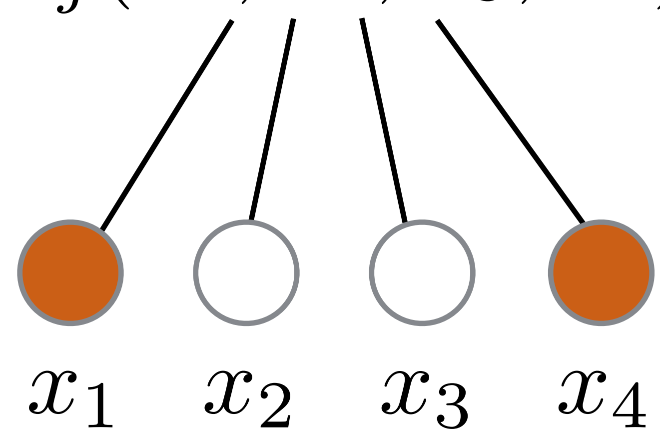
Constraint layer
KN neurons

Sparse,
decorrelated
connections

Representation layer
N neurons

Each constraint module implements a weak constraint

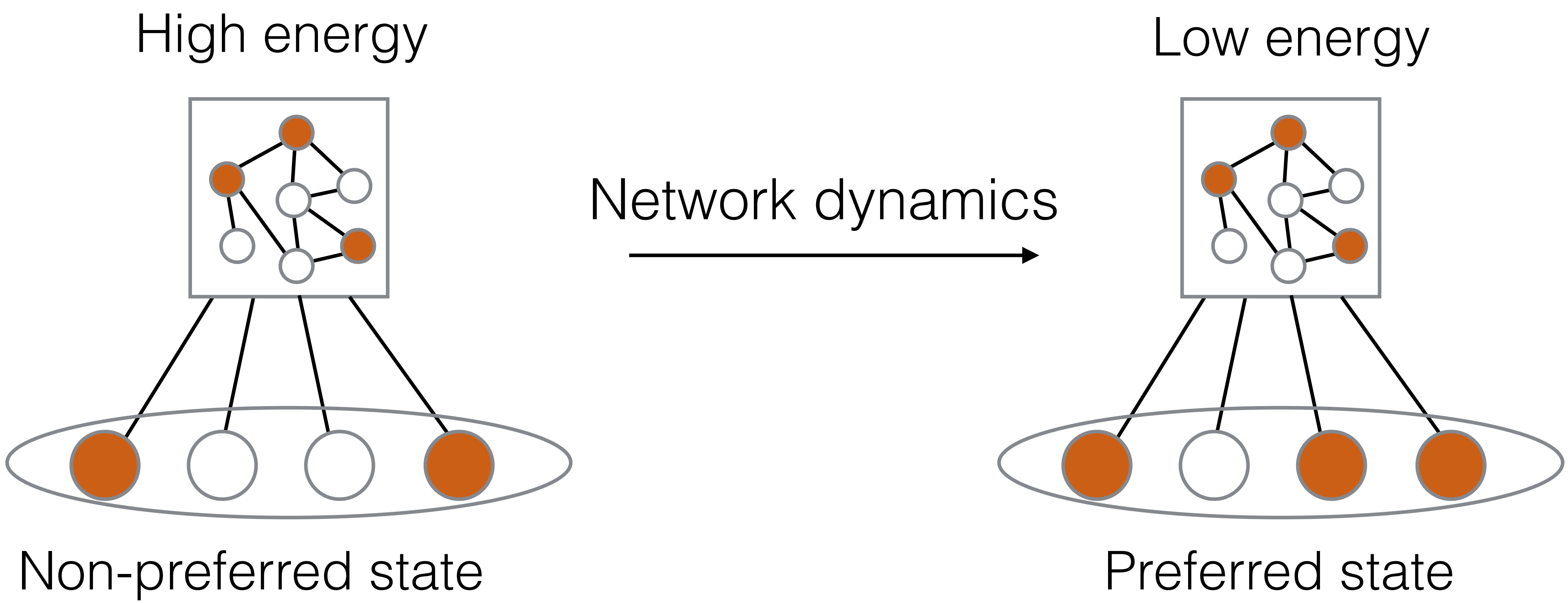
$$E_j(x_1, x_2, x_3, x_4)$$



E_j is low energy for some subset of configurations

Each constraint module is a Hopfield network

Tries to impose a set of preferred states on its inputs



Competitive (winner-takes-all) dynamics in each module

Each constraint is weak: constrains a **small** fraction of inputs to some **large** set of acceptable states

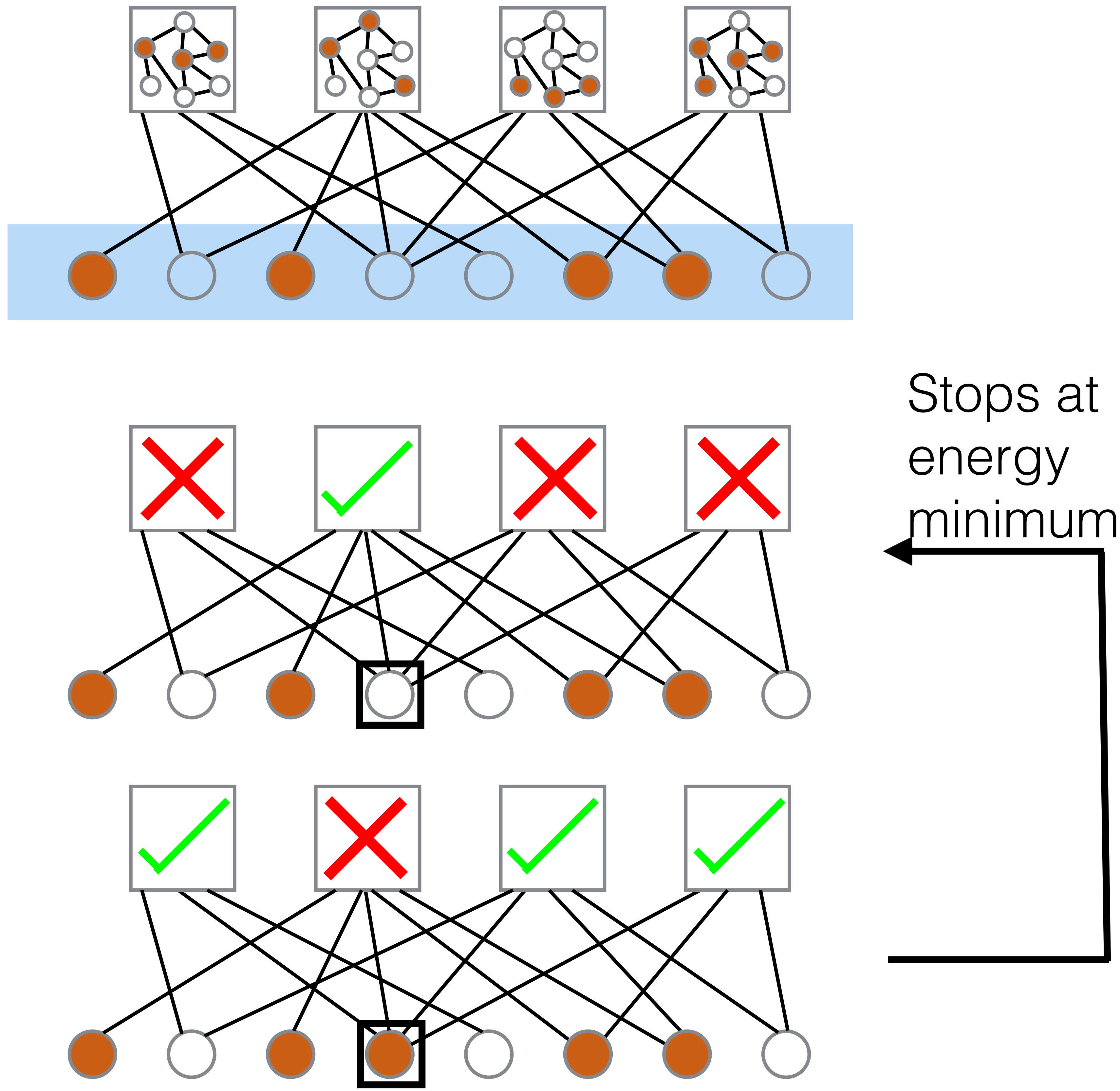
Network combines them near-optimally

Network operation

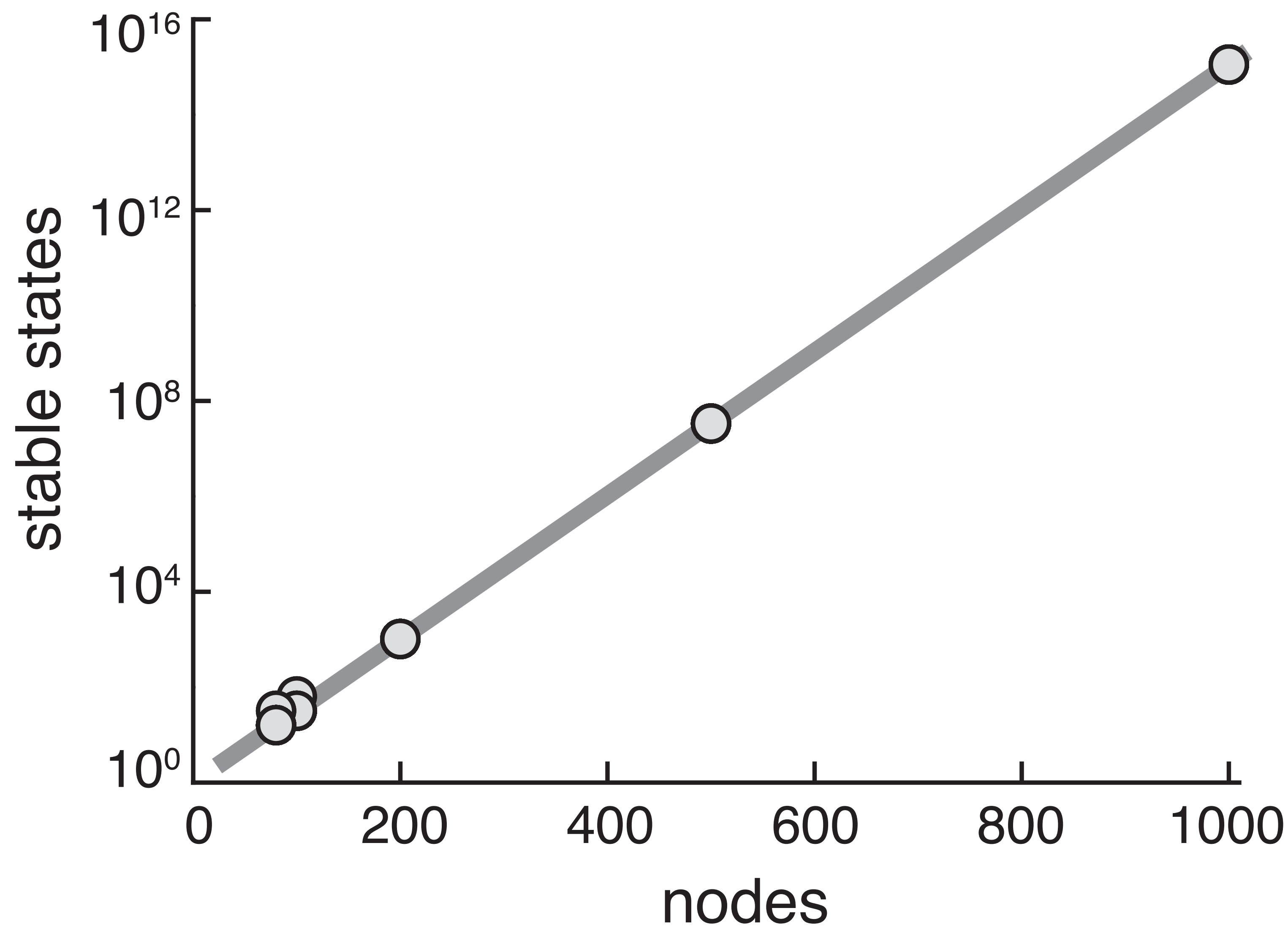
1. Input comes in

2. Constraint networks are either high or low energy

3. Neurons connected to more high than low energy constraints update their state



Exponential capacity



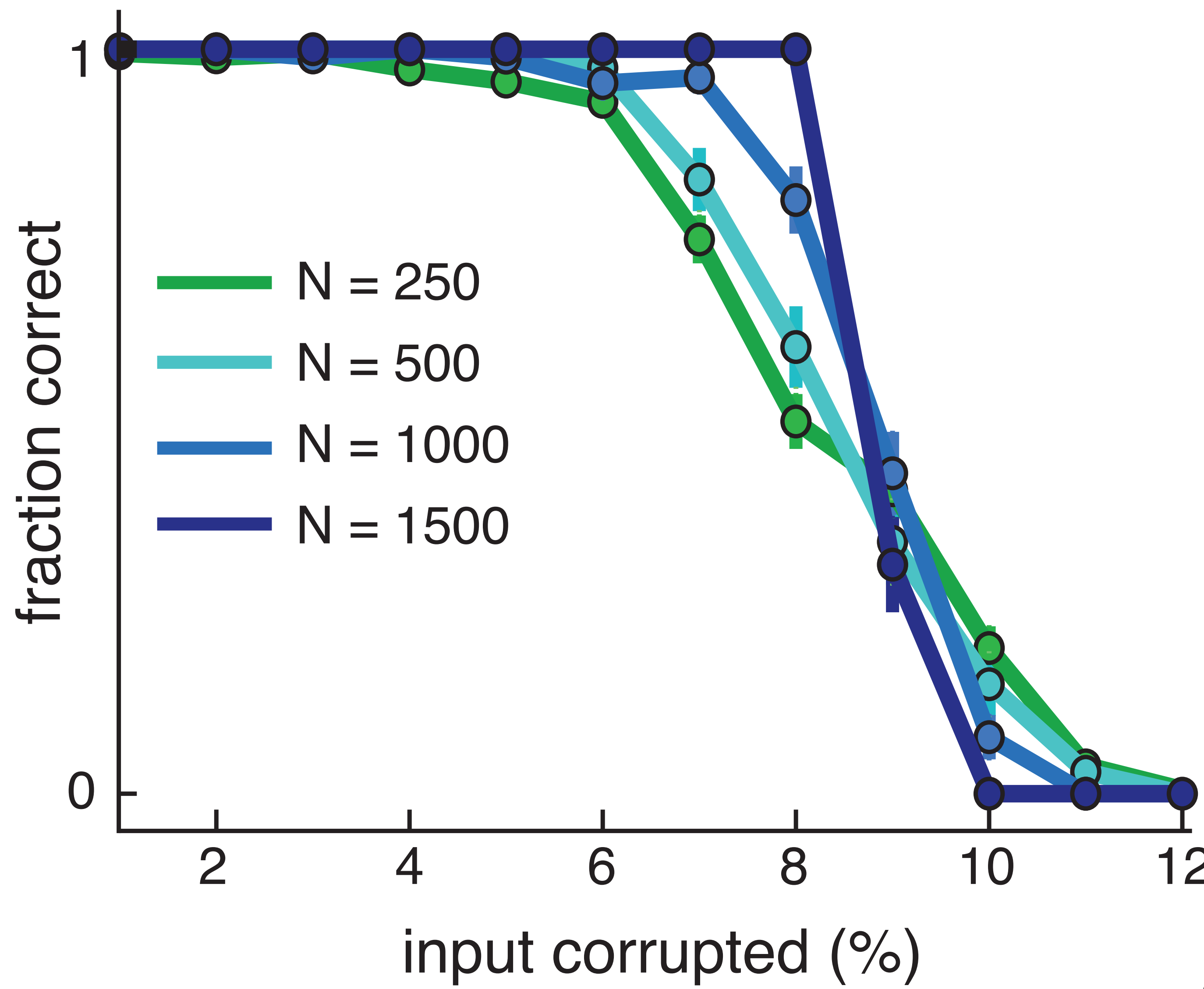
Number of steady-states > $2^{\alpha N}$

$$\alpha = \left(1 - \rho \frac{z_I}{z_C}\right) / \left(1 + 2^{z_C - 1} \frac{z_I}{z_C}\right)$$

Constraint selectivity

Degree of variables and constraints

Robust to finite error probability



$\beta = \gamma(1 - 2\epsilon)N$

Number of errors corrected > βN

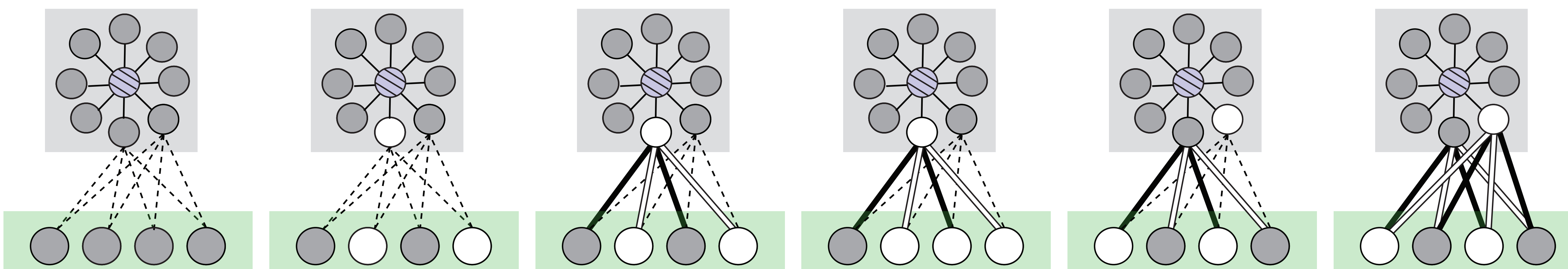
Network is $(\gamma, (1-\epsilon))$ expander: $|S| \leq \gamma N \implies |N(S)| > (1-\epsilon)z|S|$

Learning / self-organization

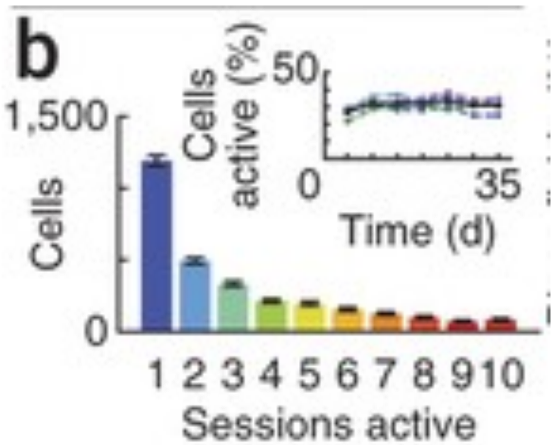
Start with random connections between input neurons and constraint nodes

Activate neurons by random feedforward inputs + recurrent inhibition

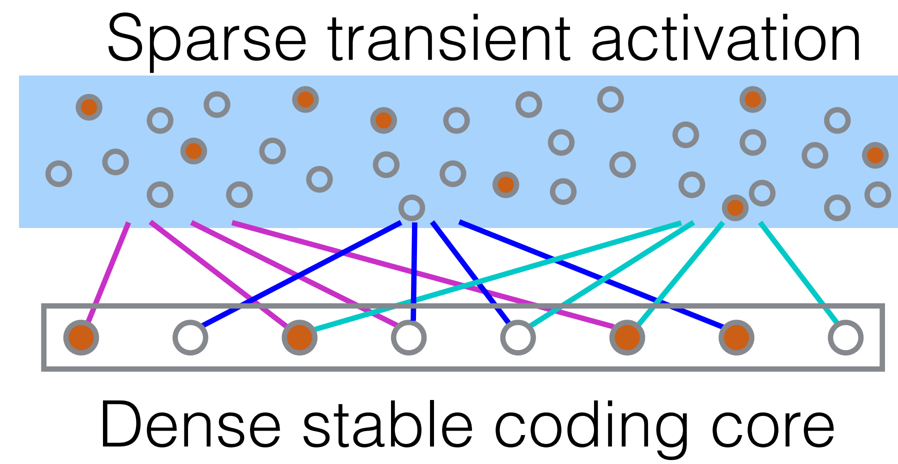
Hebbian one-shot learning



What does this predict for neural coding?

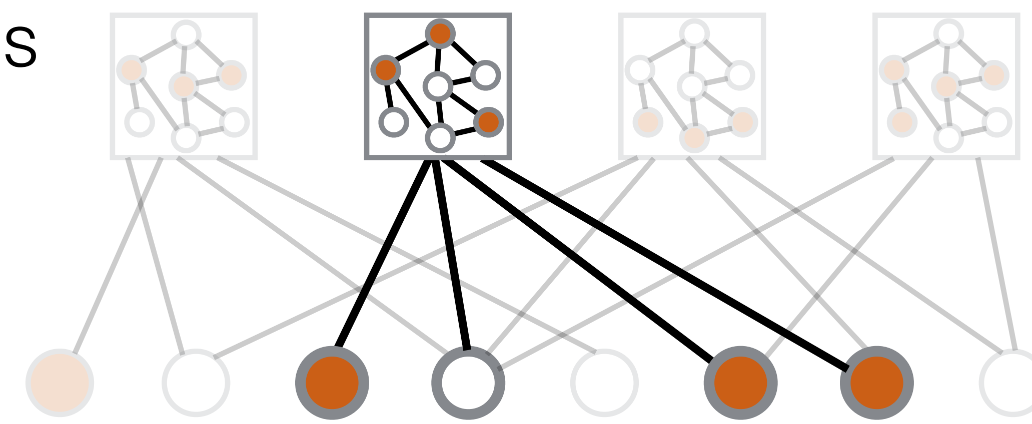


Ziv 2013



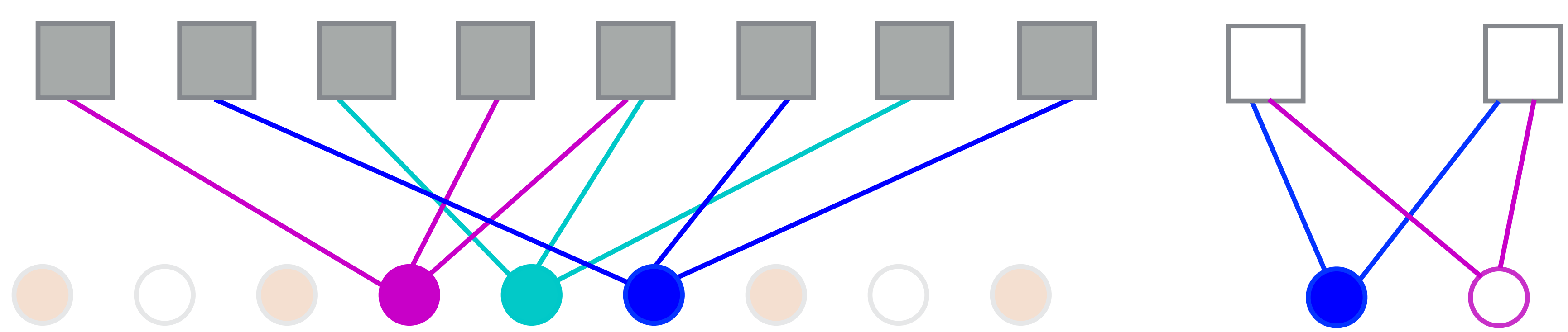
Exponential coding on a small fraction of network vastly outperforms classical strategies

Higher-order correlations for high capacity



Probe with optical imaging

Anatomical clustering patterns and motif distributions



Conclusions

ECCs can be mapped to Hopfield networks

Expander codes allow new Hopfield network construction with exponentially-many robust stable states

Two-layer network with sparse constraint structure

Most neurons are only rarely active, representations are pairwise decorrelated but have structure in higher-order moments, and connectivity between layers is random (up to constraint modules)