From Distributed Computing to Natural Algorithms and Beyond

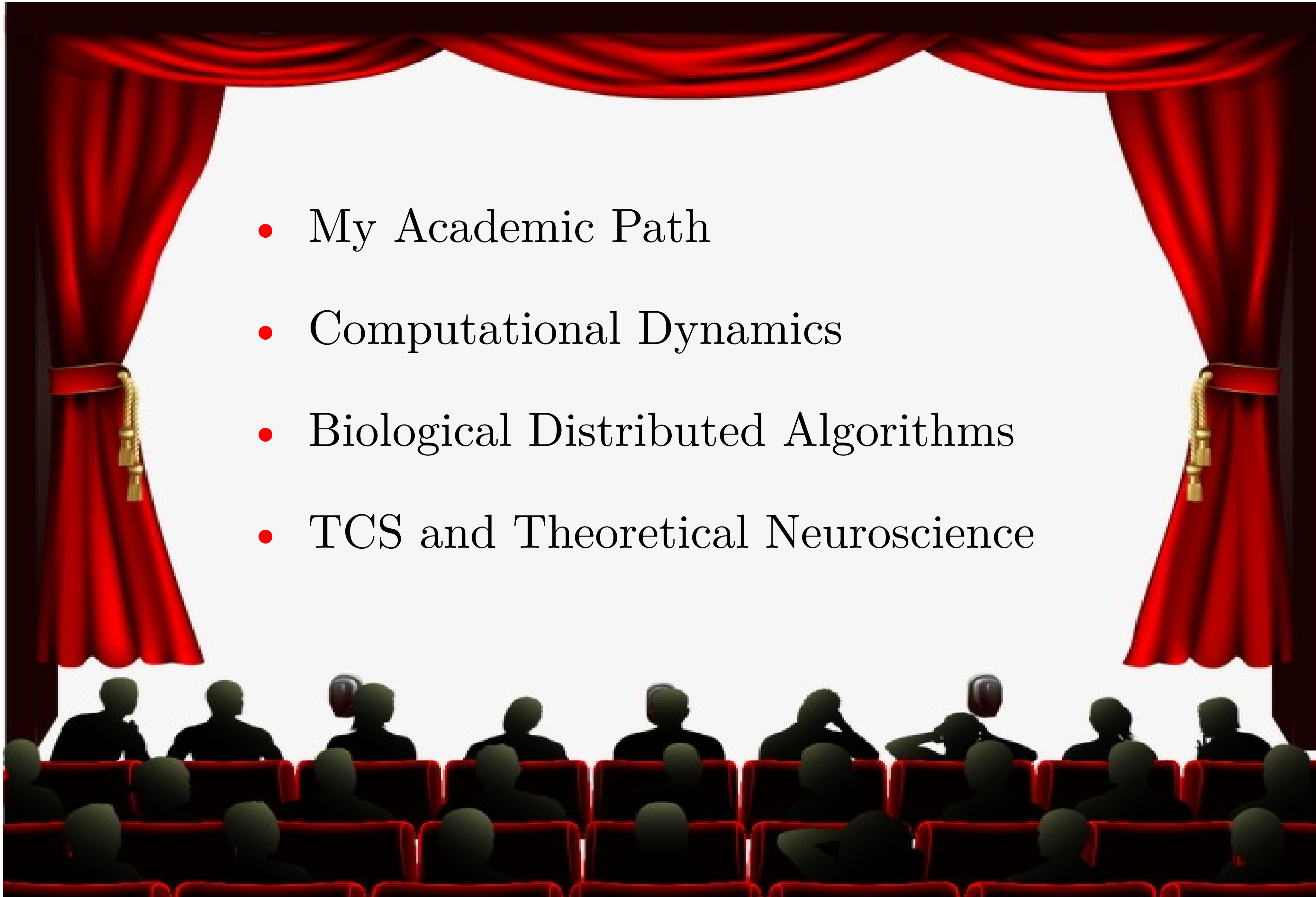
Emanuele Natale

Joint work with L. Becchetti, L. Boczkowki, V. Bonifaci, A. Clementi, L. Gualà, A. Korman, O. Feinerman, P. Manurangsi, F. Pasquale, P. Raghavendra, G. Scornavacca, R. Silvestri, L. Trevisan

ICTCS 2019

Como, 08 September 2019

Outline



- My Academic Path
- Computational Dynamics
- Biological Distributed Algorithms
- TCS and Theoretical Neuroscience

Pre-CNRS Algorithmic Biography

- 2016 - PhD at Sapienza University, in **Theory of Distributed Computing**

SAPIENZA
UNIVERSITÀ DI ROMA

Pre-CNRS Algorithmic Biography

- 2016 - PhD at Sapienza University, in **Theory of Distributed Computing**

- 2016-now - PostDoc at
  Max Planck Institute for Informatics
  D1 - Algorithms & Complexity

Pre-CNRS Algorithmic Biography

- 2016 - PhD at Sapienza University, in **Theory of Distributed Computing**

SAPIENZA
UNIVERSITÀ DI ROMA

- 2016-now - PostDoc at Max Planck Institute for Informatics D1 - Algorithms & Complexity

max planck institut
informatik

- 2016 & 2018 - **Fellow** of Simons Institute for the Theory of Computing

SIMONS INSTITUTE
for the Theory of Computing

Berkeley
UNIVERSITY OF CALIFORNIA

Part I

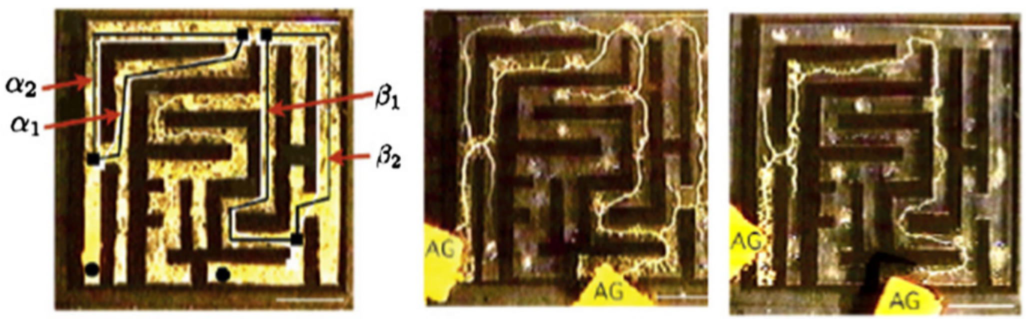
Computational Dynamics

Natural Algorithms

How do flocks of birds synchronize their flight? [Chazelle '09]

Natural Algorithms

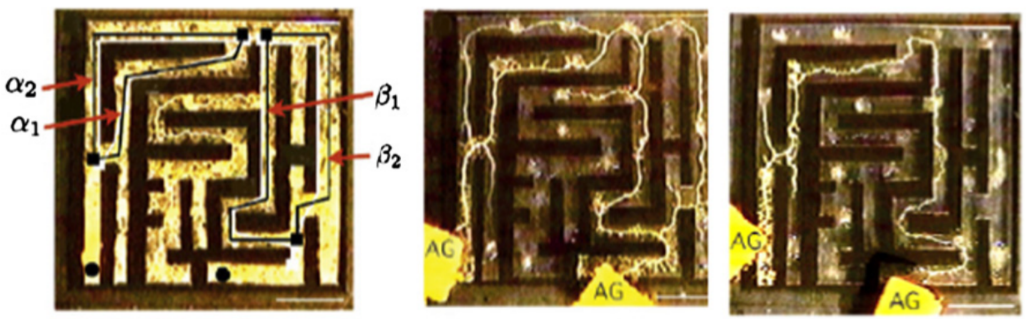
How do flocks of birds synchronize their flight? [Chazelle '09]



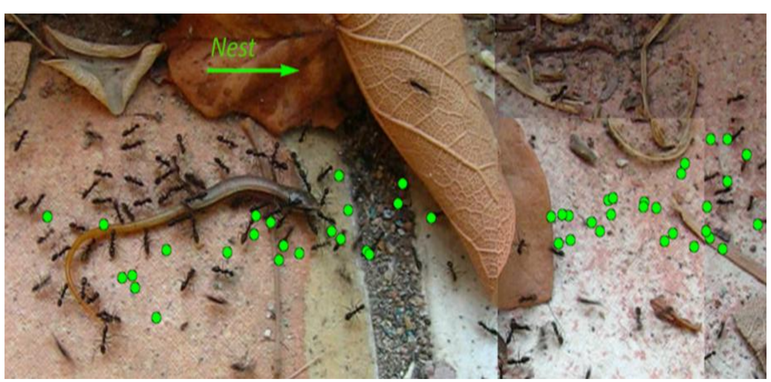
How does Physarum polycephalum finds shortest paths? [Mehlhorn et al. 2012-...]

Natural Algorithms

How do flocks of birds synchronize their flight? [Chazelle '09]

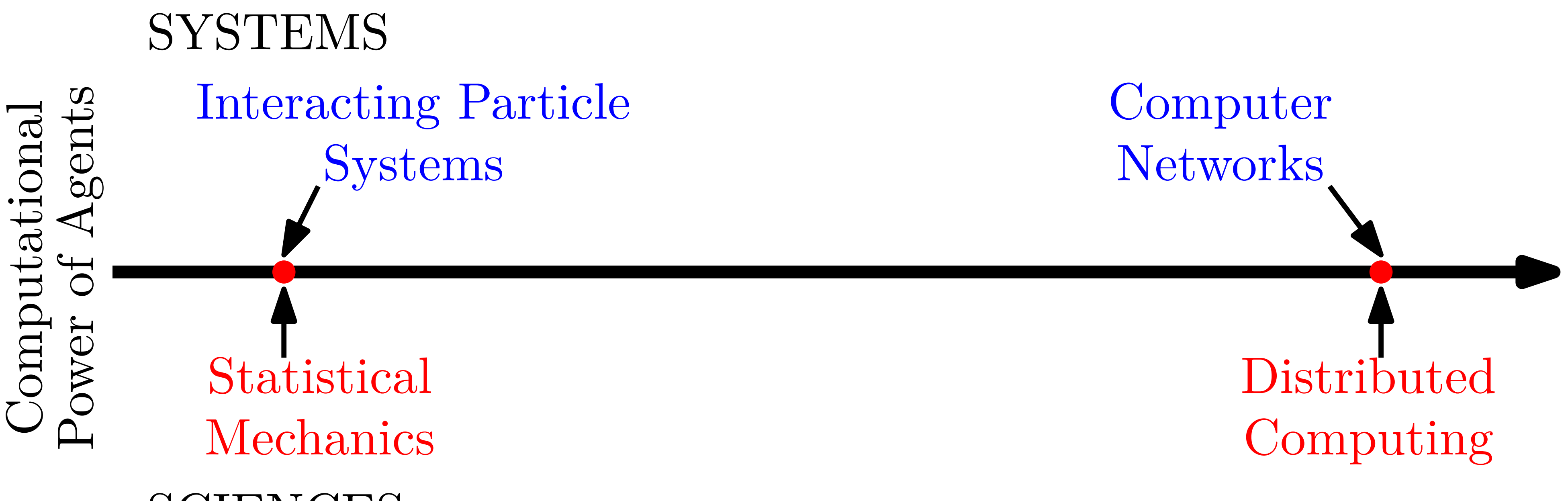


How does Physarum polycephalum finds shortest paths? [Mehlhorn et al. 2012-...]



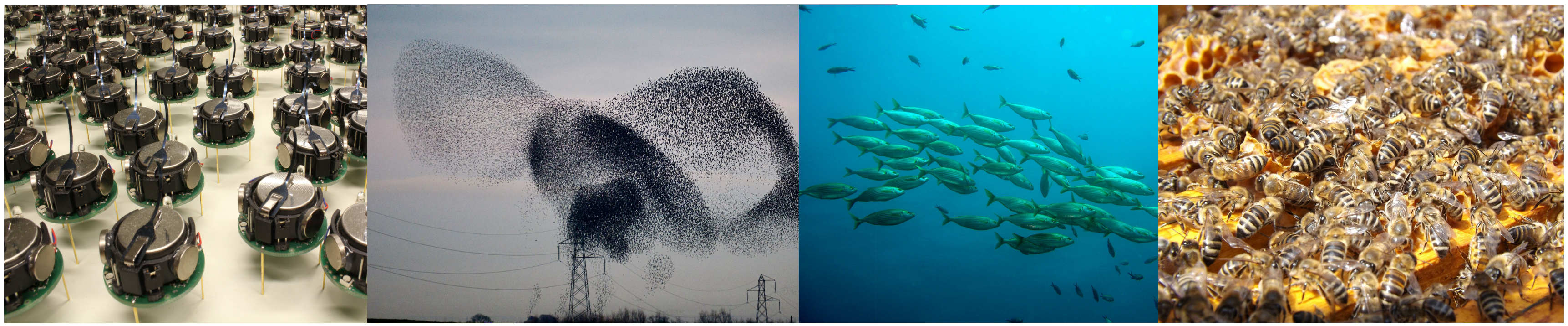
How ants perform collective navigattion? How do they decide where to relocate their nest?

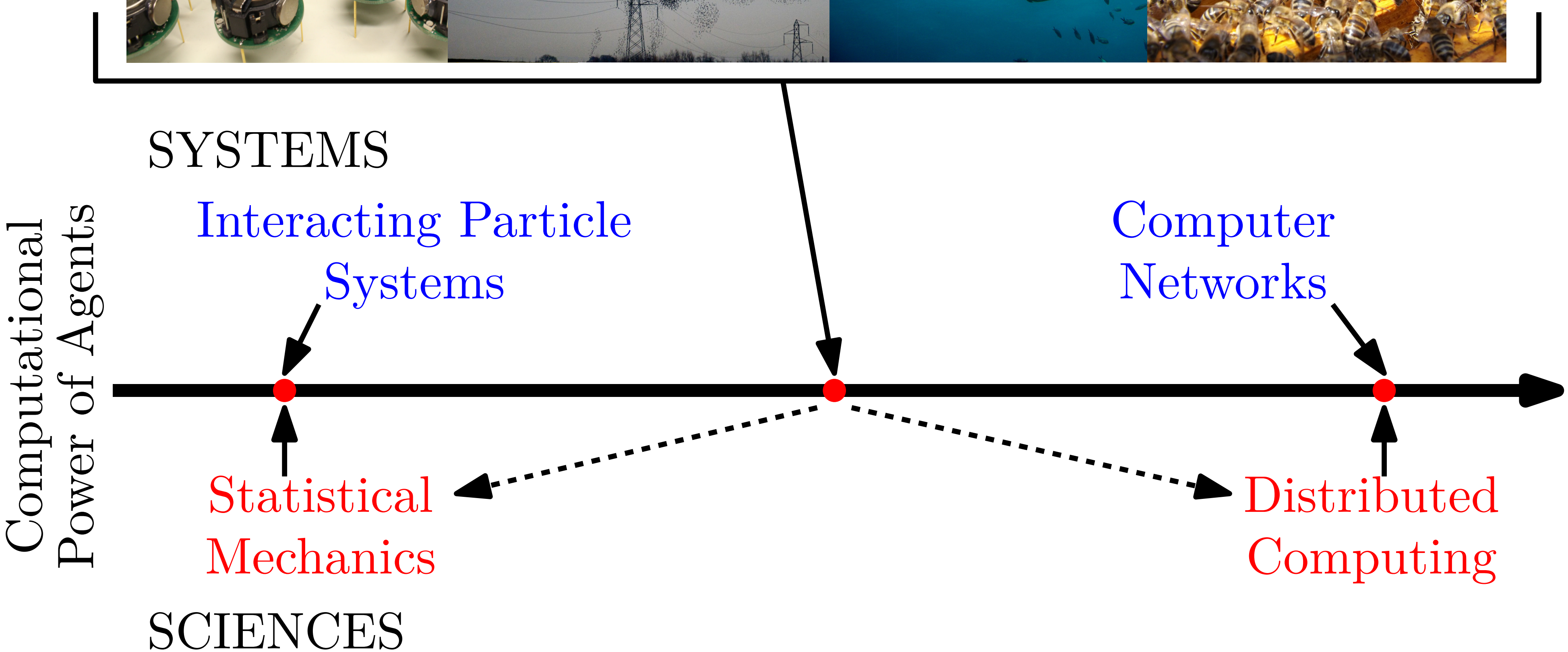
How can *Simple Stochastic* Systems *Compute*?



How can *Simple Stochastic* Systems *Compute*?

A **computational lens** on how
global behavior emerges from
simple stochastic interactions among individuals

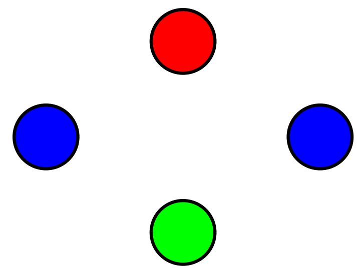




Computational **Dynamics**

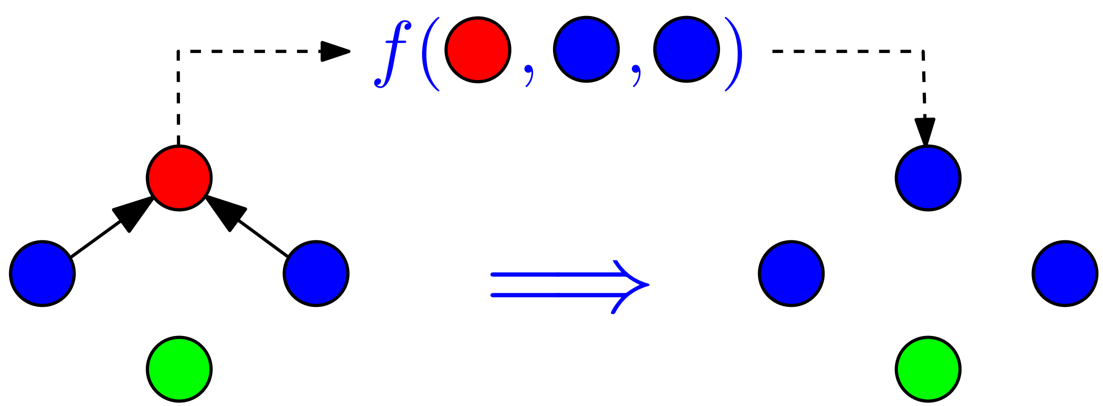
Anonymous agents

- small set of possible states
- *simple* update function f



At each step:

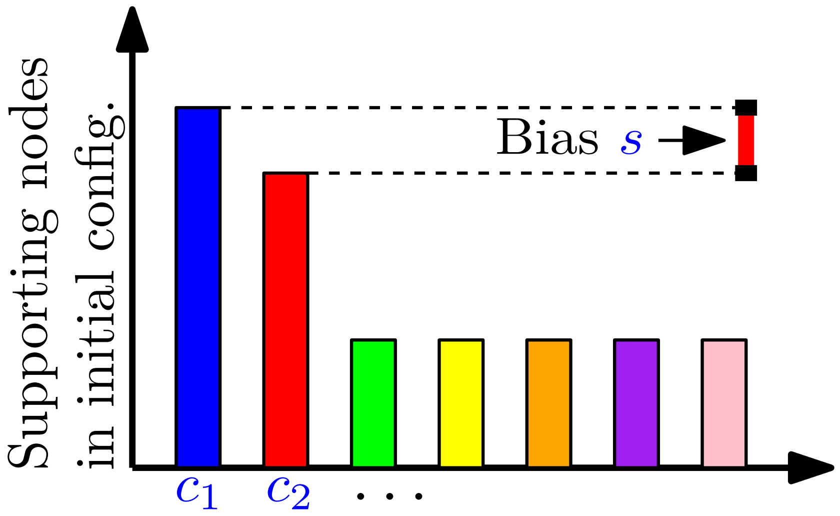
Update depends on states of random subset of agents



Dynamics for Plurality Consensus I

Plurality Consensus.

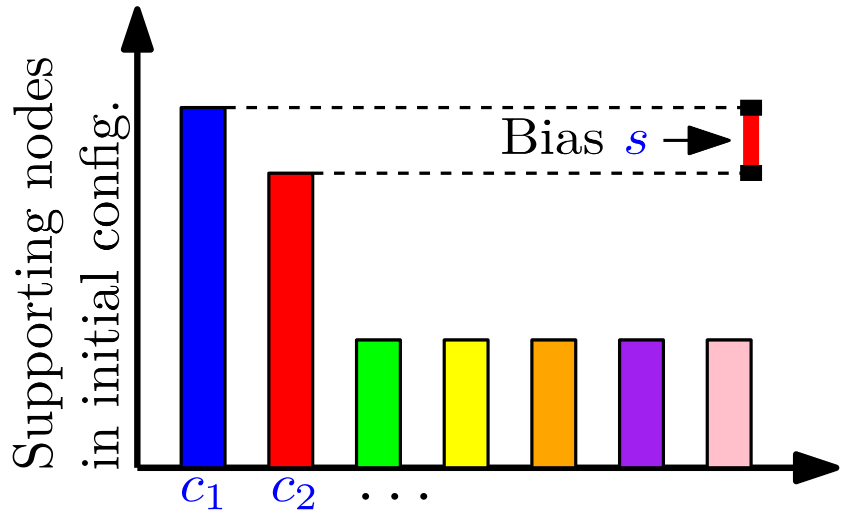
- Each agent initially has a value in $\{1, ..., k\}$.
- $\Omega(\sqrt{kn \log n})$ initial **bias** (majority − 2nd-majority color).
- Each agent eventually has the most frequent initial value.



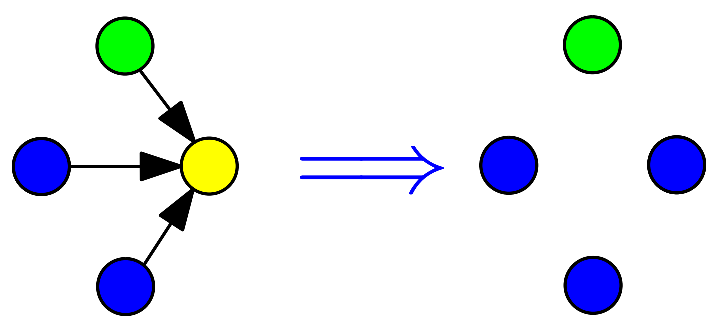
Dynamics for Plurality Consensus I

Plurality Consensus.

- Each agent initially has a value in $\{1, ..., k\}$.
- $\Omega(\sqrt{kn \log n})$ initial **bias** (majority − 2nd-majority color).
- Each agent eventually has the most frequent initial value.



3-Majority Dynamics.
At each round, each agent samples 3 agents and adopts the majority color.

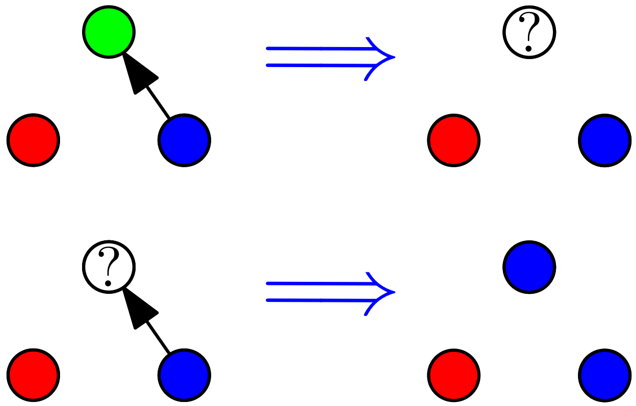


Theorem.
3-Majority Dynamics converges to plurality in $\mathcal{O}(k \log n)$ rounds

Dynamics for Plurality Consensus II

Undecided-State Dynamics.
Each agent u samples an agent v:

- *If v has a different color, u becomes **undecided**.*
- *If undecided, u copies the color of v.*

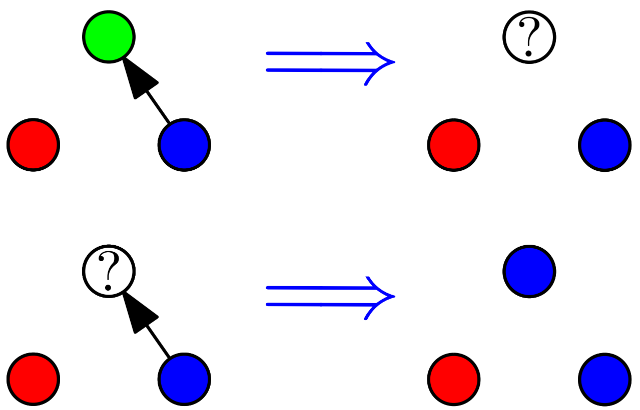


Dynamics for Plurality Consensus II

Undecided-State Dynamics.

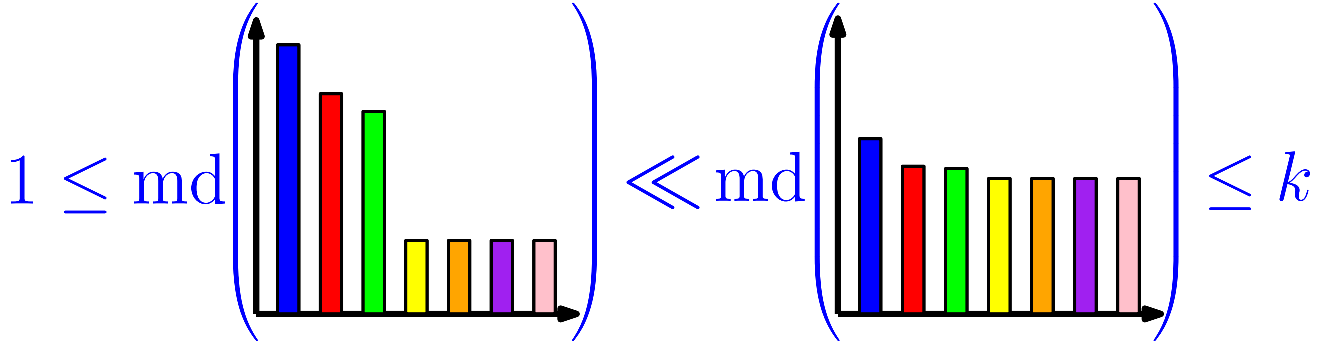
Each agent u samples an agent v:

- *If v has a different color, u becomes **undecided**.*
- *If undecided, u copies the color of v.*



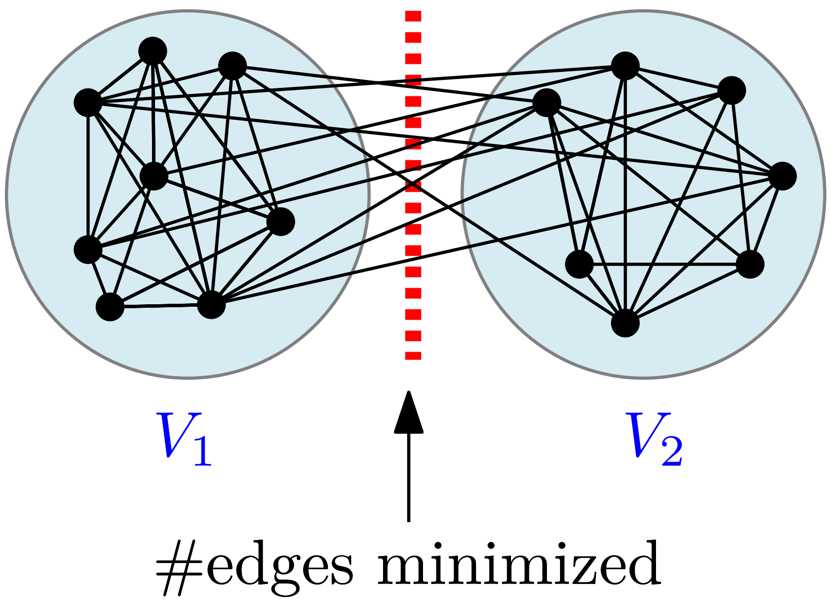
Theorem (Monochromatic Distance).

Undecided-State Dynamics converges to plurality within $\tilde{\Theta}(\text{md}(\text{initial configuration}))$ rounds with high probability.



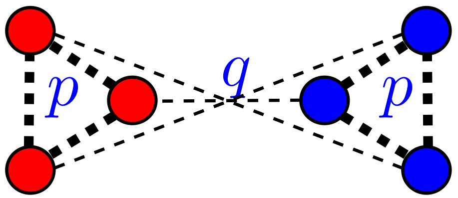
Clustering

Minimum Bisection Problem.
Find balanced bipartition $|V_1| = |V_2|$ that minimizes cut.



[Garey et al. '76]: Minimum bisection problem is NP-Complete!

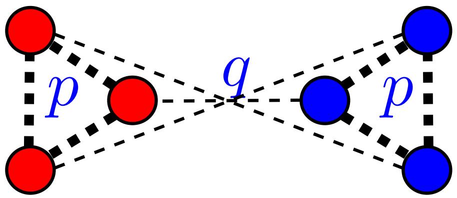
Stochastic Block Model (SBM)

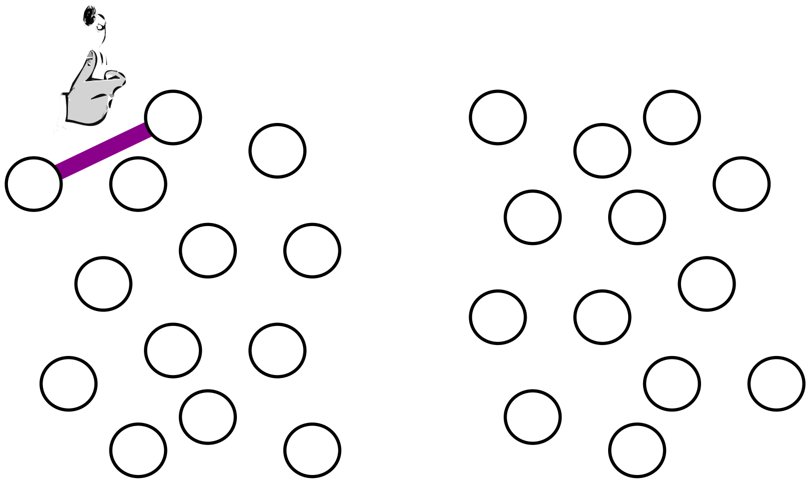


- "Communities" V_1, V_2, with $|V_1| = |V_2|$.
- include each edge with probability
  - p if edge inside V_1 or V_2,
  - q if edge between V_1 and V_2.

Stochastic Block Model (SBM)

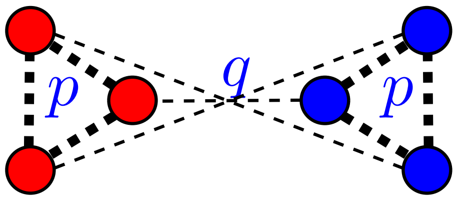
- "Communities" V_1, V_2, with $|V_1| = |V_2|$.
- include each edge with probability
  - p if edge inside V_1 or V_2,
  - q if edge between V_1 and V_2.

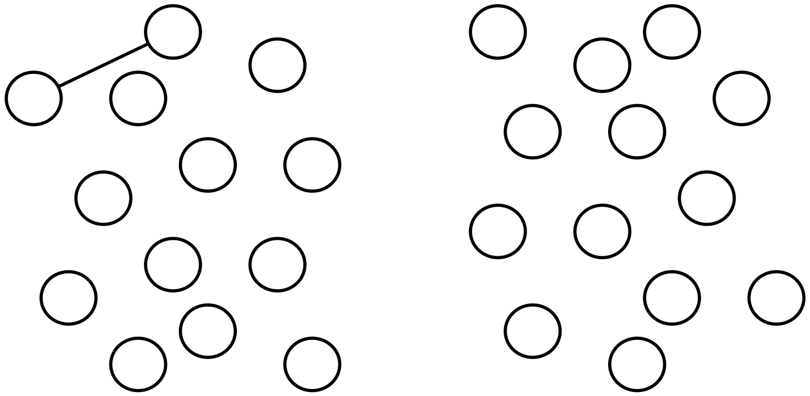




Stochastic Block Model (SBM)

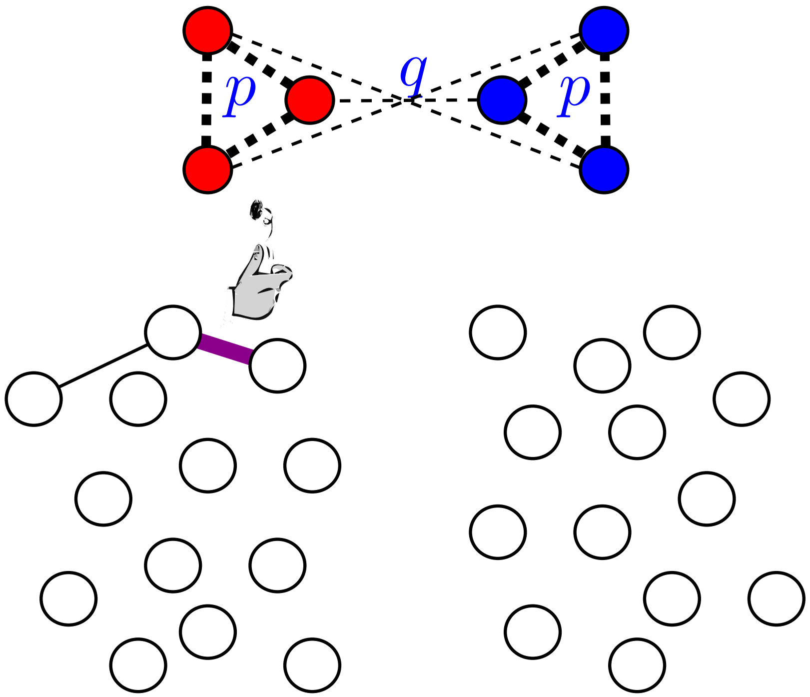
- "Communities" V_1, V_2, with $|V_1| = |V_2|$.
- include each edge with probability
  - p if edge inside V_1 or V_2,
  - q if edge between V_1 and V_2.





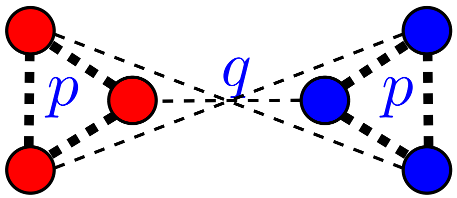
Stochastic Block Model (SBM)

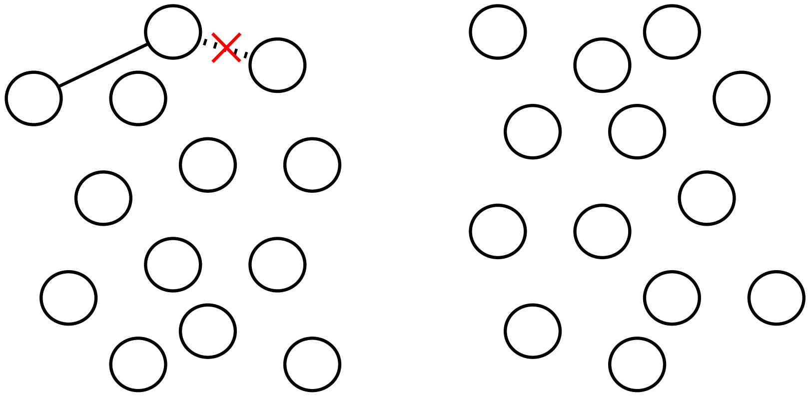
- "Communities" V_1, V_2, with $|V_1| = |V_2|$.
- include each edge with probability
  - p if edge inside V_1 or V_2,
  - q if edge between V_1 and V_2.



Stochastic Block Model (SBM)

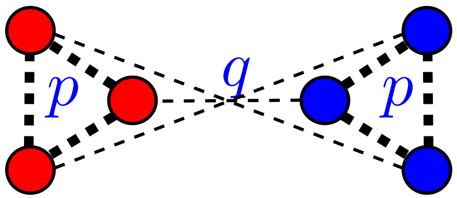
- "Communities" V_1, V_2, with $|V_1| = |V_2|$.
- include each edge with probability
  - p if edge inside V_1 or V_2,
  - q if edge between V_1 and V_2.

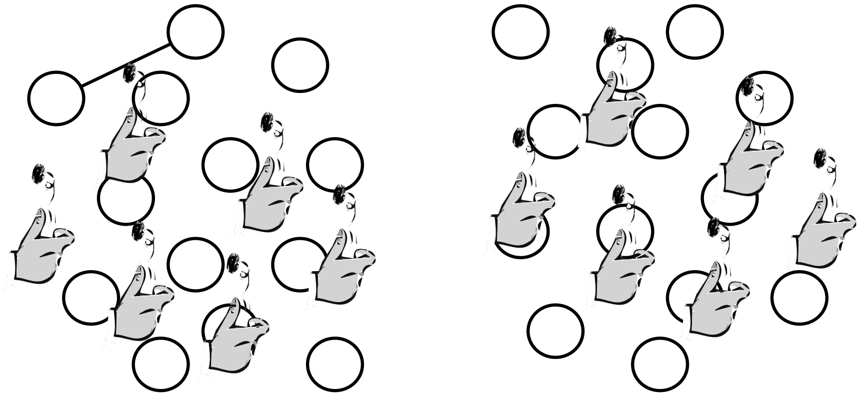




Stochastic Block Model (SBM)

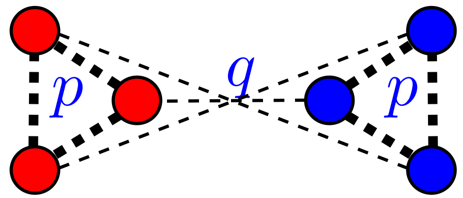
- "Communities" V_1, V_2, with $|V_1| = |V_2|$.
- include each edge with probability
  - p if edge inside V_1 or V_2,
  - q if edge between V_1 and V_2.

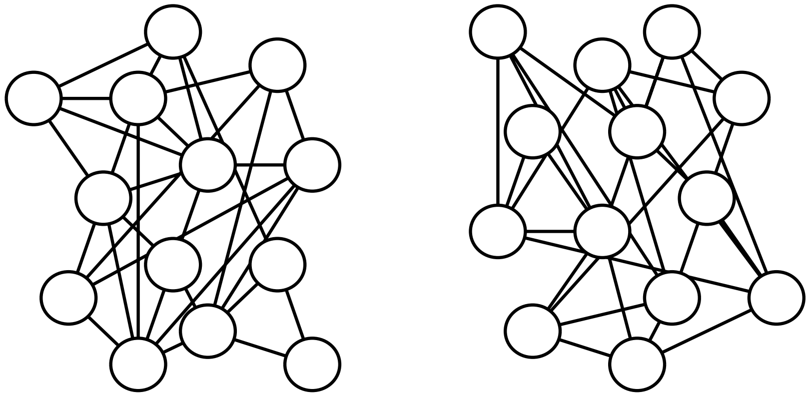




Stochastic Block Model (SBM)

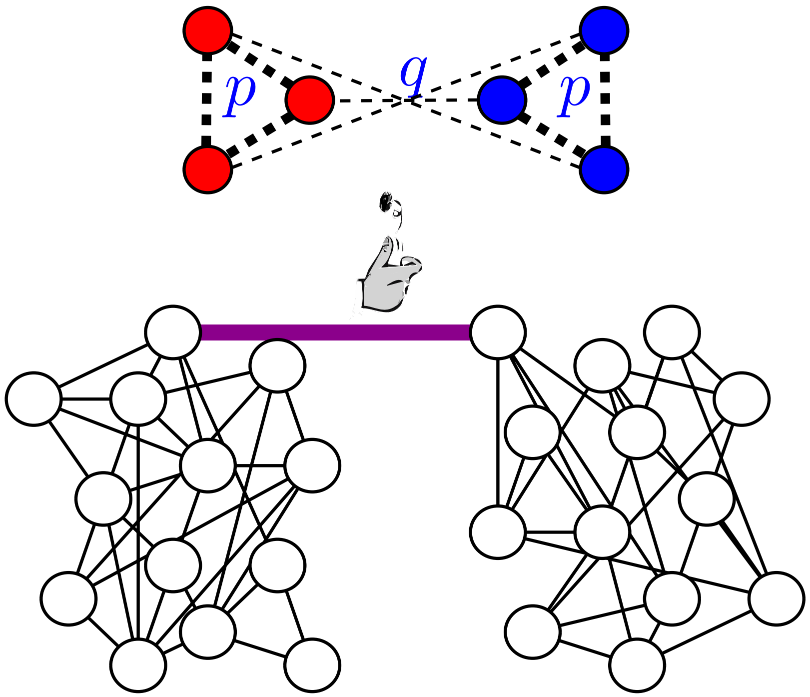
- "Communities" V_1, V_2, with $|V_1| = |V_2|$.
- include each edge with probability
  - p if edge inside V_1 or V_2,
  - q if edge between V_1 and V_2.





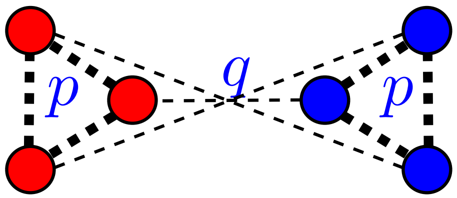
Stochastic Block Model (SBM)

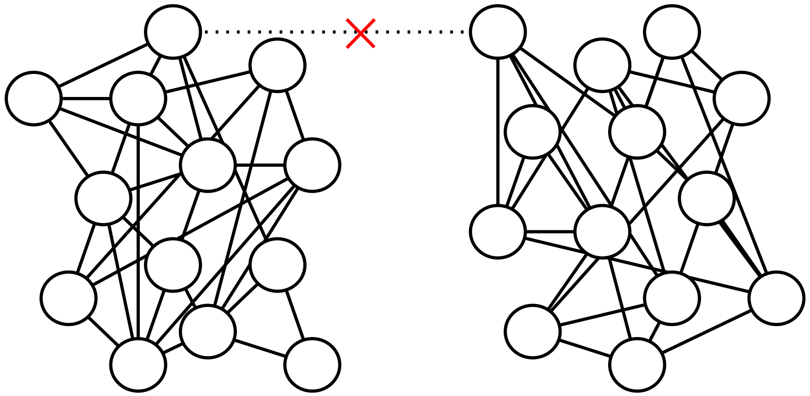
- "Communities" V_1, V_2, with $|V_1| = |V_2|$.
- include each edge with probability
  - p if edge inside V_1 or V_2,
  - q if edge between V_1 and V_2.



Stochastic Block Model (SBM)

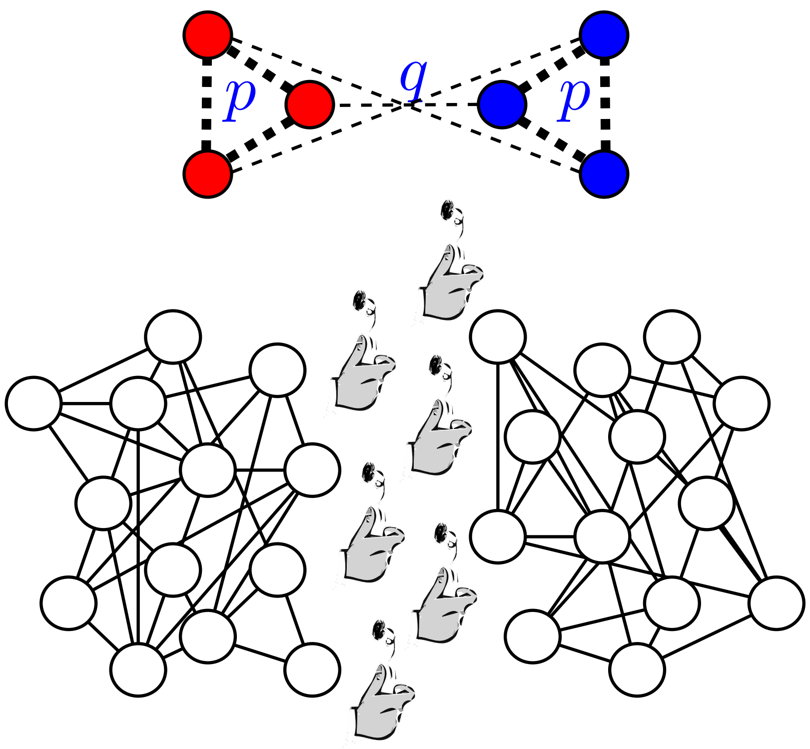
- "Communities" V_1, V_2, with $|V_1| = |V_2|$.
- include each edge with probability
  - p if edge inside V_1 or V_2,
  - q if edge between V_1 and V_2.





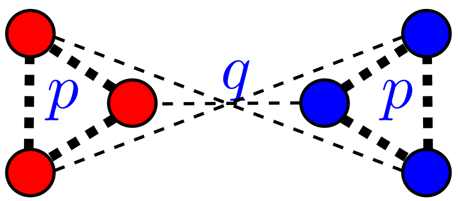
Stochastic Block Model (SBM)

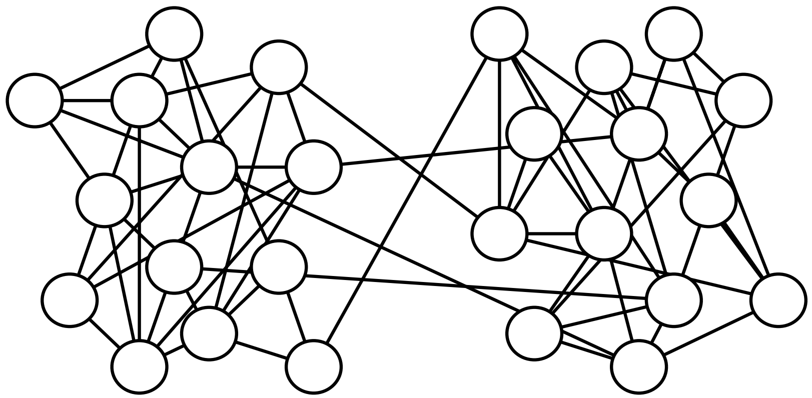
- "Communities" V_1, V_2, with $|V_1| = |V_2|$.
- include each edge with probability
  - p if edge inside V_1 or V_2,
  - q if edge between V_1 and V_2.



Stochastic Block Model (SBM)

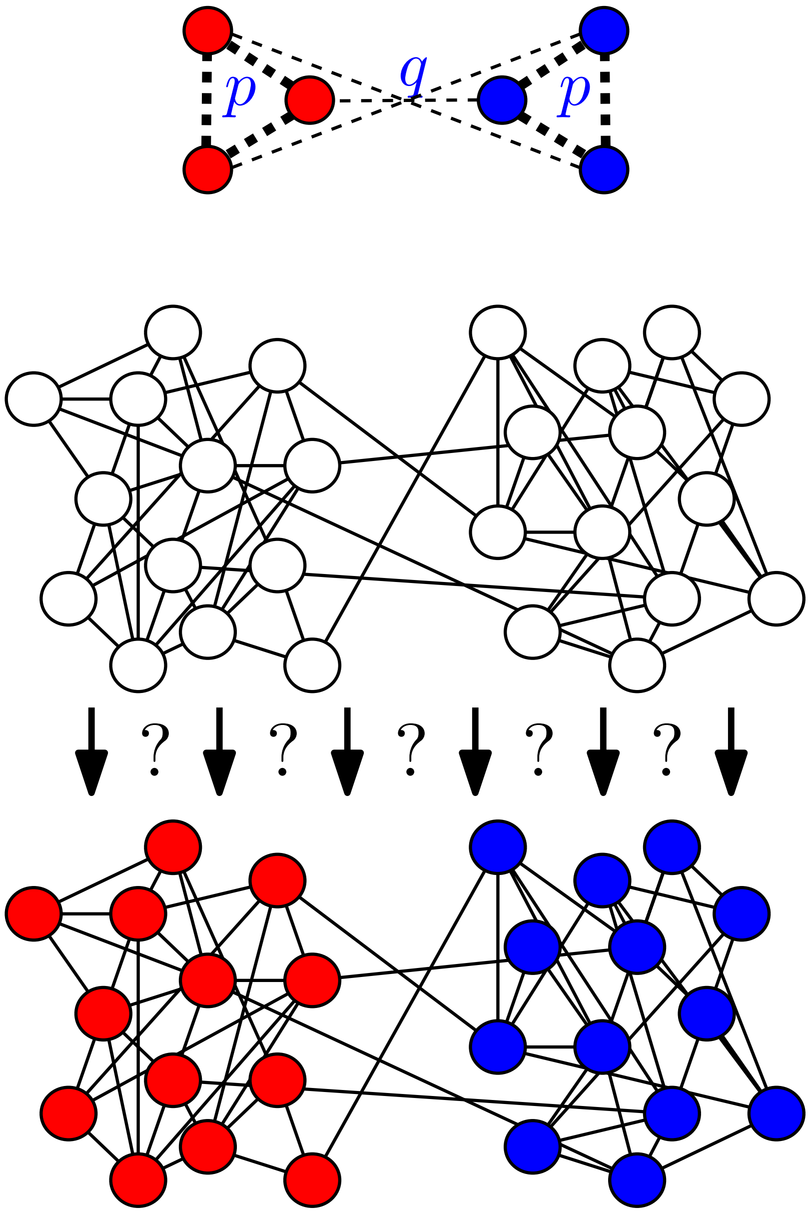
- "Communities" V_1, V_2, with $|V_1| = |V_2|$.
- include each edge with probability
  - p if edge inside V_1 or V_2,
  - q if edge between V_1 and V_2.





Stochastic Block Model (SBM)

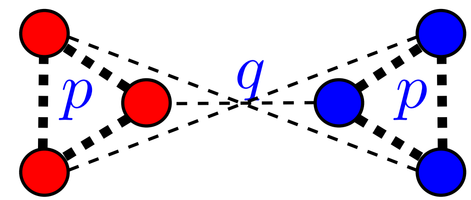
- "Communities" V_1, V_2, with $|V_1| = |V_2|$.
- include each edge with probability
  - p if edge inside V_1 or V_2,
  - q if edge between V_1 and V_2.



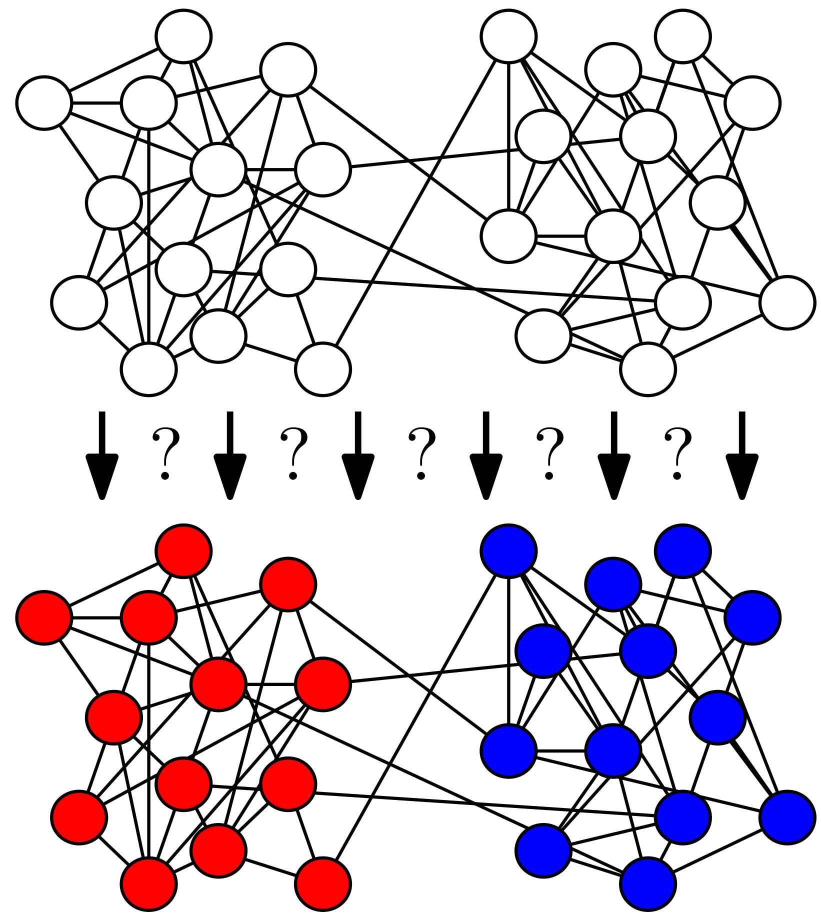
"Reconstruction" problem.
Given graph generated by SBM, find original clusters.

Stochastic Block Model (SBM)

- "Communities" V_1, V_2, with $|V_1| = |V_2|$.
- include each edge with probability
  - p if edge inside V_1 or V_2,
  - q if edge between V_1 and V_2.



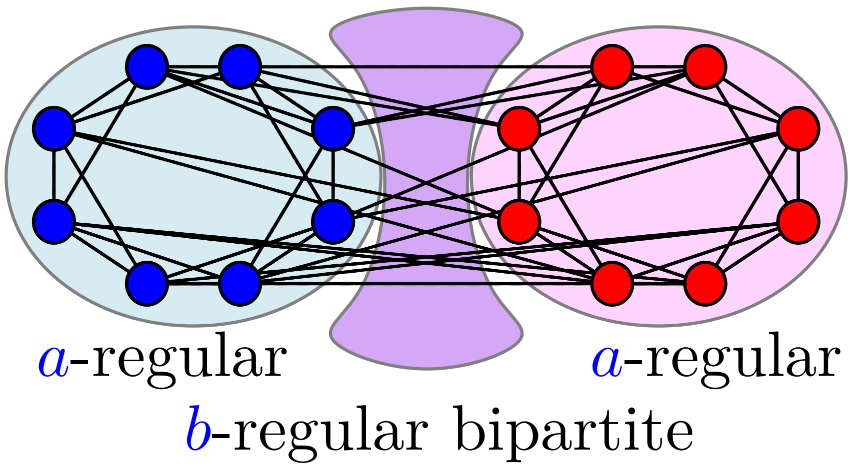
"Reconstruction" problem. Given graph generated by SBM, find original clusters.



Theorem. [Mossel et al. 2012-] Clustering possible **if and only if** p and q in a precise regime.

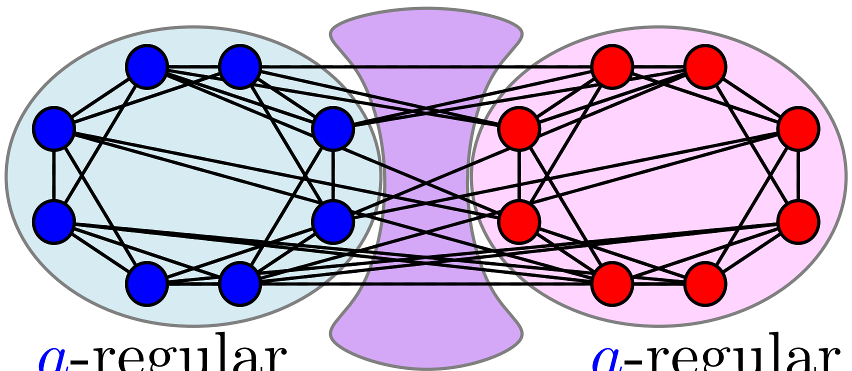
Clustering with **Averaging Dynamics**

Regular Stochastic Block Model:



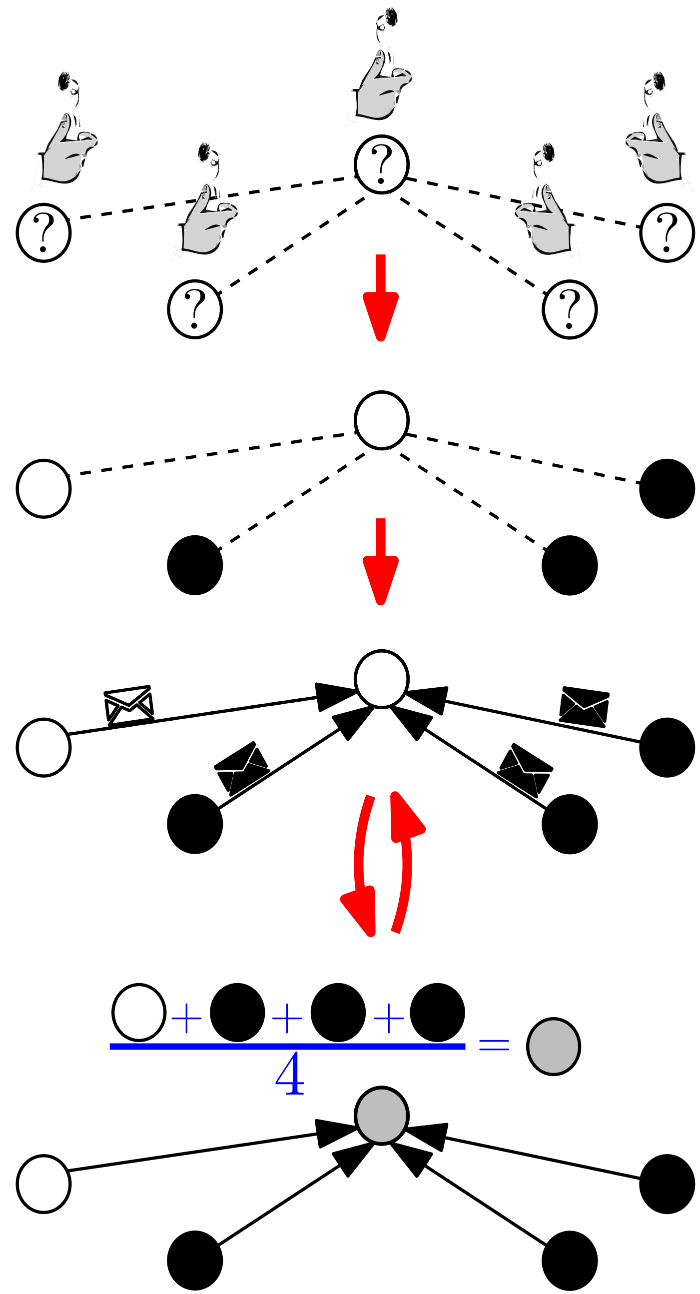
Clustering with **Averaging Dynamics**

Regular Stochastic Block Model:

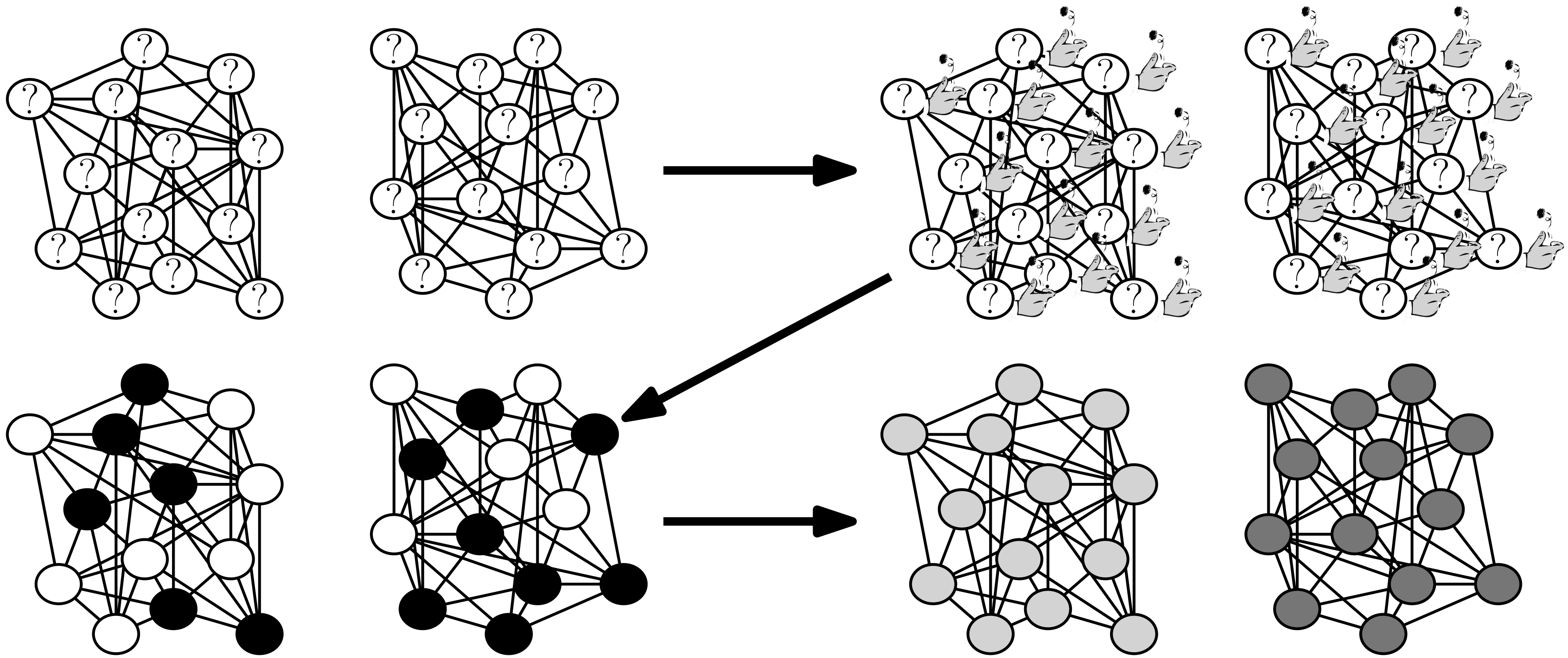


All nodes at the same time:

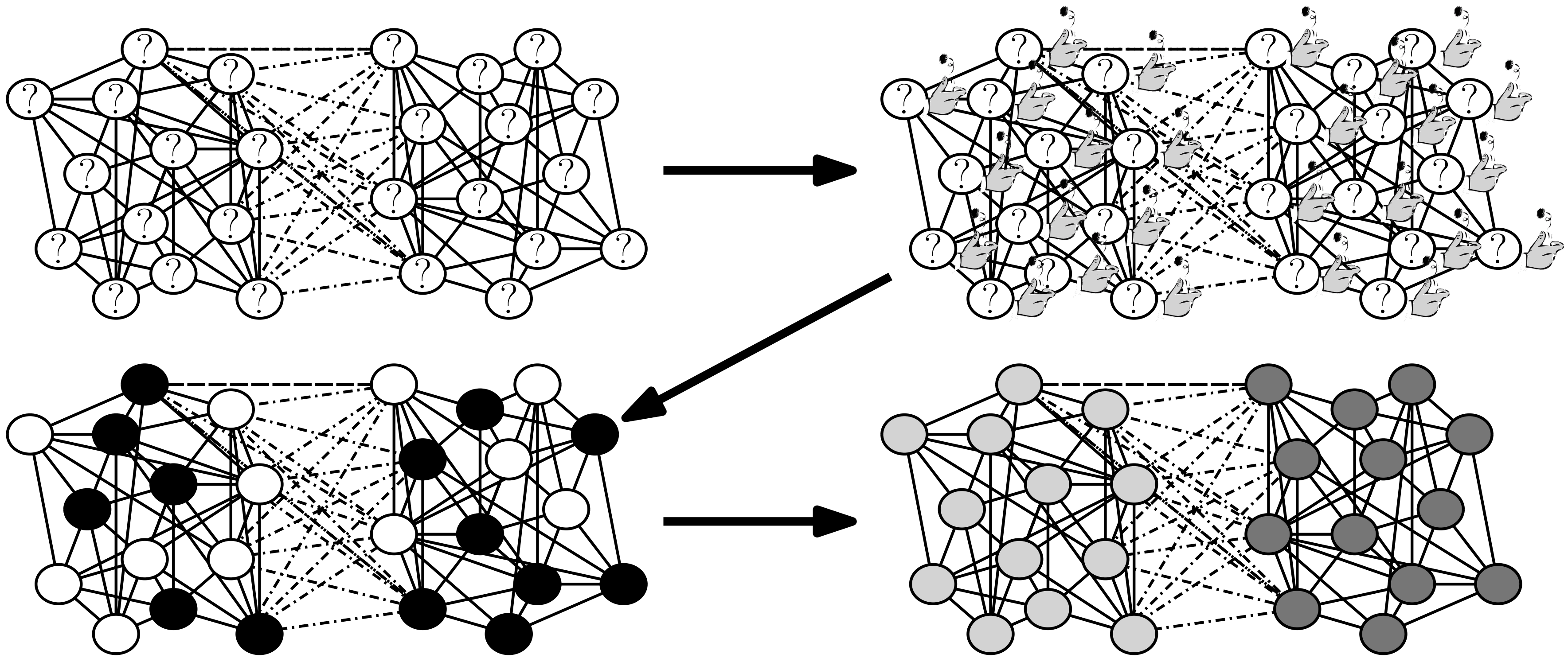
- At $t = 0$, randomly pick value $x^{(t)} \in \{+1, -1\}$
- Then, at each round set value $x^{(t)}$ to average of neighbors



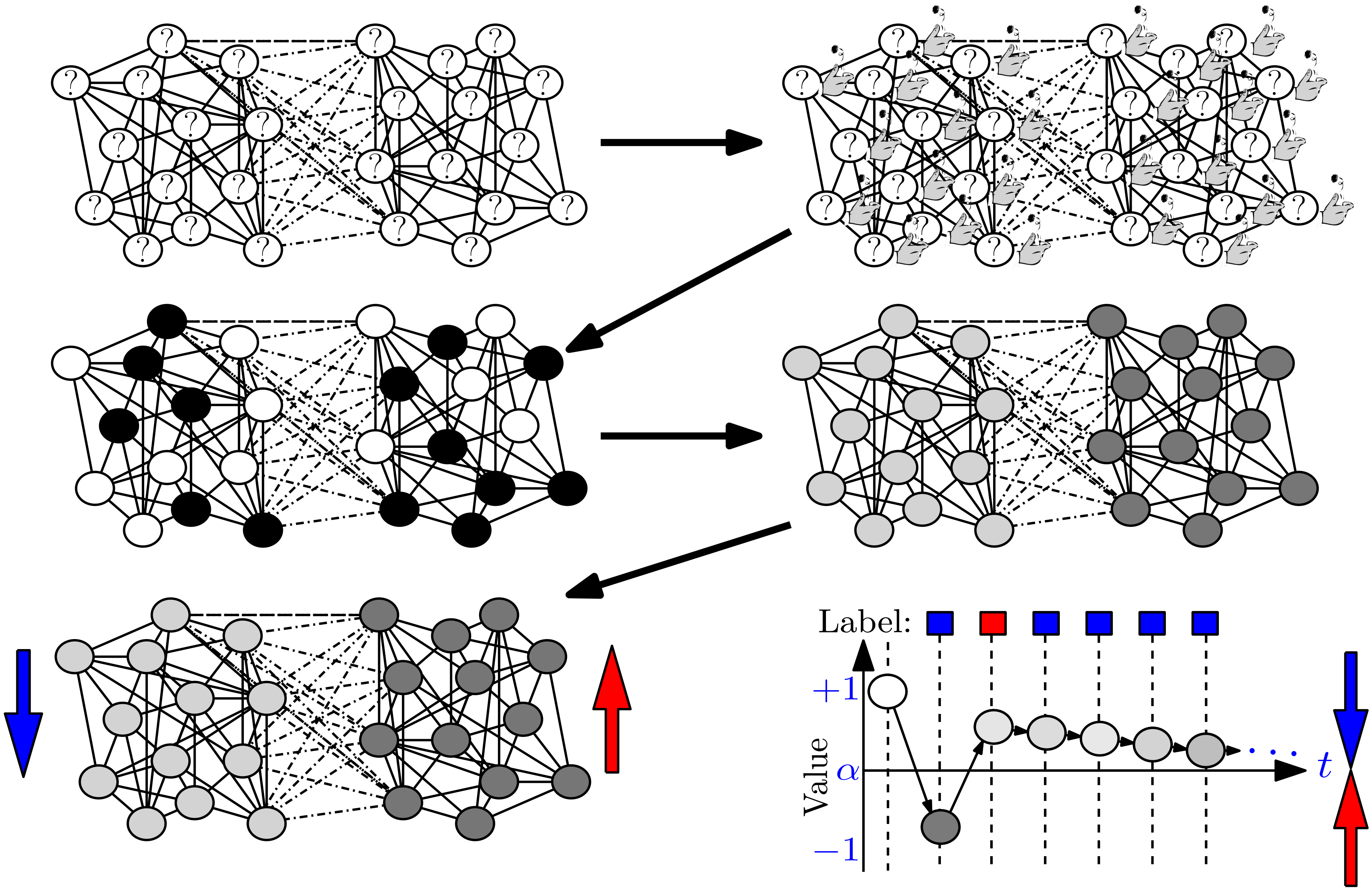
Why it Works: Intuition



Why it Works: Intuition



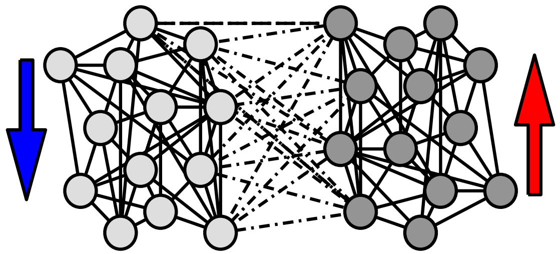
Why it Works: Intuition



- Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise

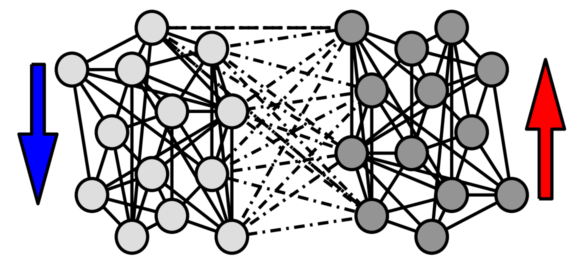
Why It Works: Proof Idea

Theorem. In Regular Stochastic Block Model with $a - b > \sqrt{2(a+b)}$, Averaging Dynamics finds clusters after $\frac{\log n}{\log \lambda_2 / \lambda_3}$ steps with high probability.



Why It Works: Proof Idea

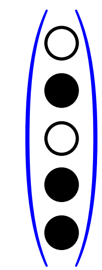
Theorem. In Regular Stochastic Block Model with $a - b > \sqrt{2(a+b)}$, Averaging Dynamics finds clusters after $\frac{\log n}{\log \lambda_2/\lambda_3}$ steps with high probability.



Averaging is a **linear** dynamics:

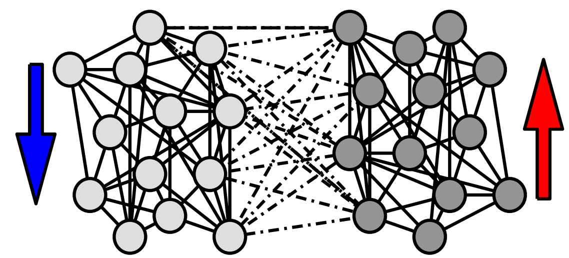
$$\mathbf{x}^{(t)} = P \cdot \mathbf{x}^{(t-1)} = P^t \cdot \mathbf{x}^{(0)}$$

P transition matrix of random walk on G and $\mathbf{x}^{(t)} =$



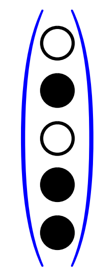
Why It Works: Proof Idea

Theorem. In Regular Stochastic Block Model with $a-b>\sqrt{2(a+b)}$, Averaging Dynamics finds clusters after $\frac{\log n}{\log \lambda_2/\lambda_3}$ steps with high probability.



Averaging is a **linear** dynamics:

$$\boxed{\mathbf{x}^{(t)} = P\cdot\mathbf{x}^{(t-1)} = P^t\cdot\mathbf{x}^{(0)}}$$

P transition matrix of random walk on G and $\mathbf{x}^{(t)} =$ 

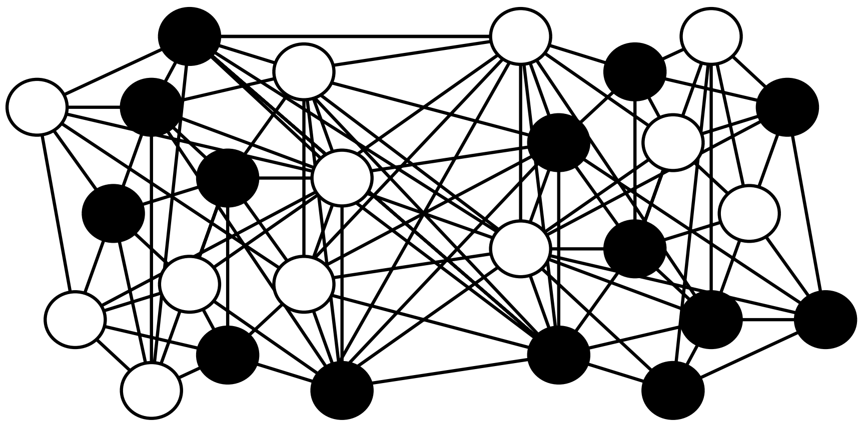
$$\mathbf{x}^{(t)} = \frac{1}{\tilde{\Theta}(\sqrt{n})}\begin{pmatrix}1\\ \vdots\\ 1\\ 1\\ \vdots\\ 1\end{pmatrix} + \left(\frac{a-b}{a+b}\right)^t \frac{1}{\tilde{\Theta}(\sqrt{n})}\begin{pmatrix}1\\ \vdots\\ 1\\ -1\\ \vdots\\ -1\end{pmatrix} + \mathbf{e}^{(t)}$$

negligible after $t \gg \frac{\log n}{\log \lambda_2/\lambda_3}$

$$\text{sign}(\mathbf{x}^{(t)}(u) - \mathbf{x}^{(t-1)}(u)) = \text{sign}\left(\begin{pmatrix}1\\ \vdots\\ 1\\ -1\\ \vdots\\ -1\end{pmatrix}\right)$$

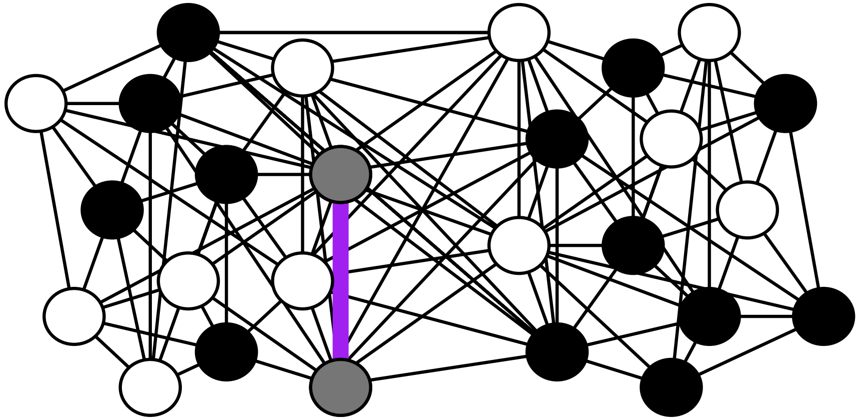
Asynchronous Averaging Dynamics

Asynchronous Averaging Dynamics (AAD):
Each node u *initially flips a coin and gets value* $+1$ *or* -1.
At each step, an edge $\{u, v\}$ *is chosen u.a.r. and* u *and* v *average their values.*



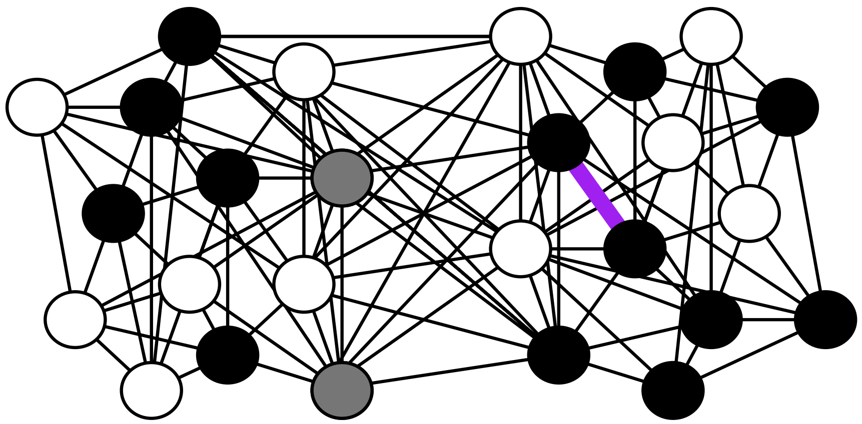
Asynchronous Averaging Dynamics

Asynchronous Averaging Dynamics (AAD):
Each node u *initially flips a coin and gets value* $+1$ *or* -1.
At each step, an edge $\{u, v\}$ *is chosen u.a.r. and* u *and* v *average their values.*



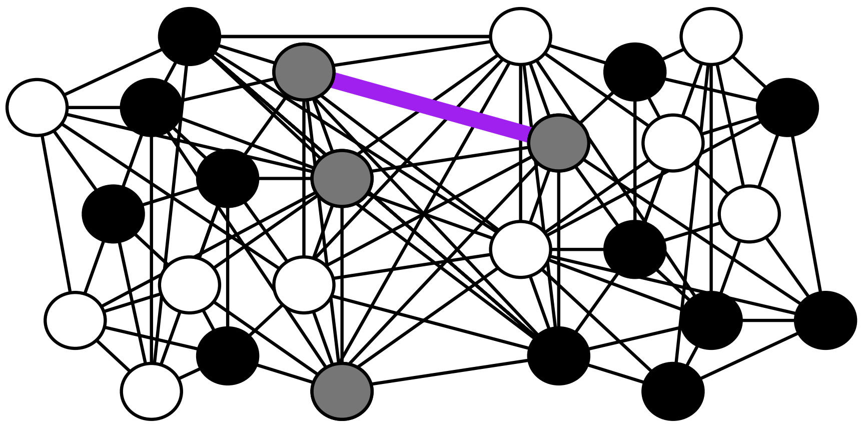
Asynchronous Averaging Dynamics

Asynchronous Averaging Dynamics (AAD):
Each node u initially flips a coin and gets value $+1$ or -1.
At each step, an edge $\{u, v\}$ is chosen u.a.r. and u and v average their values.



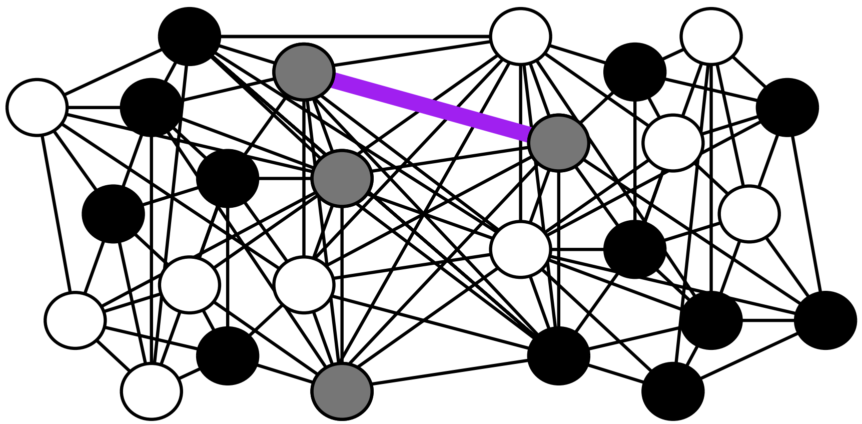
Asynchronous Averaging Dynamics

Asynchronous Averaging Dynamics (AAD):
Each node u *initially flips a coin and gets value* $+1$ *or* -1.
At each step, an edge $\{u, v\}$ *is chosen u.a.r. and* u *and* v *average their values.*



Asynchronous Averaging Dynamics

Asynchronous Averaging Dynamics (AAD):
Each node u initially flips a coin and gets value $+1$ or -1. At each step, an edge $\{u, v\}$ is chosen u.a.r. and u and v average their values.



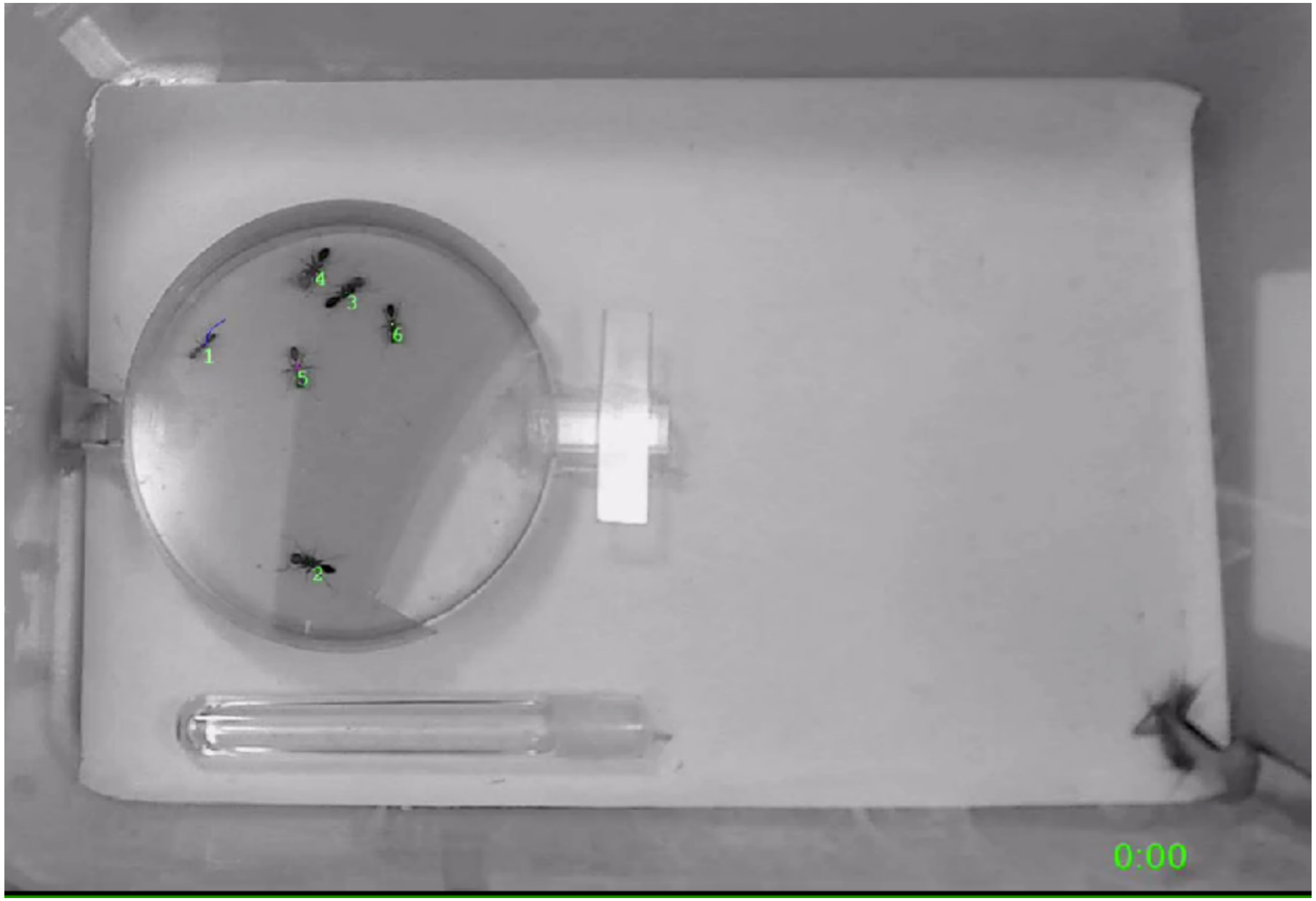
Theorem. In Regular Stochastic Block Model

- An AAD-based protocol finds clusters in $C_{\lambda_2-\lambda_1} n(\frac{a}{b} + \log n)$ with high probability.
- If $\lambda_2 \ll \frac{\lambda_3^2}{\log^2 n}$, another AAD-based protocol finds clusters after $\mathcal{O}(\frac{n}{\lambda_3} \log^2 n)$ steps with high probability.

Part II

Biological Distributed Algorithms

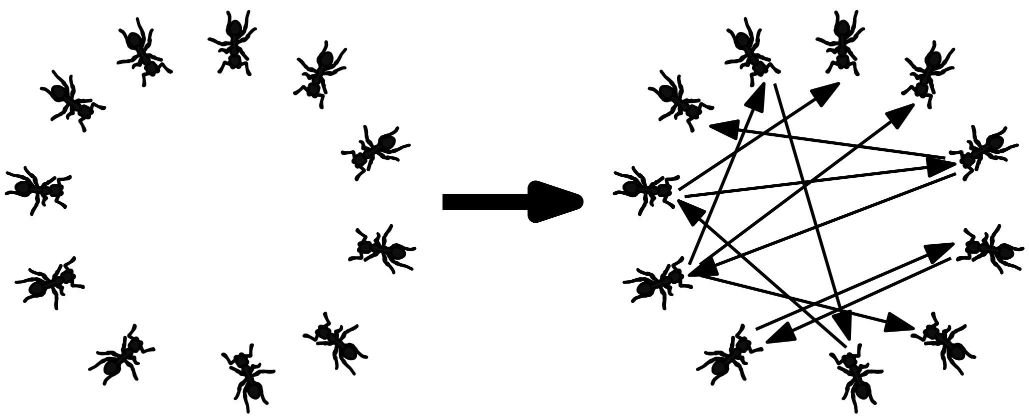
Recruitment in Desert **Ants**



Cataglyphis niger needs to recruit nest mates to carry food.
Data suggest that ants communicate by simple *noisy* interactions.

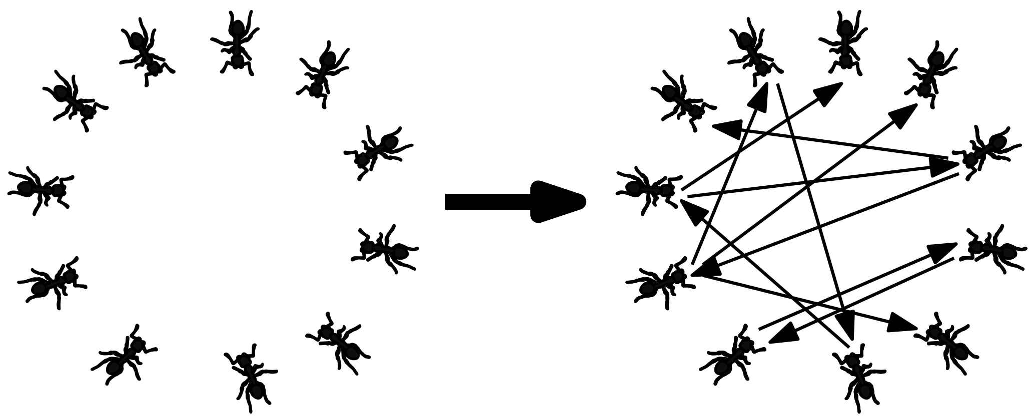
Noisy & Stochastic Interactions

Stochastic Interactions. At each round, each agent receives a message from another random agent.

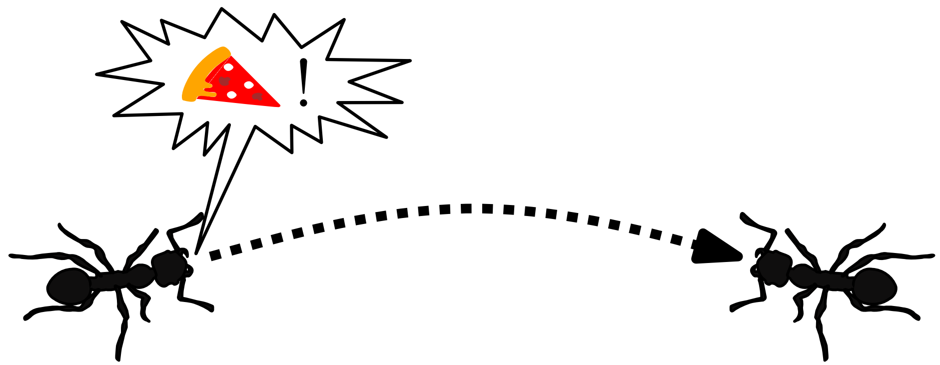


Noisy & Stochastic Interactions

Stochastic Interactions. At each round, each agent receives a message from another random agent.

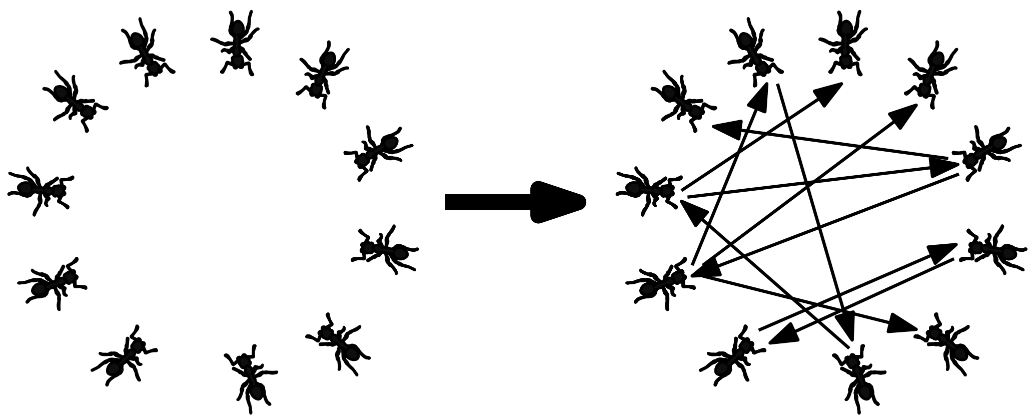


Noisy Communication. Before being received, each bit is flipped with probability $1/2 - \epsilon_n$.

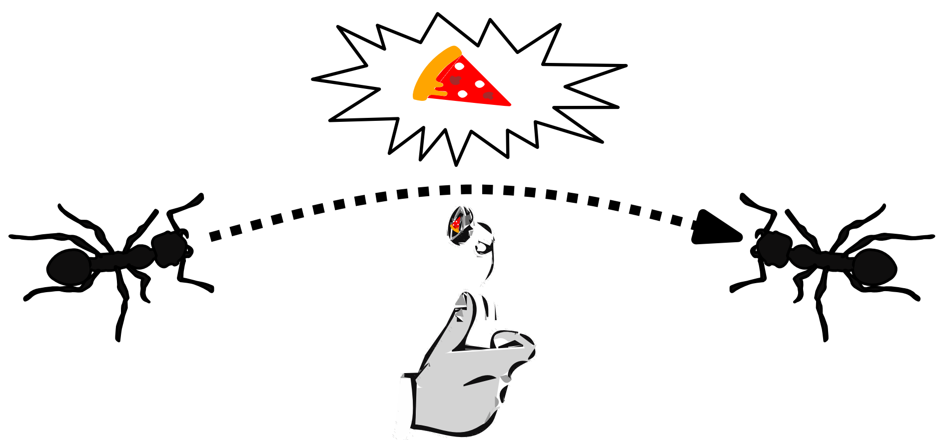


Noisy & Stochastic Interactions

Stochastic Interactions.
At each round, each agent receives a message from another random agent.

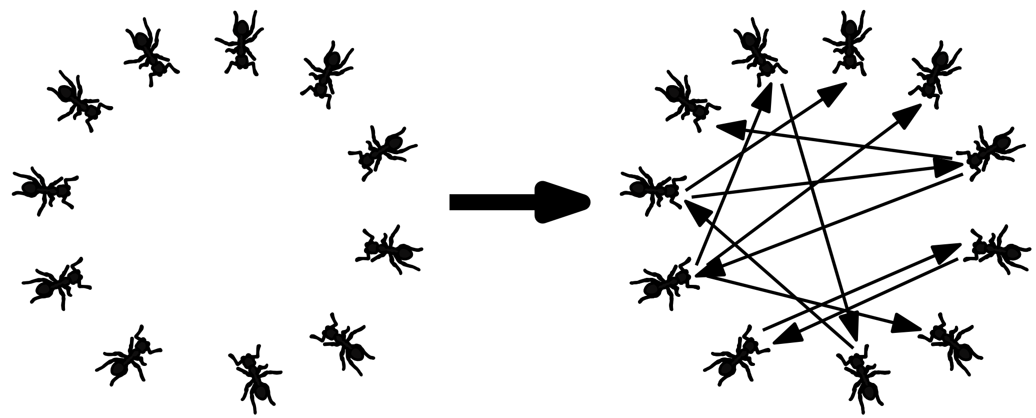


Noisy Communication.
Before being received, each bit is flipped with probability $1/2 - \epsilon_n$.

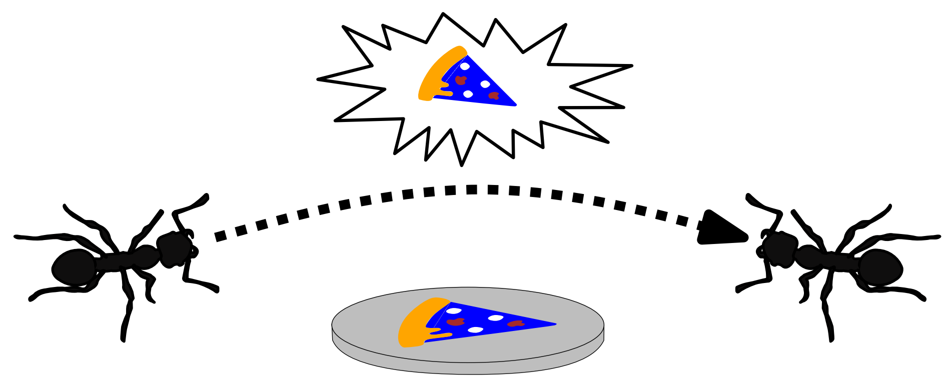


Noisy & Stochastic Interactions

Stochastic Interactions.
At each round, each agent receives a message from another random agent.

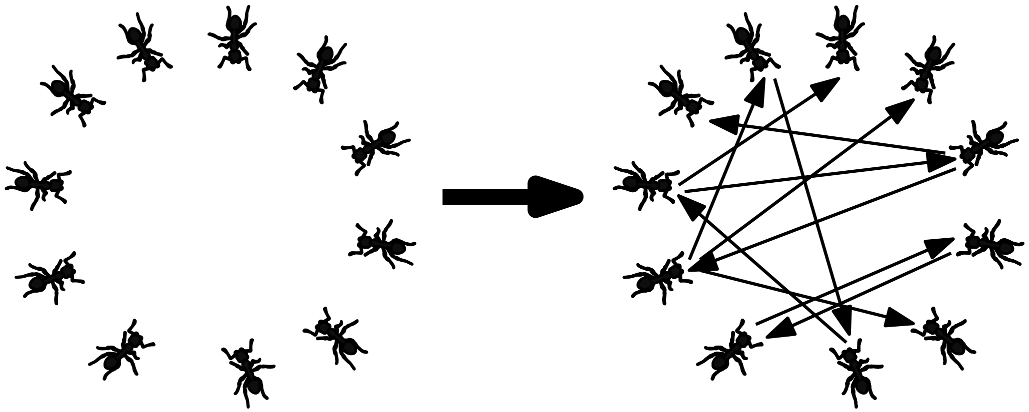


Noisy Communication.
Before being received, each bit is flipped with probability $1/2 - \epsilon_n$.

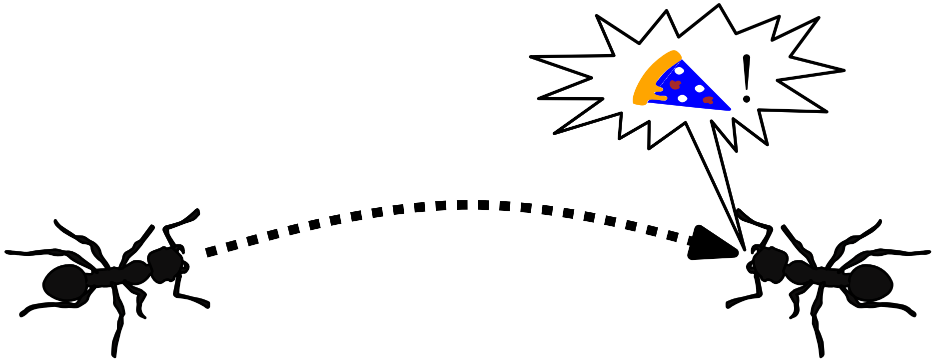


Noisy & Stochastic Interactions

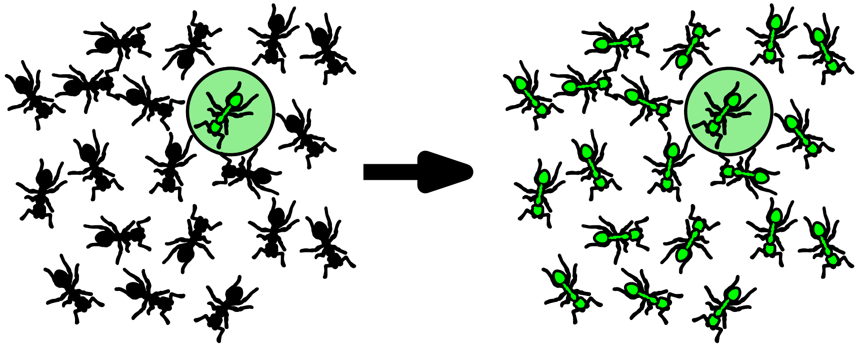
Stochastic Interactions.
At each round, each agent receives a message from another random agent.



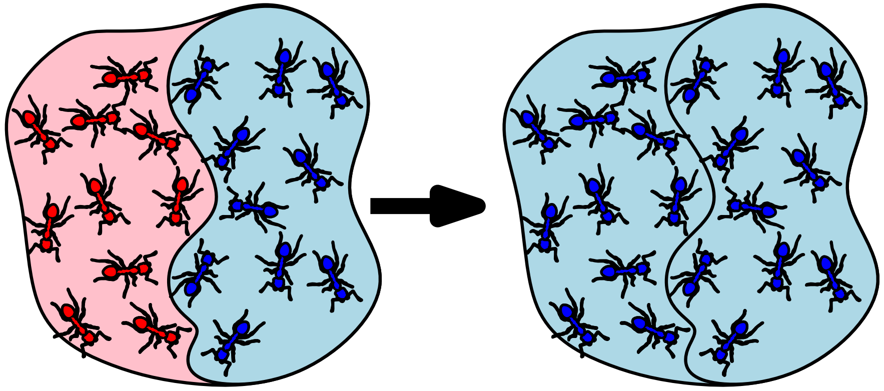
Noisy Communication.
Before being received, each bit is flipped with probability $1/2 - \epsilon_n$.



Noisy vs Noiseless Broadcast and Consensus



Broadcast. All nodes eventually receive the message of the source.

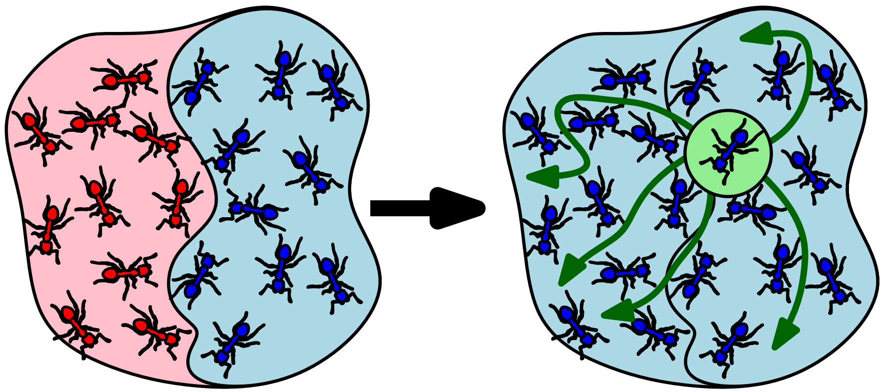


(Valid) Consensus. All nodes eventually support the value initially supported by one of them.

Reductions and Lower Bounds

Broadcast $\Longrightarrow$ Consensus

Noiseless Consensus
$\Longrightarrow$ **Noiseless**
(variant of) Broadcast

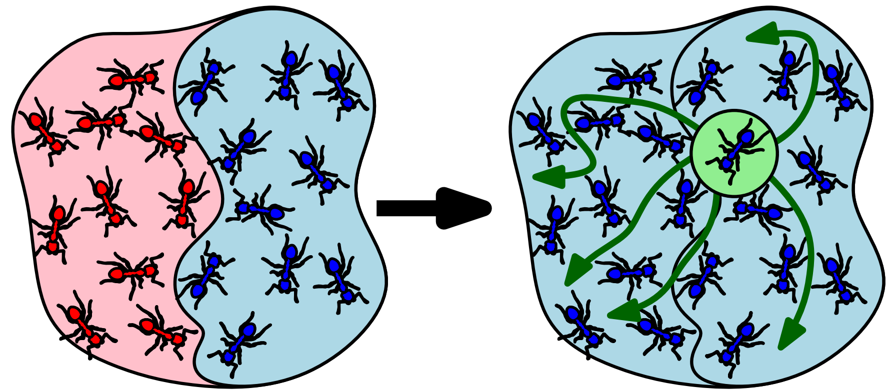


Noiseless Consensus and Broadcast are "*equivalent*"

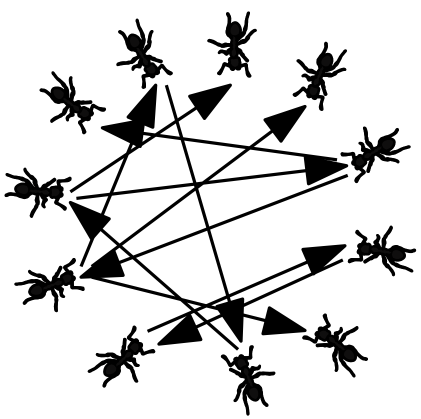
Reductions and Lower Bounds

Broadcast $\Longrightarrow$ Consensus

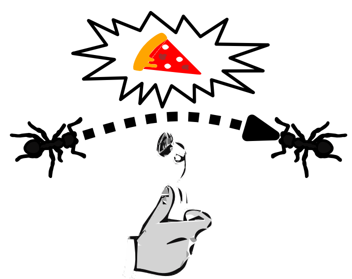
Noiseless Consensus
$\Longrightarrow$ **Noiseless**
(variant of) Broadcast



Noiseless Consensus and Broadcast are "*equivalent*"



$+$



$\Longrightarrow$

Noisy Consensus:
$\Theta(\frac{\log n}{\epsilon^2})$ rounds

Noisy Broadcast:
$\Theta(n \cdot \frac{\log n}{\epsilon^2})$ rounds

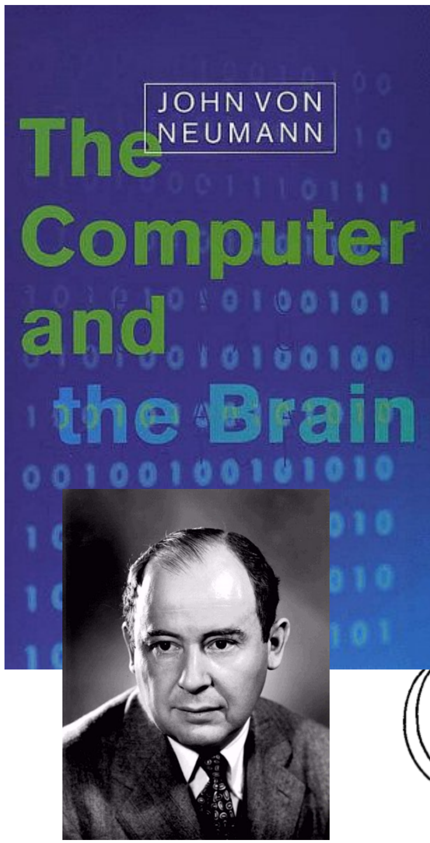
Noisy Broadcast is *exponentially harder* than **Noisy** Consensus

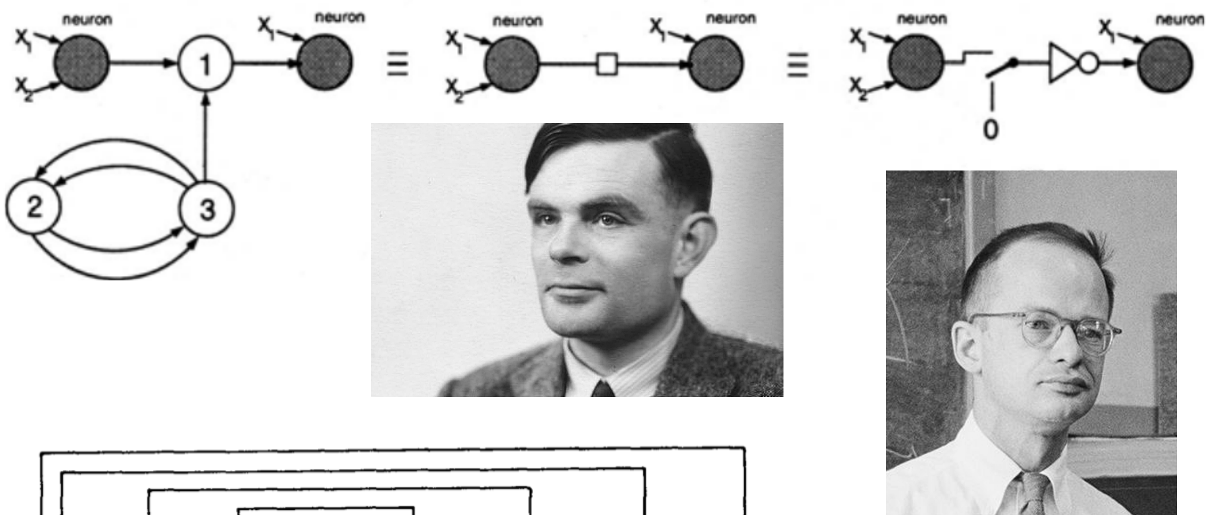
Part III

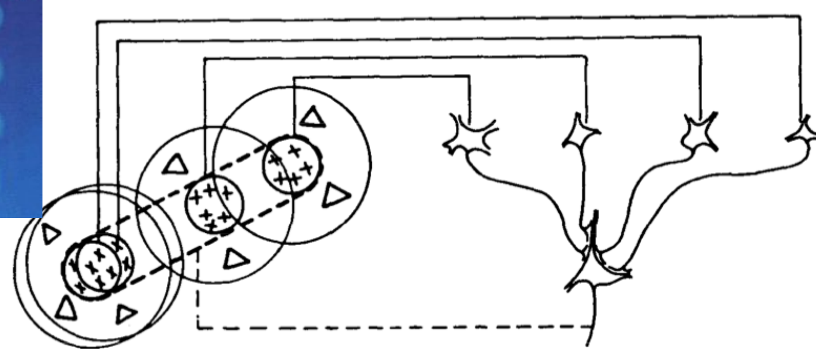
TCS and Theoretical Neuroscience

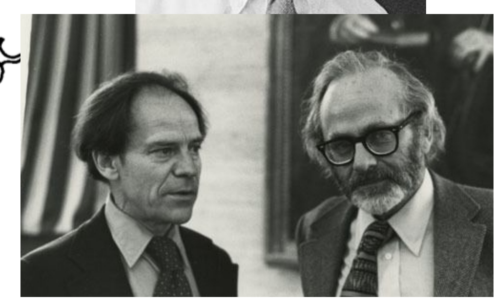
The Brain and Computation

Von Neumann, Turing, McCulloch, Pitts, Barlow... were interested in the other field to better understand theirs.



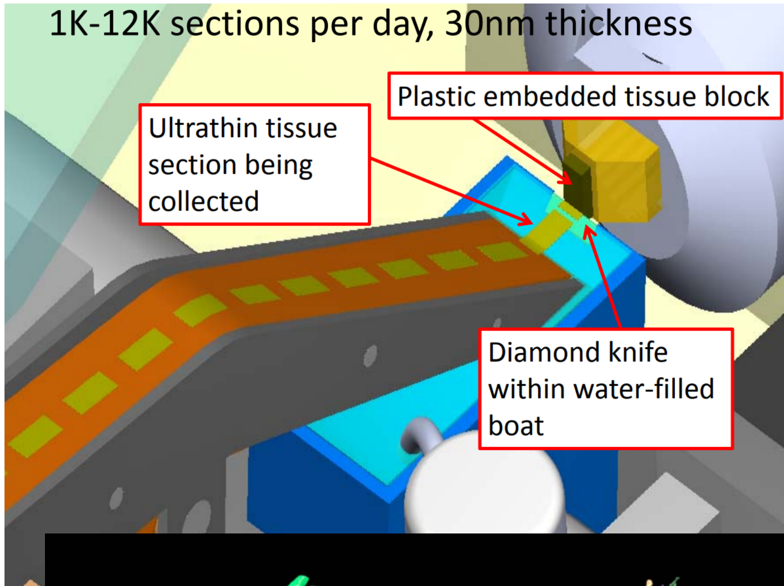


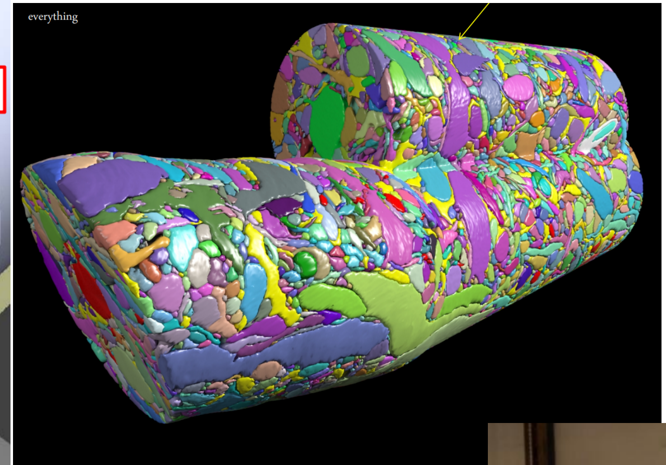




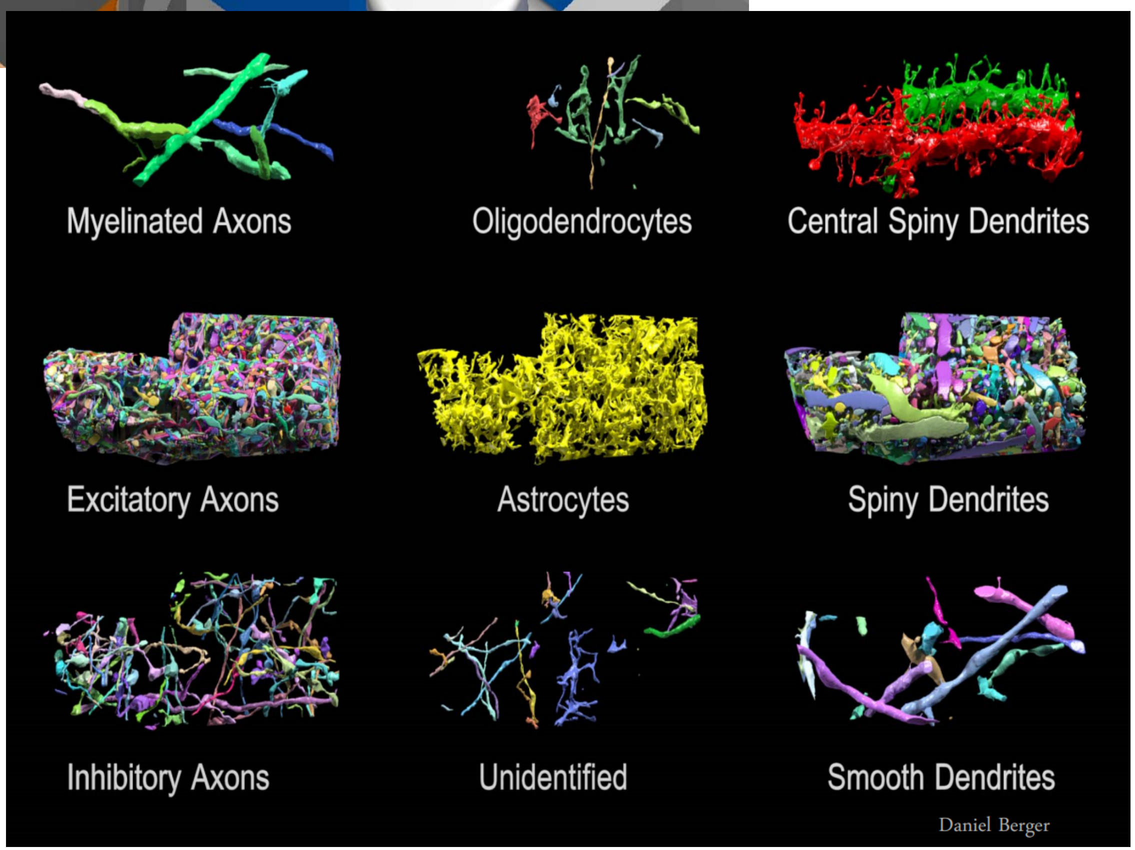
Both fields have exploded in knowledge but have also grown further apart.

Computational Neuroscience: Data





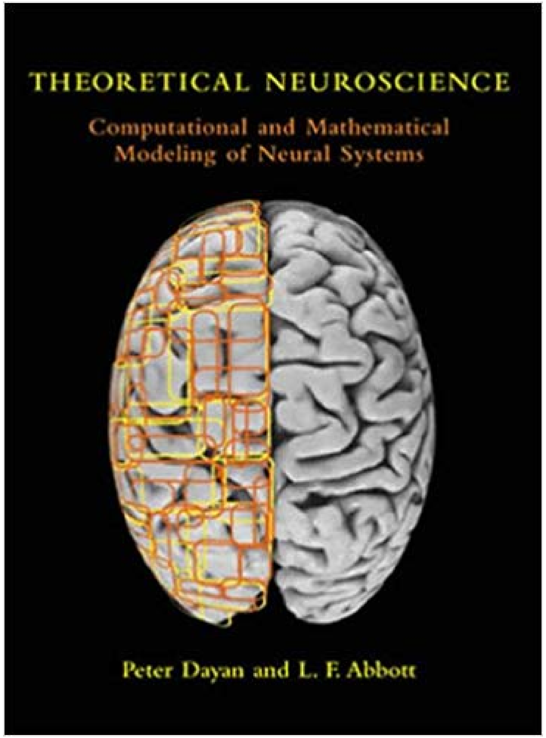
1 mm^3 of mouse brain $\Longrightarrow$ 300 TB of image data



Computational Neuroscience: Theory

Issues:

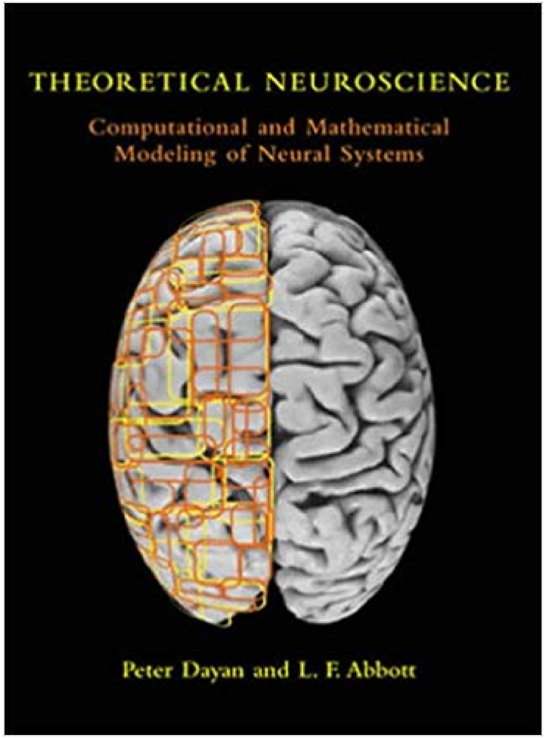
- Far from experimentalists



Computational Neuroscience: Theory

Issues:

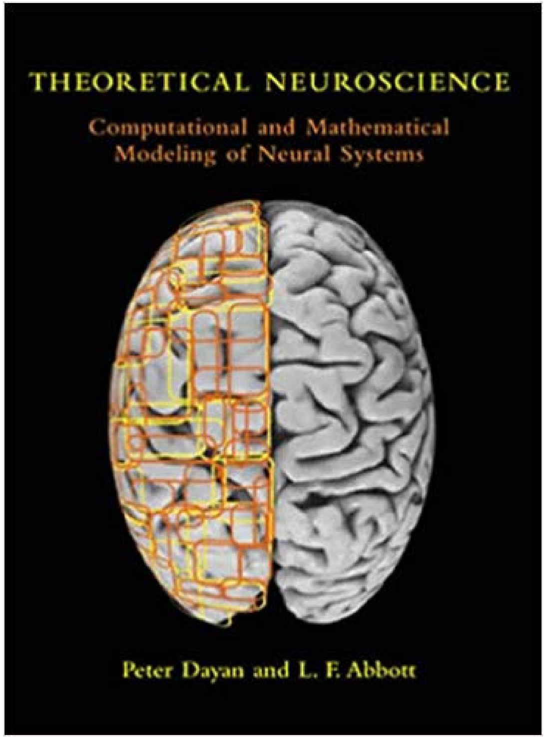
- Far from experimentalists
- Internally divided



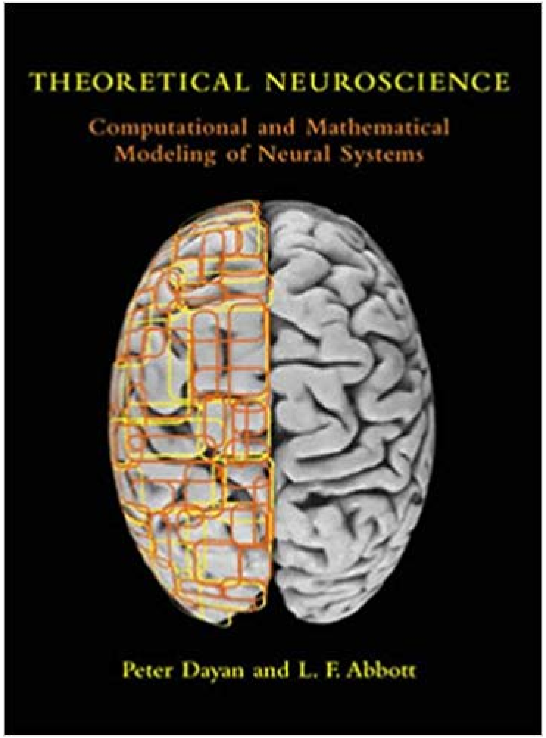
Computational Neuroscience: Theory

Issues:

- Far from experimentalists
- Internally divided
- Led mostly by physicists



Computational Neuroscience: Theory



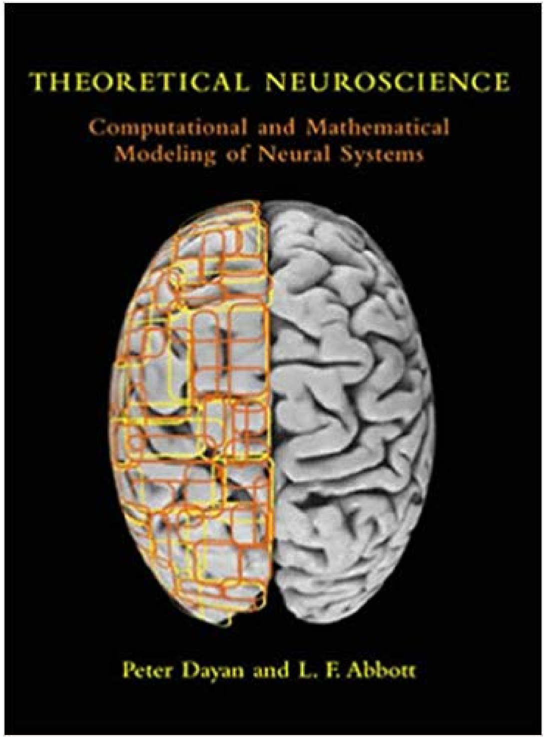
Issues:

- Far from experimentalists
- Internally divided
- Led mostly by physicists

Theories:

- Neural networks for learning: Pitts & McCulloch ('47), Rosenblatt ('58), Hubel & Wiesel ('62), ...

Computational Neuroscience: Theory



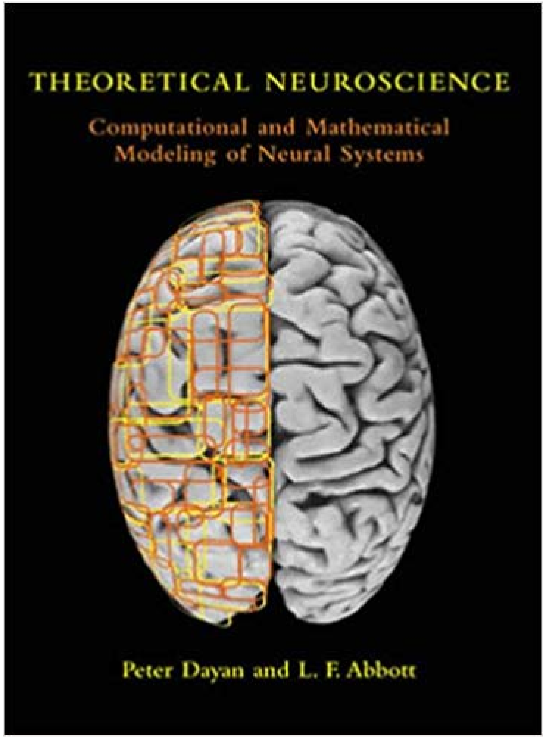
Issues:

- Far from experimentalists
- Internally divided
- Led mostly by physicists

Theories:

- Neural networks for learning: Pitts & McCulloch ('47), Rosenblatt ('58), Hubel & Wiesel ('62), ...
- Neural-dynamics model for specific neural phenomena (associative memory, grid cells, place cells, oscillations, ...)

Computational Neuroscience: Theory



Issues:

- Far from experimentalists
- Internally divided
- Led mostly by physicists

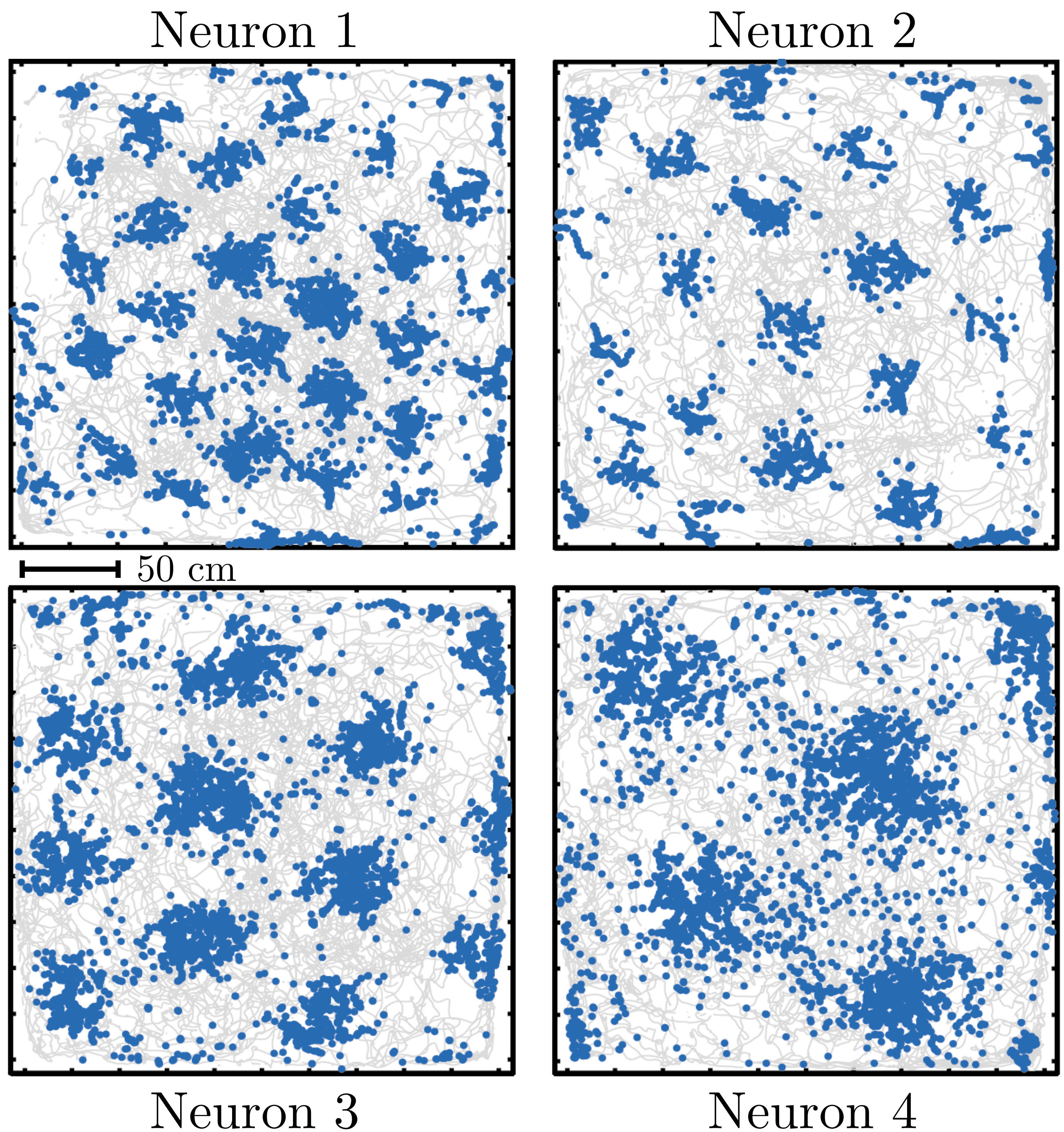
Theories:

- Neural networks for learning: Pitts & McCulloch ('47), Rosenblatt ('58), Hubel & Wiesel ('62), ...
- Neural-dynamics model for specific neural phenomena (associative memory, grid cells, place cells, oscillations, ...)
- Works from *Theoretical Computer Science*: Neuroidal Model by Valiant ('94), models of associative memory by Papadimitriou et al, ('15), Lynch et al. ('16) and Navlakha et al. ('17), ...

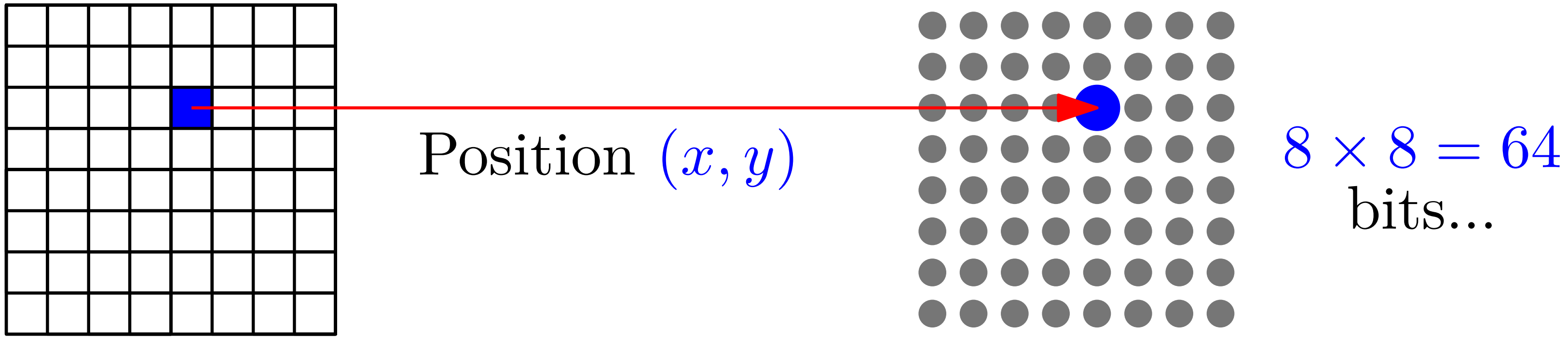
Does the Brain use Algorithms?

How are you aware of your location in space?

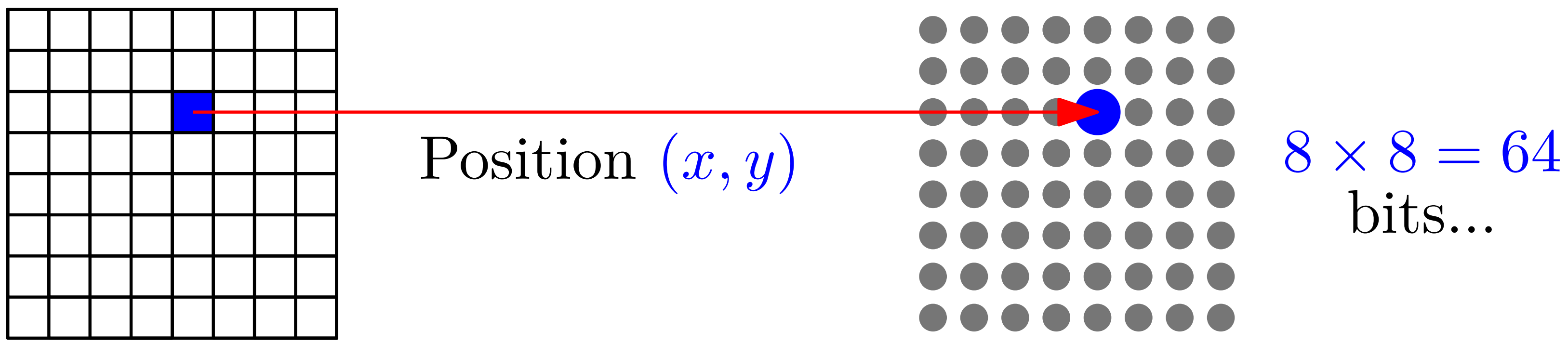
2014 Nobel Prize in Physiology to J. O'Keefe & M. B. and E. Moser for discovery of cells that constitute a positioning system in the brain

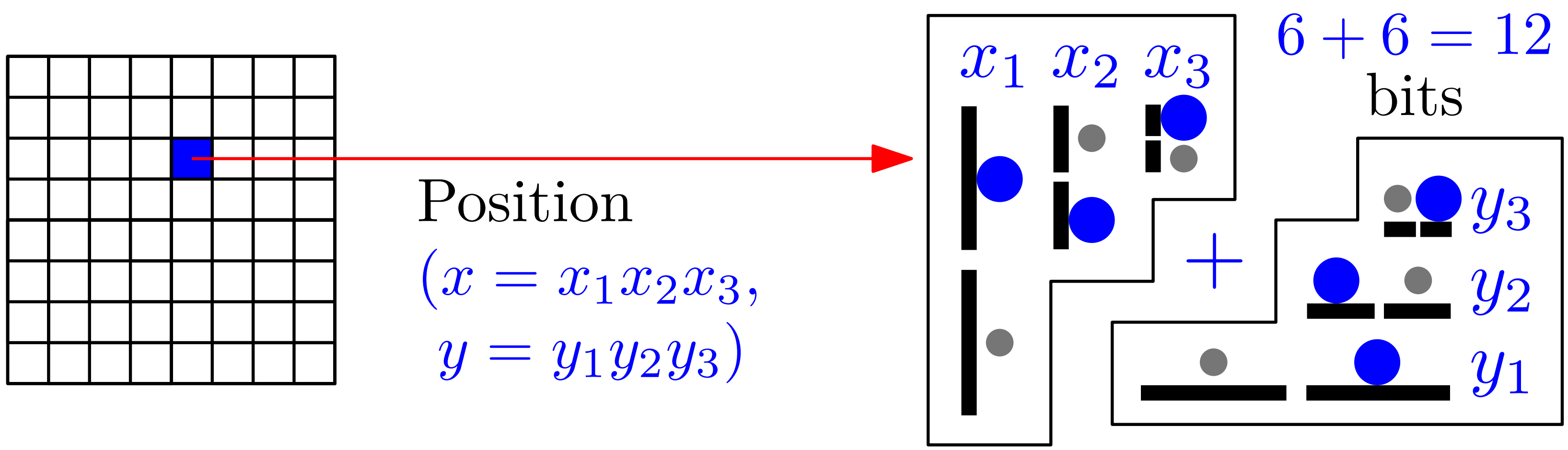


The Principle of Efficiency

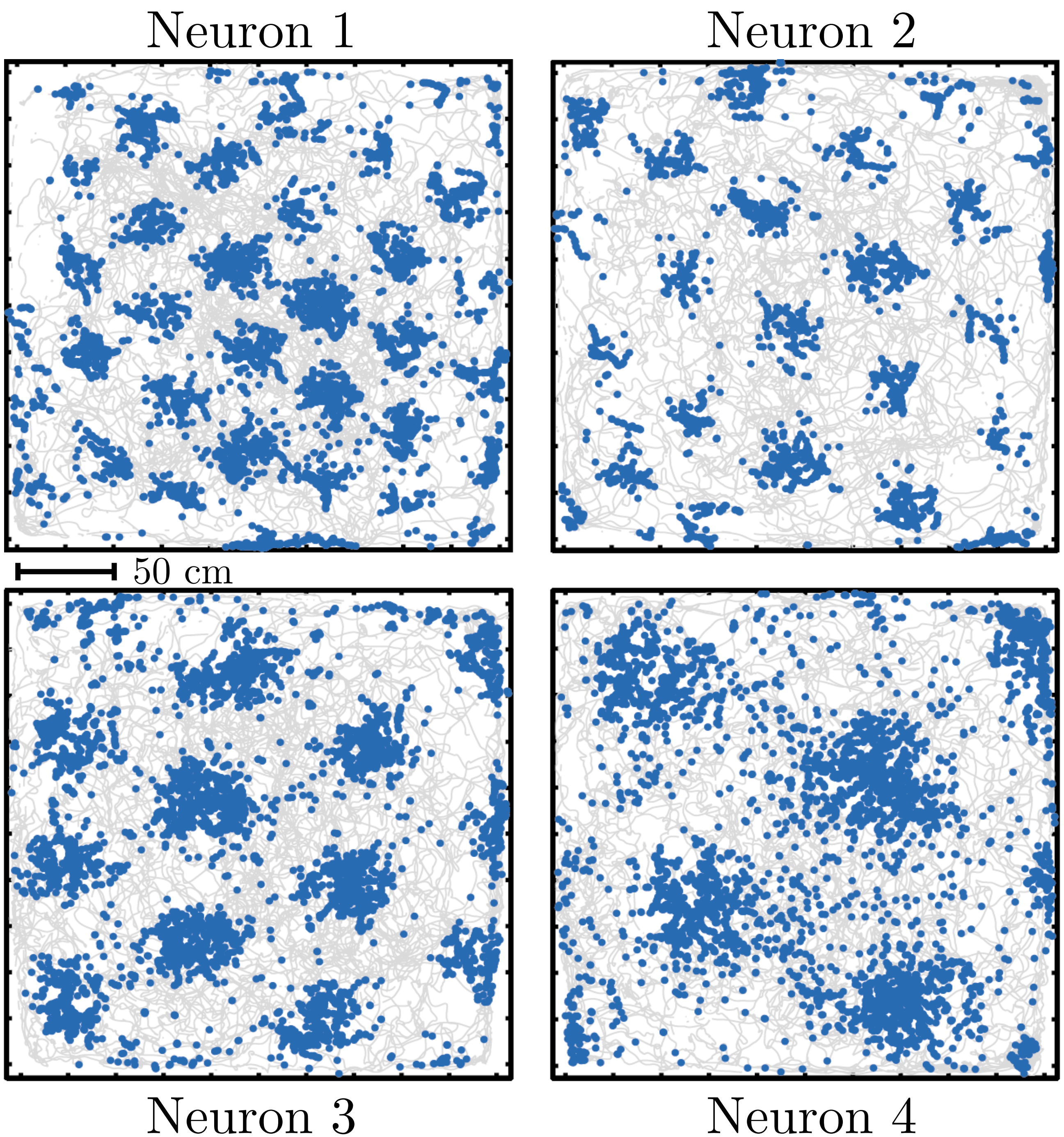


The Principle of Efficiency





Grid Cells Encodes Position Efficiently



A Model of Associative Memory

A model of *content-addressable associative memory*:
Hopfield networks [PNAS '84]
($\approx$ 8000 citations)

A Model of Associative Memory

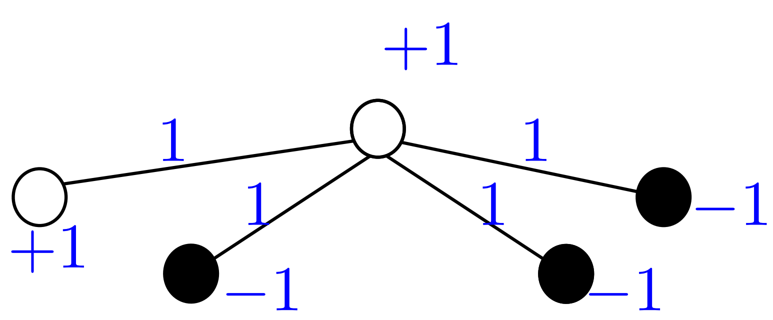
A model of *content-addressable associative memory*:
Hopfield networks [PNAS '84]
(≈ 8000 citations)

Each node v has initial state
$s_v \in \{-1, +1\}$

Dynamics.
Pick a node v at random and set

$$s_v \leftarrow \mathrm{sign}(\textstyle\sum_u s_u w_{u,v})$$

until changes don't occur anymore



A Model of Associative Memory

A model of *content-addressable associative memory*:
Hopfield networks [PNAS '84]
(≈ 8000 citations)

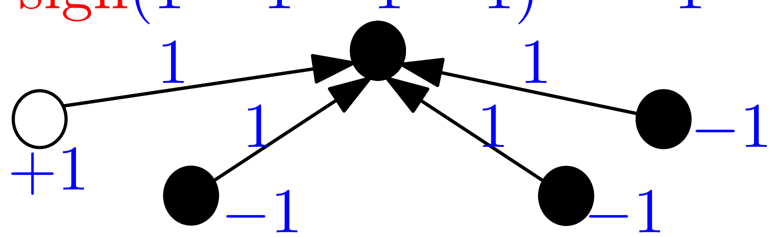
Each node v has initial state
$s_v \in \{-1, +1\}$

Dynamics.
Pick a node v at random and set

$$s_v \leftarrow \mathrm{sign}(\textstyle\sum_u s_u w_{u,v})$$

until changes don't occur anymore

$\mathrm{sign}(1 - 1 - 1 - 1) = -1$



A Model of Associative Memory

A model of *content-addressable associative memory*:
Hopfield networks [PNAS '84]
($\approx$ 8000 citations)

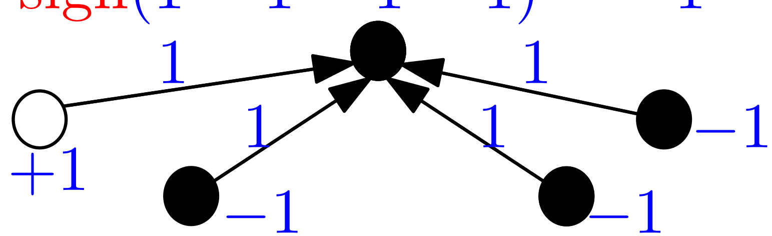
Each node v has initial state
$s_v \in \{-1, +1\}$

Dynamics.
Pick a node v at random and set

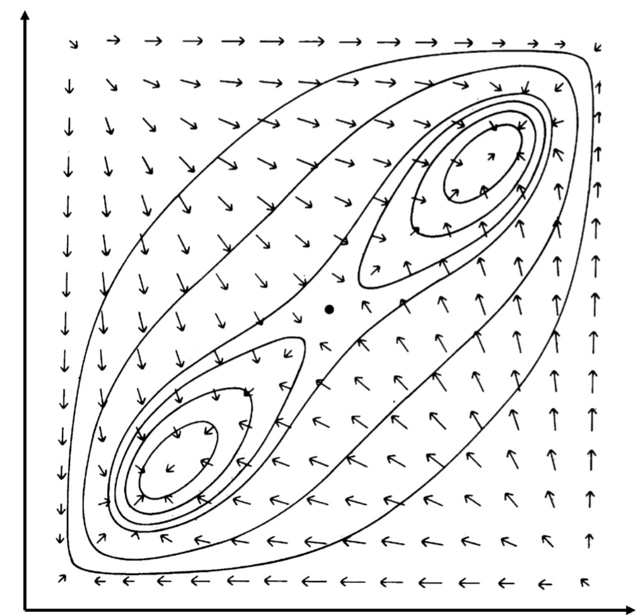
$$s_v \leftarrow \text{sign}(\textstyle\sum_u s_u w_{u,v})$$

until changes don't occur anymore

$$\text{sign}(1 - 1 - 1 - 1) = -1$$



Convergence to binary N-dimensional vectors $\{\mathbf{v}^{(i)}\}_i$



How to set weights $w_{u,v}$?

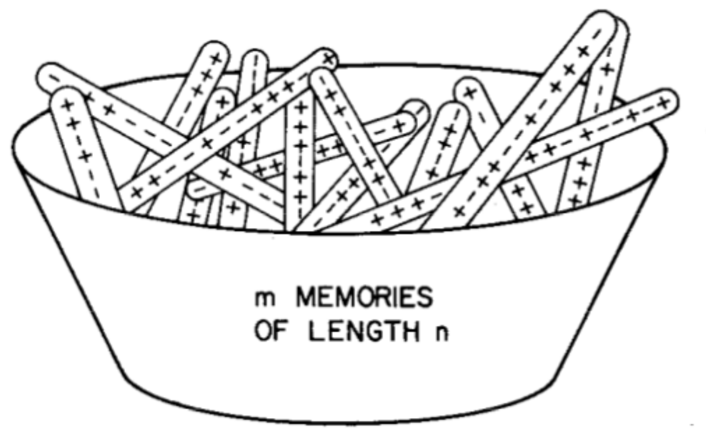
Hebbian learning ('49):

$$w_{i,j} = \frac{1}{N} \sum_k^N \mathbf{v}_i^{(k)} \mathbf{v}_j^{(k)}$$

"fire together, wire togheter"

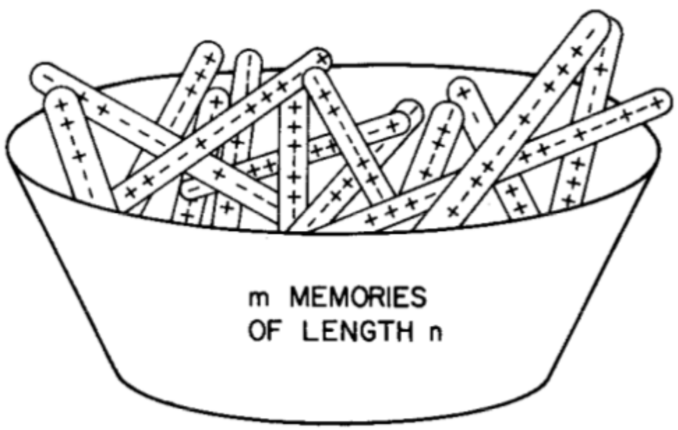
Capacity of Hopfield Networks

How many vectors before errors appear?



Capacity of Hopfield Networks

How many vectors before errors appear?

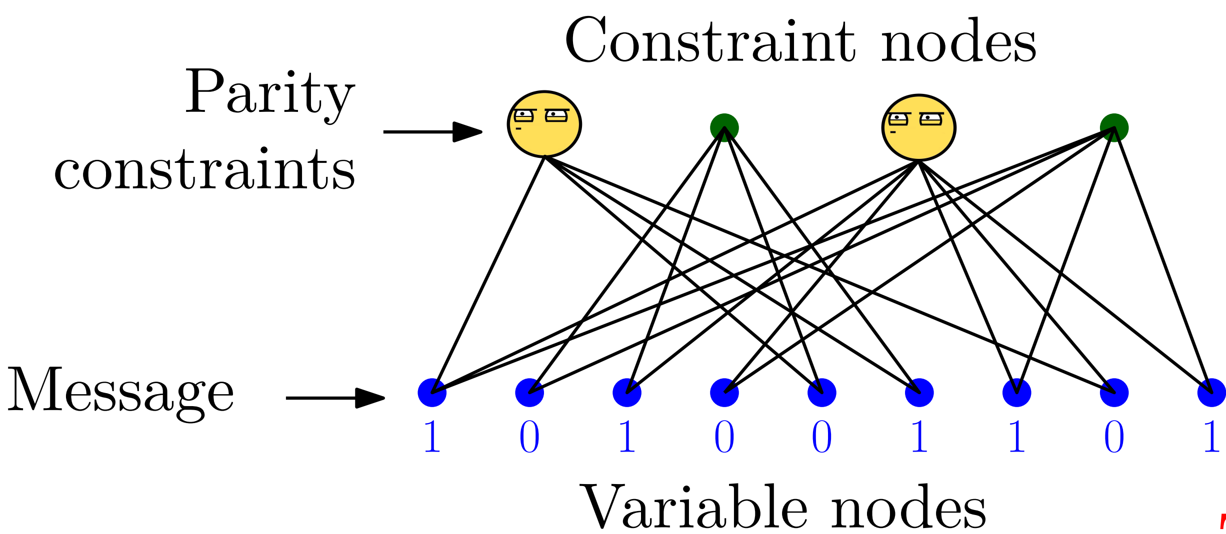


For random vectors, capacity is $\approx \sqrt{N}$
For structured patterns with other dynamics,
capacities are $\approx N$, $2^{(\sqrt{n})}$, $2^{\mathcal{O}\frac{n}{\log n}}$ (but not *robust*)

Problem. Exponential capacity $2^{\Omega(n)}$ in Hopfield networks
with structured patters?

From Expander Codes to Hopfield Networks

Expander Codes. [Sipser & Spielman '96]



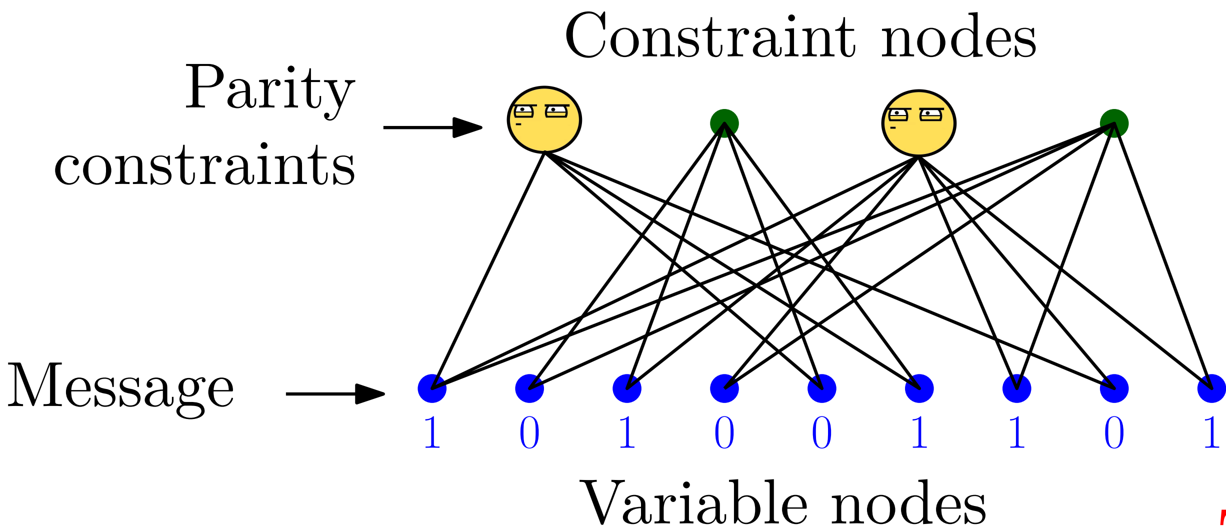
S set of $< \gamma N$ variables has $\geq (1-\epsilon)d|S|$ neighbors

Dynamics. Flip a variable with #unsatisfied $>$ satisfied neighbors

Thm. Dynamics recovers $2^{\Omega(n)}$ vectors, if $< \frac{\gamma}{2}N$ entries are flipped,

From Expander Codes to Hopfield Networks

Expander Codes. [Sipser & Spielman '96]



S set of $< \gamma N$ variables has $\geq (1-\epsilon)d|S|$ neighbors

Dynamics. Flip a variable with #unsatisfied > satisfied neighbors → **Thm.** Dynamics recovers $2^{\Omega(n)}$ vectors, if $< \frac{\gamma}{2}N$ entries are flipped,

[Chauduri & Fiete '18]

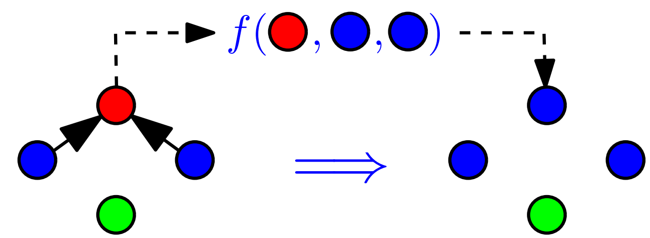
Exponential-Capacity Hopfield Network.

Constraint nodes $\rightarrow$ small Hopfield networks.

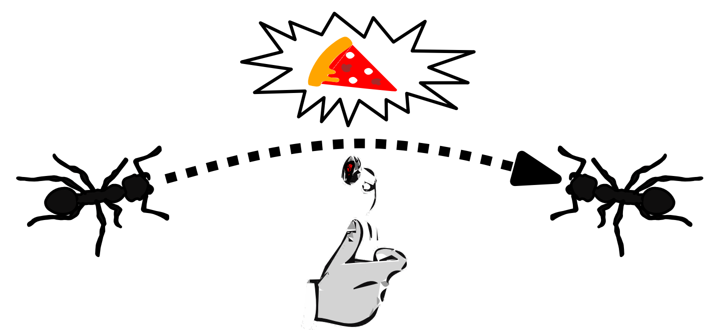
Dynamics $\rightarrow$ pick a random node and flip it to majority.

Three Messages

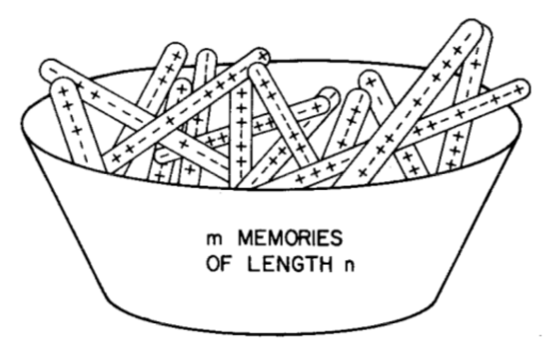
- **Computational Dynamics.**
  Achieving simplicity in randomized distributed algorithms.



- **Biological Distributed Algorithms.**
  Investigating Biology through the algorithmic lens (Natural Algorithms).



- **Theoretical Neuroscience.**
  Investigating Neuroscience through the algorithmic lens.



Thank You!