Spontaneous Emergence of Computation in Network Cascades

Galen Wilkerson[*+], *Sotiris Moschoyiannis*[*] *and Henrik Jeldtoft Jensen*[+]
University of Surrey[*] *and Imperial College*[+], *London*
Correspondence: g.wilkerson21@imperial.ac.uk

Abstract:
*Neuronal network computation and computation by avalanche supporting networks are of interest to the fields of physics, computer science (computation theory as well as statistical or machine learning) and neuroscience. Here we show that computation of complex Boolean functions arises spontaneously in threshold networks as a function of connectivity and antagonism (inhibition), computed by logic automata (*motifs*) in the form of* computational cascades. *We explain the emergent inverse relationship between the computational complexity of the motifs and their rank-ordering by function probabilities due to motifs, and its relationship to symmetry in function space. We also show that the optimal fraction of inhibition observed here supports results in computational neuroscience, relating to optimal information processing.*

The Team

Henrik Jeldtoft Jensen
Centre for Complexity Science
Mathematics Department
Imperial College London
https://www.ma.ic.ac.uk/~hjjens/

Galen Wilkerson
g.wilkerson21@imperial.ac.uk

Sotiris Moschoyiannis
Department of Computer Science
University of Surrey
https://www.surrey.ac.uk/people/sotiris-moschoyiannis

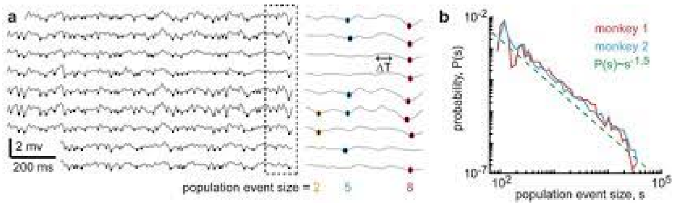
Outline

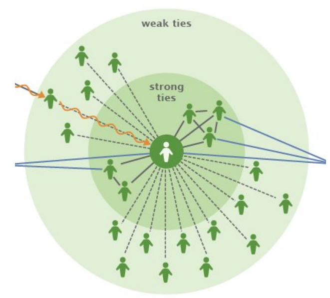
- *Cascades and Computation*
  - *Cascades are universal, ubiquitous*
  - *Linear Threshold Model (LTM)*
  - *LTM computes logic via sub-networks (motifs)*
  - *Antagonistic cascades compute universal basis*
- *Statistics of Attractors in Boolean space*
  - *Rank-ordering according to function complexity*
  - *Relation between:*
    - *function symmetries, complexity, and frequency*

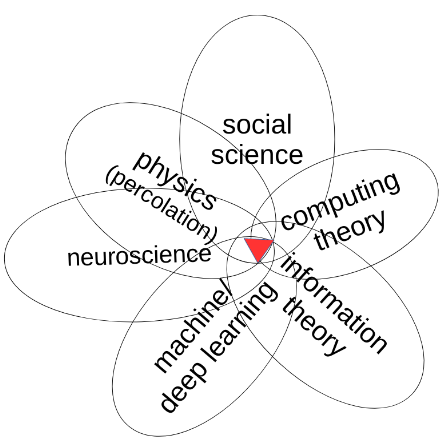
Cascades are ubiquitous and fundamental

- *cascades seen in many systems and domains*
  - *brain – neuronal avalanches*
  - *social networks*
    - *epidemics !!*
    - *information diffusion,*
    - *viral marketing (CS: 'influence maximization')*
  - *physical, chemical systems*
- *percolation – basic branching process*
- *simplest possible perturbation of system → **point-bit***
- *interaction of system elements → **interaction-bit***
- *elements of any system that interact → **network***
- *interactions of perturbation patterns → **Boolean functions***

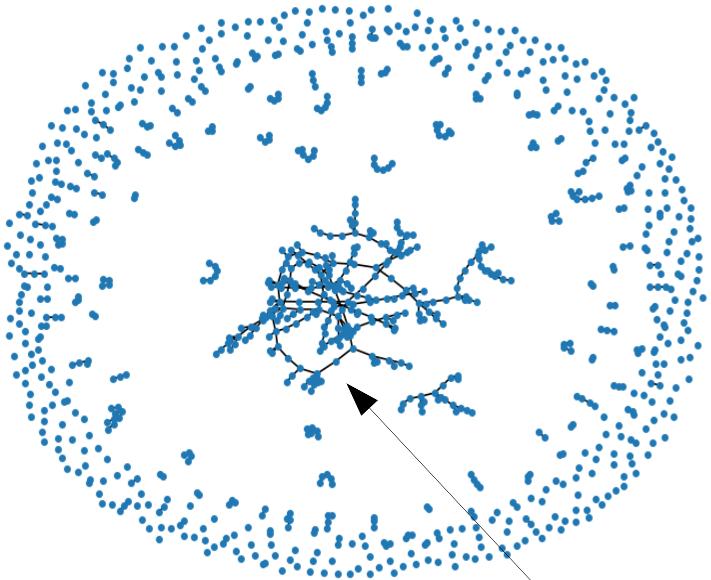
- *basic research into physics and mathematics of information in systems* [Christensen 2005, Watts 2002].



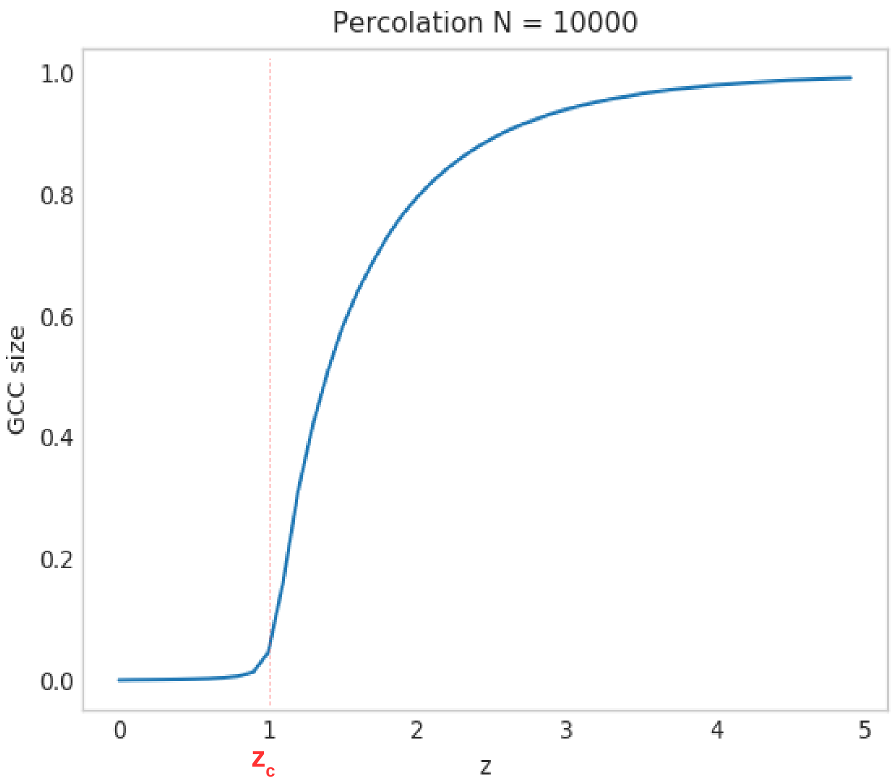




Recall Percolation



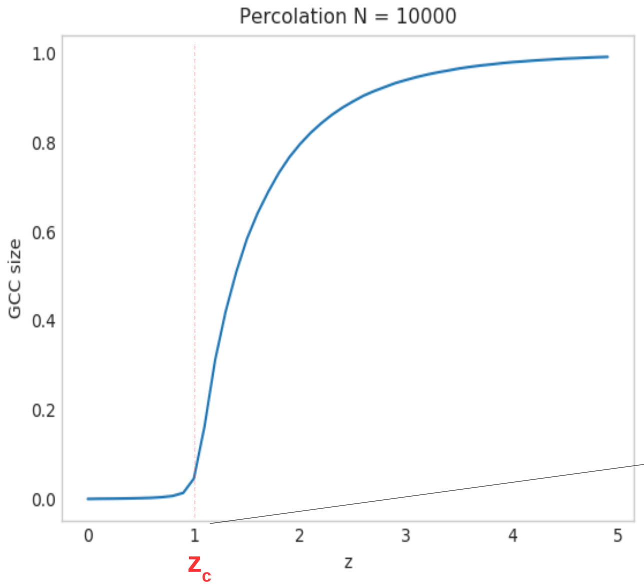
at critical connectivity,
a giant connected component (GCC)
suddenly appears.

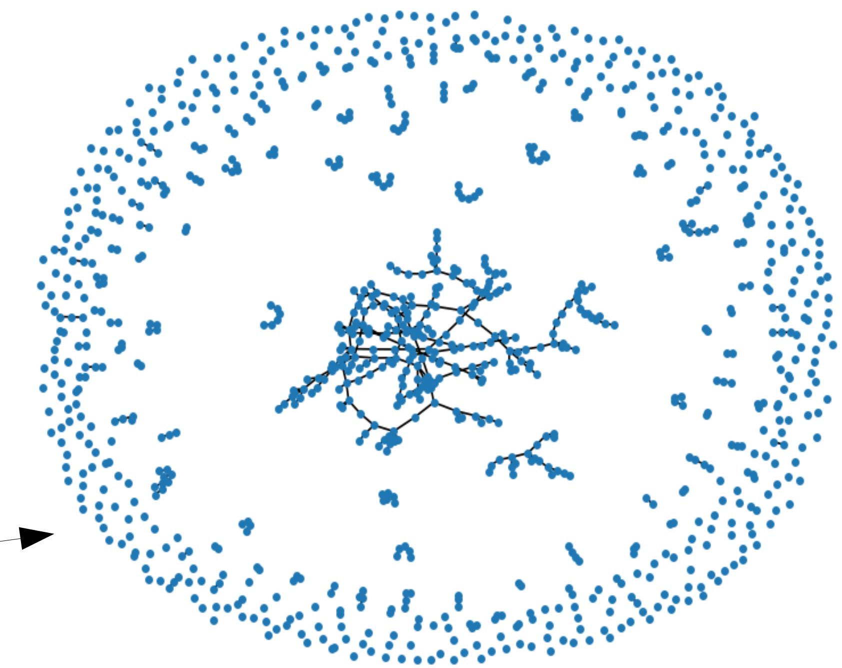


for random (Erdos-Renyi-Gilbert) graphs, over many trials,
the size of largest connected component (GCC)
explodes at critical average degree (z_c)

(recall $z \approx Np$ *)*

Recall Percolation

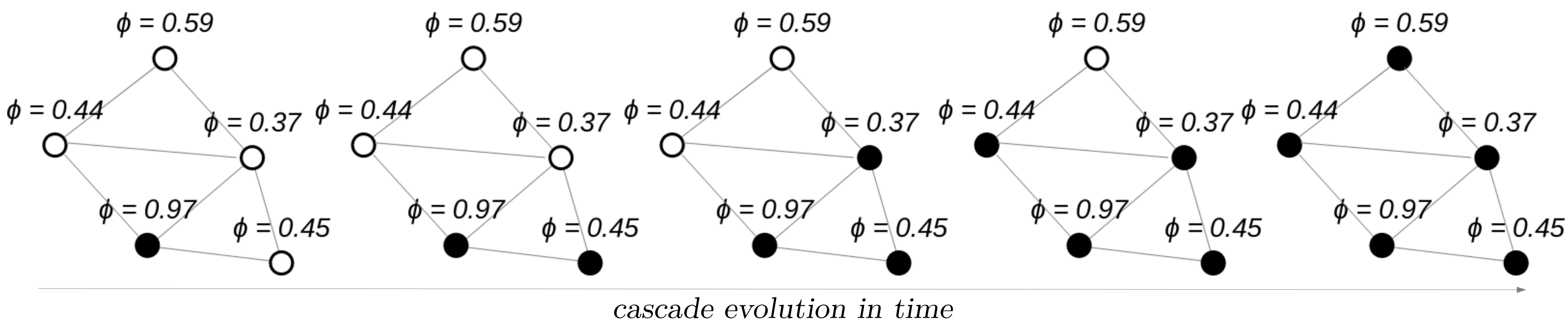




At critical degree z_c, there seems to be scale invariance of component size distribution: $p(x) \sim cx^{-\beta}$

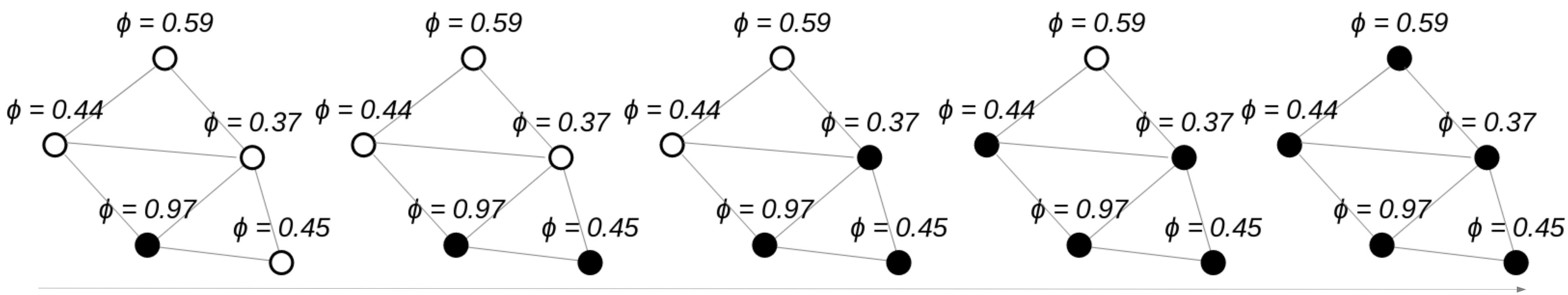
[Hofstad]

Linear Threshold Model (LTM)



- *create Erdos-Renyi random graph G(N, p)*
- *assign each node random threshold* $\boldsymbol{\phi} \sim U[0, 1]$
- *set all nodes unlabelled*
- *label a few seeds*
- *randomly check each unlabelled node* $\mathbf{u}$ *until no change:*
    - *If fraction of* $\mathbf{u}$*'s neighbors labelled* $\geq \boldsymbol{\phi} \rightarrow$ *label* $\mathbf{u}$ *(1)*

Linear Threshold Model (LTM)

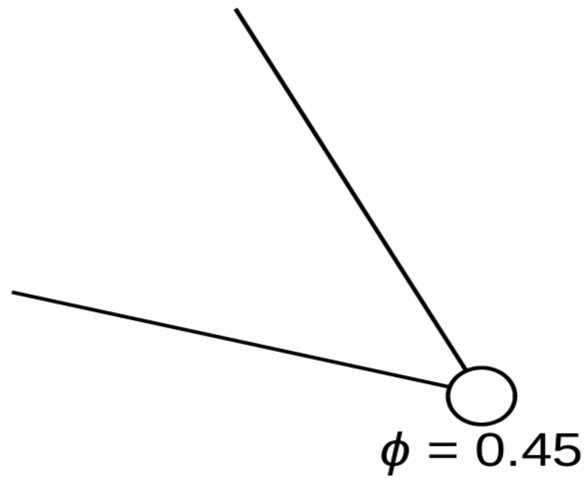


cascade evolution in time

- *create Erdos-Renyi random graph G(N, p)*
- *assign each node random threshold* $\boldsymbol{\phi} \sim U[0, 1]$
- *set all nodes unlabelled*
- *label a few seeds*
- *randomly check each unlabelled node* **u** *until no change:*
    - *If fraction of* **u***'s neighbors labelled* $\geq \boldsymbol{\phi} \rightarrow$ *label* **u** **(1)**

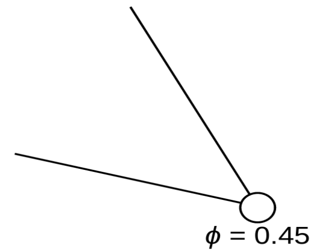
(can be written in vector form à la deep learning) $\bar{L}_{t+1} \leftarrow \bar{L}_t \vee \mathcal{H}(A \cdot \bar{L}_t - (A \cdot \bar{1}) \cdot \bar{\phi})$

LTM critical cascades are percolation

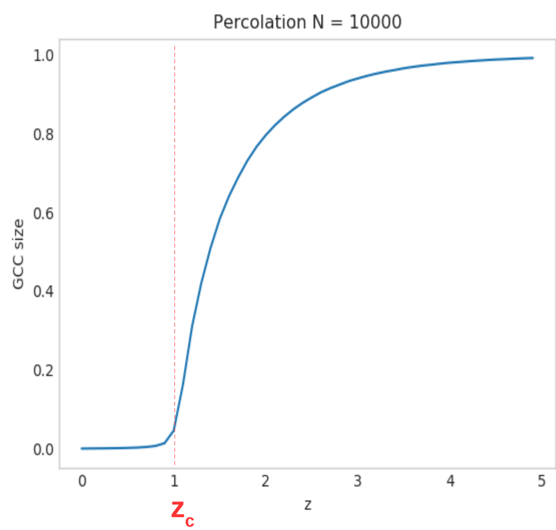


in LTM, vulnerable nodes require only 1 labelled neighbor to become labelled

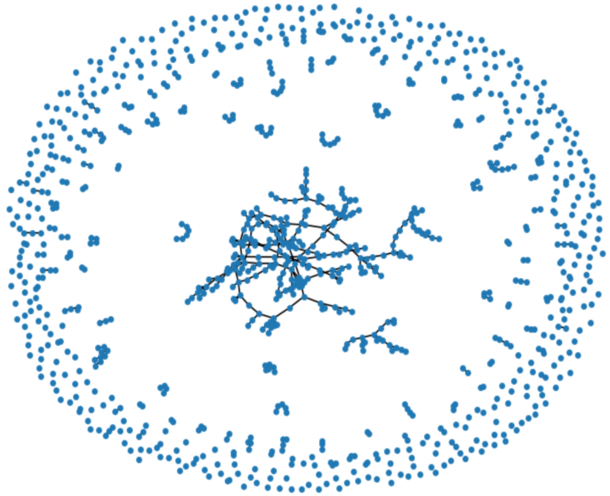
LTM critical cascades are percolation



Cascades are sudden increase (percolation) of ***giant component (GCC) of vulnerable nodes*** *[Watts, 2002].*



for random graphs, over many trials, the size of largest connected component (GCC) explodes at critical average degree (z_c)

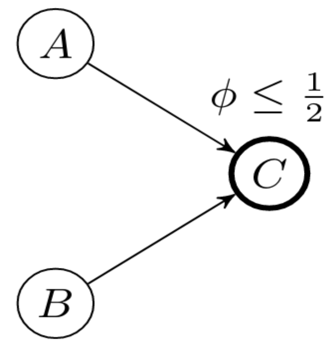


At critical degree z_c, there is scale invariance of size distribution: $p(x) \sim cx^{-\beta}$

LTM cascades compute (monotone) logic

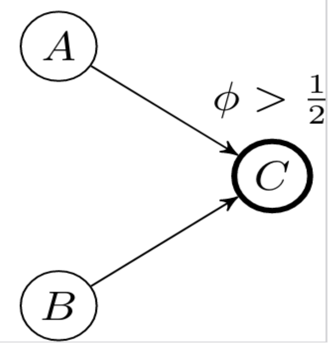
A	B	ν	C
0	0	0/2	0
0	1	1/2	1
1	0	1/2	1
1	1	2/2	1

$\equiv$ **OR**



A	B	ν	C
0	0	0/2	0
0	1	1/2	0
1	0	1/2	0
1	1	2/2	1

$\equiv$ **AND**



LTM computes logic, depending on threshold and input values.
These are logical automata or logic motifs*.*
[Von Neumann 1956, Milo 2002]

(note similarity to thresholding nodes in McCulloch-Pitts
and other neural networks)

Obtain Antagonistic LTM (ALTM) by reversing threshold rule (1):

- *create Erdos-Renyi random graph G(N, p)*
- *assign each node random threshold* $\boldsymbol{\phi} \sim U[0, 1]$
- *set all nodes unlabelled*
- *label a few seeds*
- *randomly check each unlabelled node* $\boldsymbol{u}$ *until no change:*

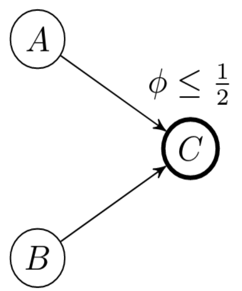
~~- If fraction of~~ $\cancel{\boldsymbol{u}}$~~'s neighbors labelled~~ $\cancel{\geq \boldsymbol{\phi} \rightarrow}$ ~~label~~ $\cancel{\boldsymbol{u}}$ ~~(1)~~

if fraction of u's neighbors labelled $< \boldsymbol{\phi} \rightarrow$ *label u*

Obtain Antagonistic LTM (ALTM) by reversing threshold rule (1):

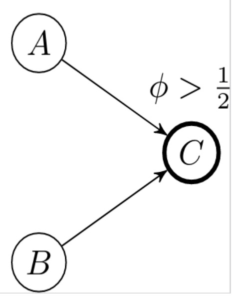
A	B	ν	C
0	0	0/2	0
0	1	1/2	1
1	0	1/2	1
1	1	2/2	1

$\equiv$ **OR**



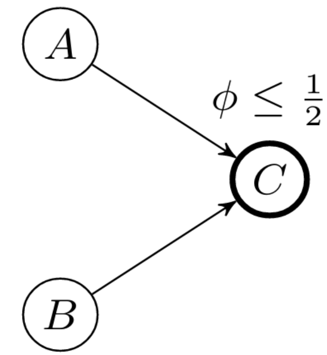
A	B	ν	C
0	0	0/2	0
0	1	1/2	0
1	0	1/2	0
1	1	2/2	1

$\equiv$ **AND**



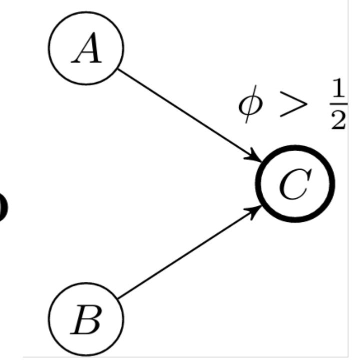
A	B	ν	C
0	0	0/2	1
0	1	1/2	0
1	0	1/2	0
1	1	2/2	0

$\equiv$ **NOR**



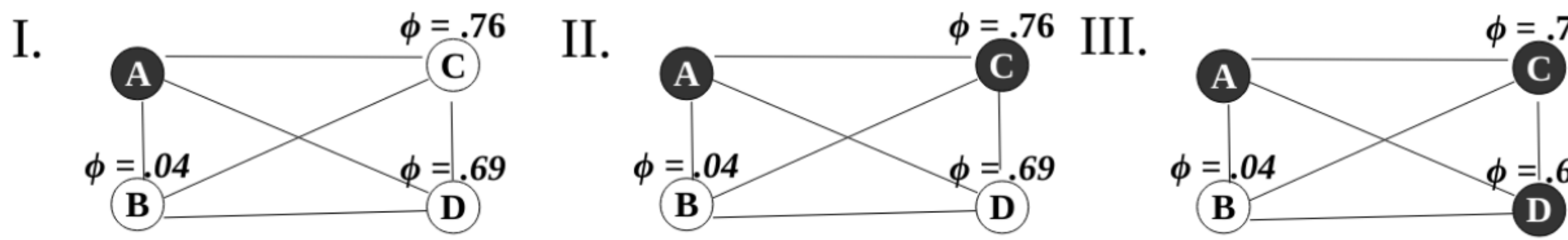
A	B	ν	C
0	0	0/2	1
0	1	1/2	1
1	0	1/2	1
1	1	2/2	0

$\equiv$ **NAND**

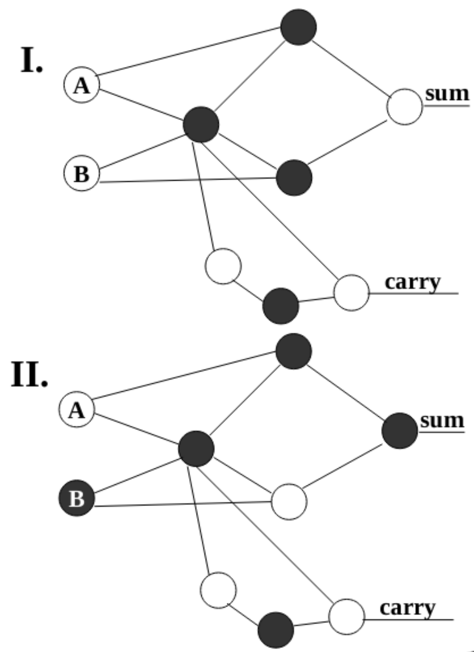


ALTM computes NAND, NOR

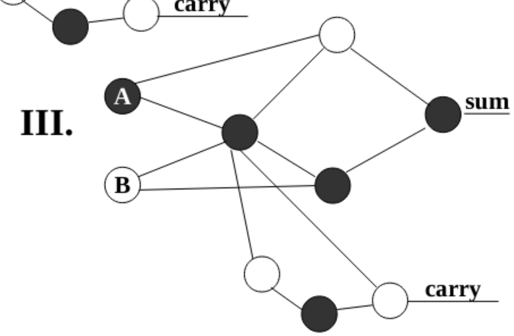
Antagonism (ALTM) also undergoes cascades

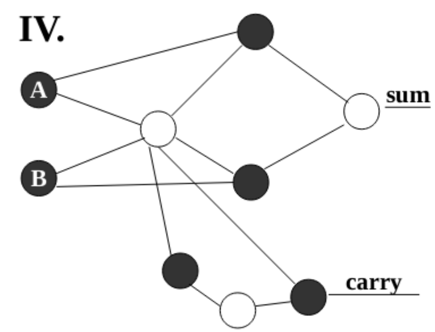


Antagonism (ALTM) Computes Universal Boolean Logic



truth table				
	Input		Output	
	A	B	Sum	Carry
I.	0	0	0	0
II.	0	1	1	0
III.	1	0	1	0
IV.	1	1	0	1

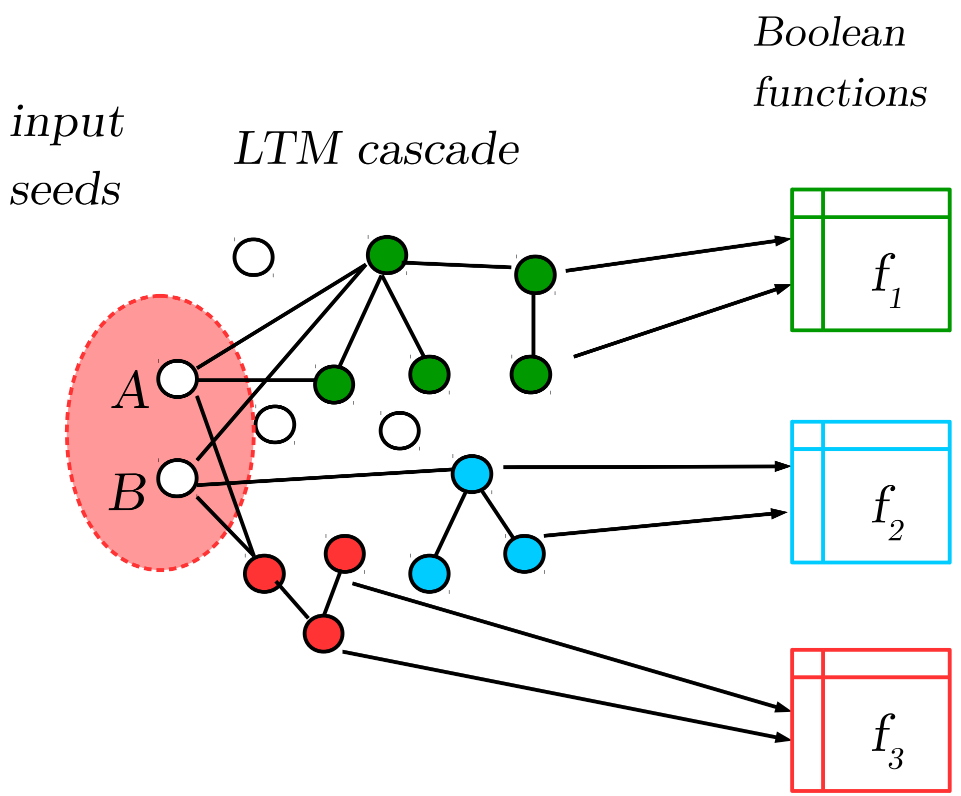




ALTM computes NAND, NOR, which when composed can compute any Boolean function (universal basis)

e.g. Half-adder

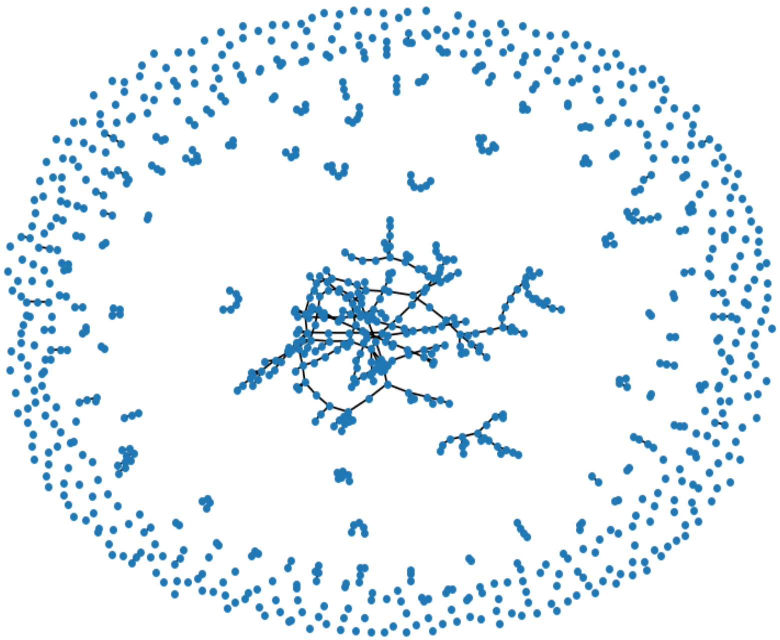
LTM computes functions on seed perturbations



Simulation:

- *Create LTM, freeze edges E and thresholds* ϕ
- *Run cascades on all possible values of A and B,*
- *each node in the LTM computes some Boolean function on the seed nodes.*

LTM computes functions on seeds



"It has been pointed out by A. M. Turing in 1937 and by W. S. McCulloch and W. Pitts in 1943 that effectively constructive logics, that is, intuitionistic logics, can be best studied in terms of ***automata***. Thus *logical propositions can be represented as electrical networks or (idealized) nervous systems*. ***Whereas logical propositions are built up by combining certain primitive symbols, networks are formed by connecting basic components, such as relays in electrical circuits and neurons in the nervous system.***"

– J. von Neumann, 1956

sub-networks are like logic functions, imported by connection

$\rightarrow$ *logical automata*

network patterns = logic motifs

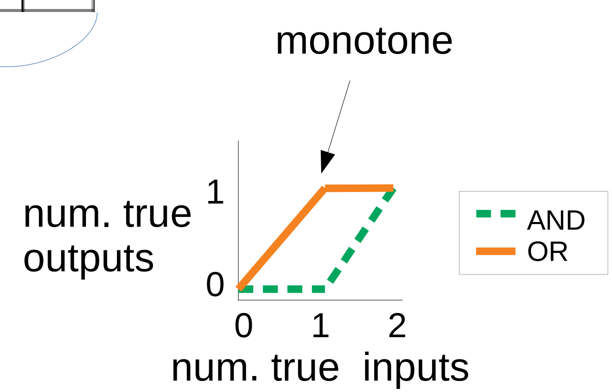
The LTM only computes monotone Boolean functions

A	B	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

inputs

unique functions

The LTM computes ***monotone***
(non-decreasing in number of true inputs)
Boolean functions ***(green, above)***



Boolean function space grows very large in the inputs k

A	B	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

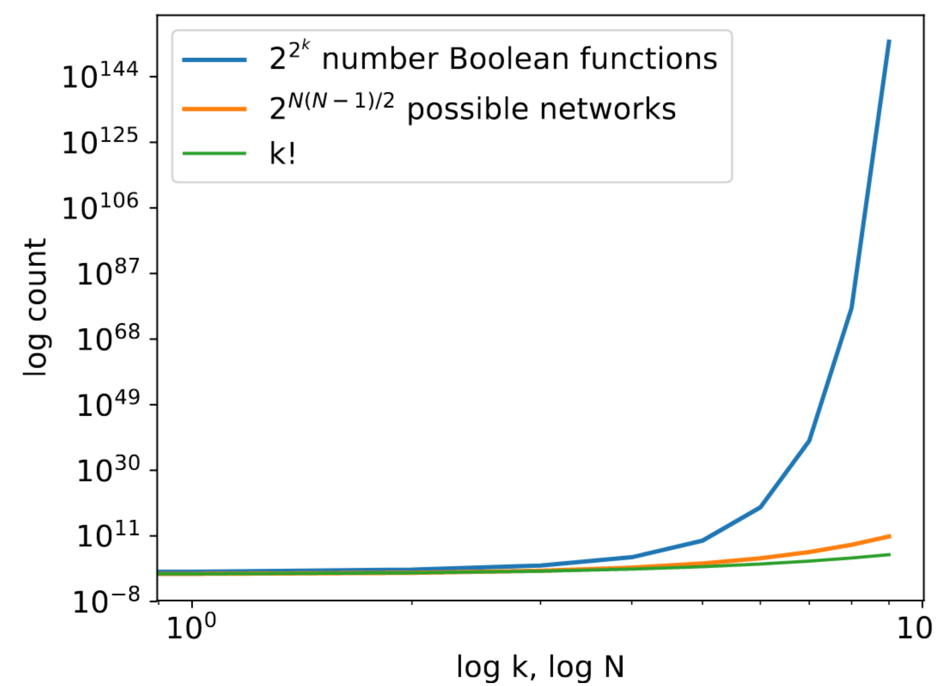
$k = 2$ inputs

$2^{2^k} = 16$ *unique functions*

The number of Boolean functions 2^{2^k}

The number of networks $2^{N(N-1)/2}$

How do these combinatorial spaces interact?

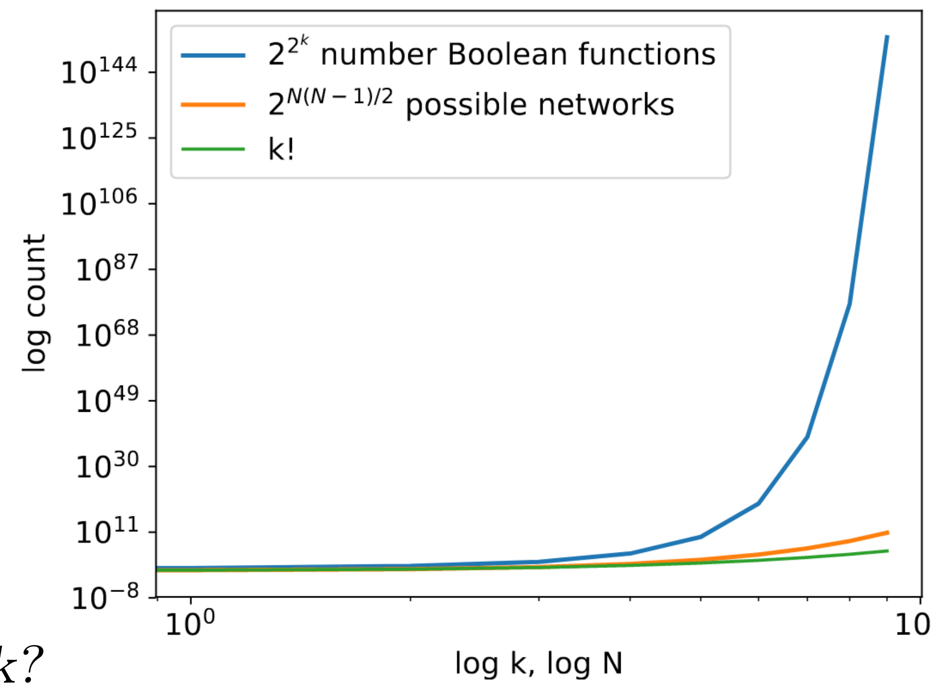


Boolean function space grows very large in the inputs k

A	B	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

$k = 2$ inputs

$2^{2^k} = 16$ unique functions



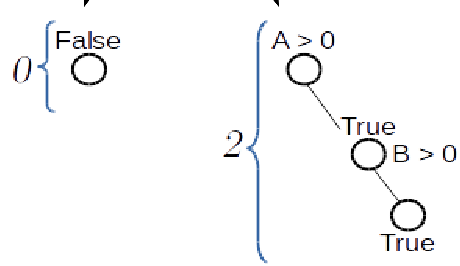
What if you want one particular function to dominate (or even occur) in the network?

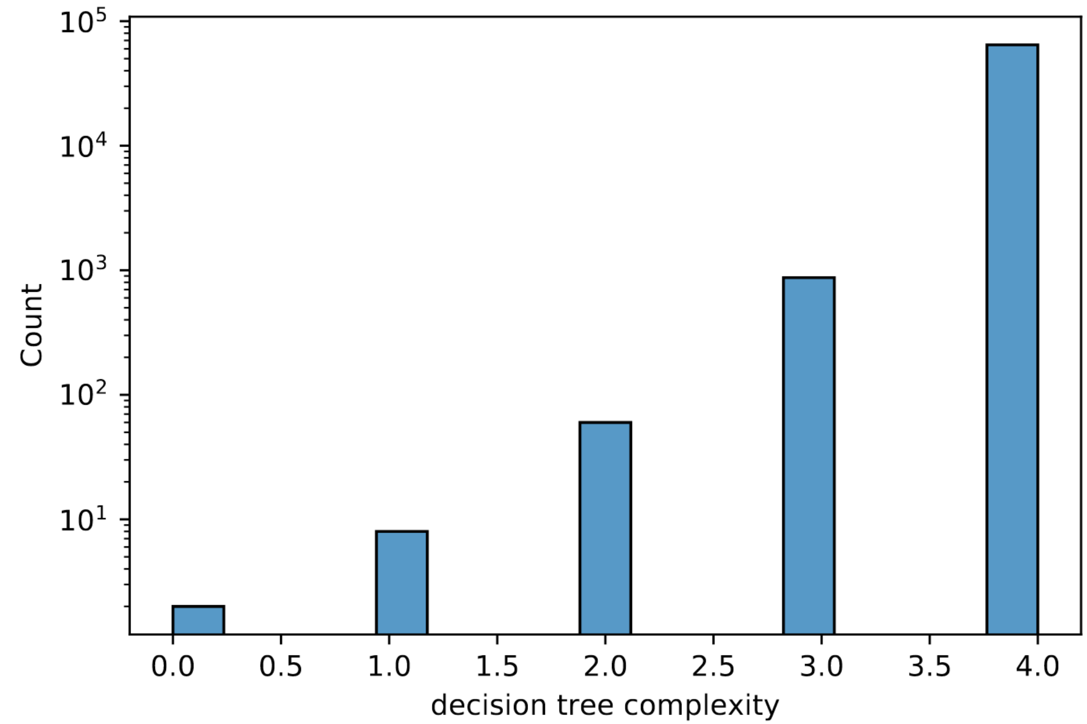
Available (truth table)
Boolean functions tend to be complex

A	B	0	1
0	0	0	0
0	1	0	0
1	0	0	0
1	1	0	1

...

Decision tree complexity

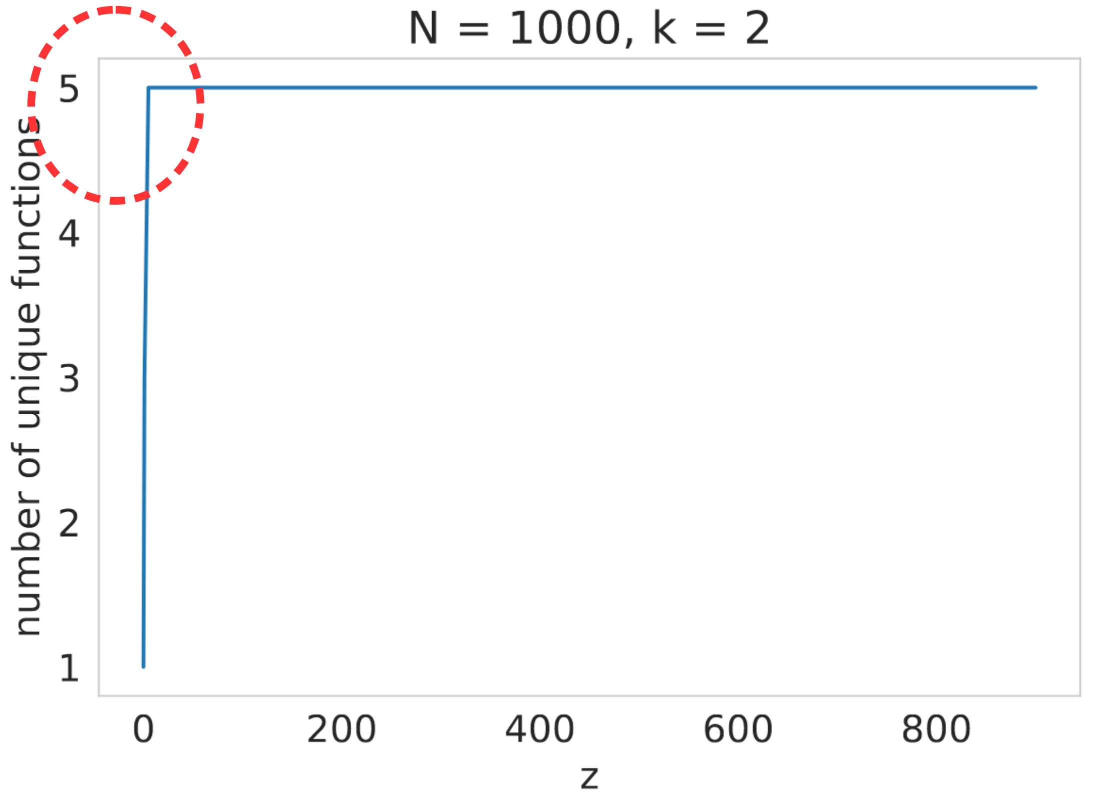




complexity histogram
for $k = 4$ inputs

Decision tree complexity (depth of decision tree)
of all available Boolean functions is hyper-exponentially distributed.

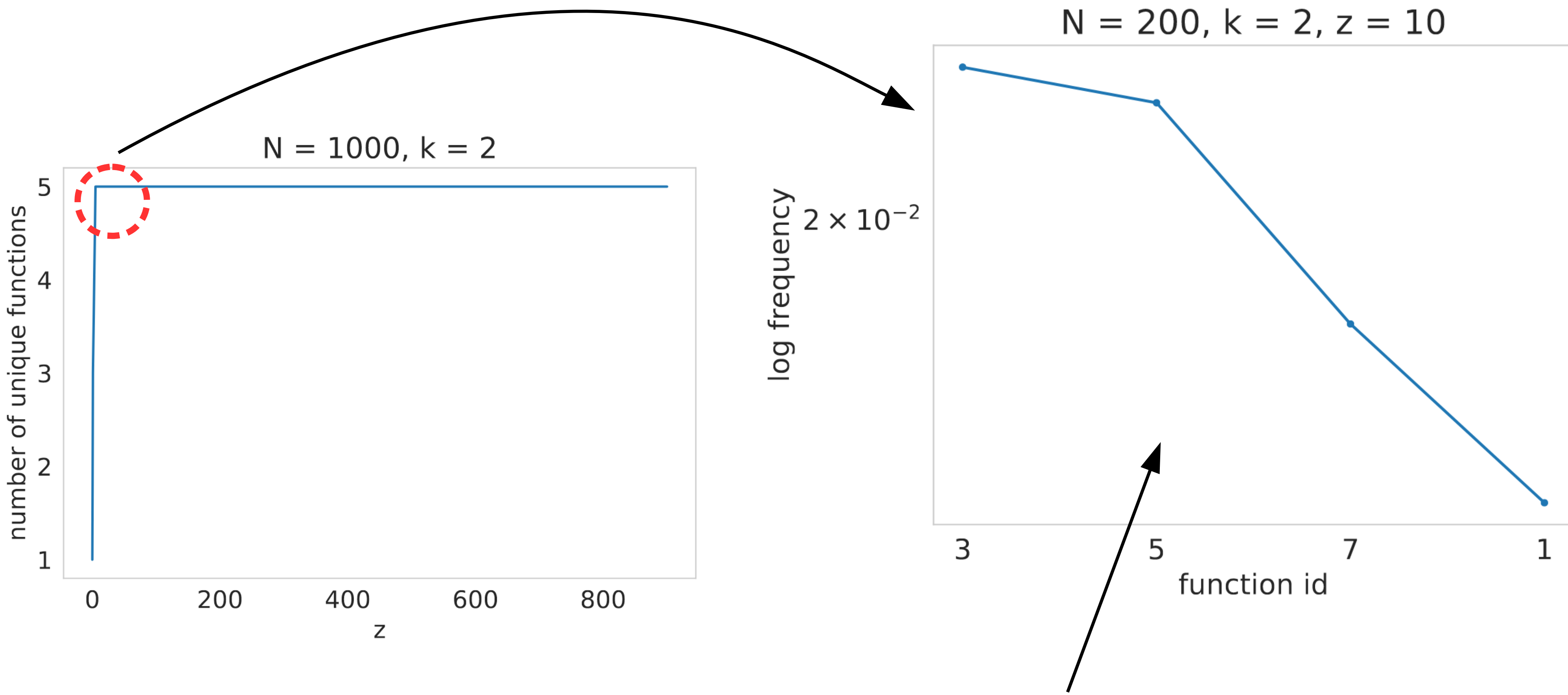
Emergence of complex Boolean functions



'Toy' example ($k = 2 \rightarrow 5$ monotone functions available)

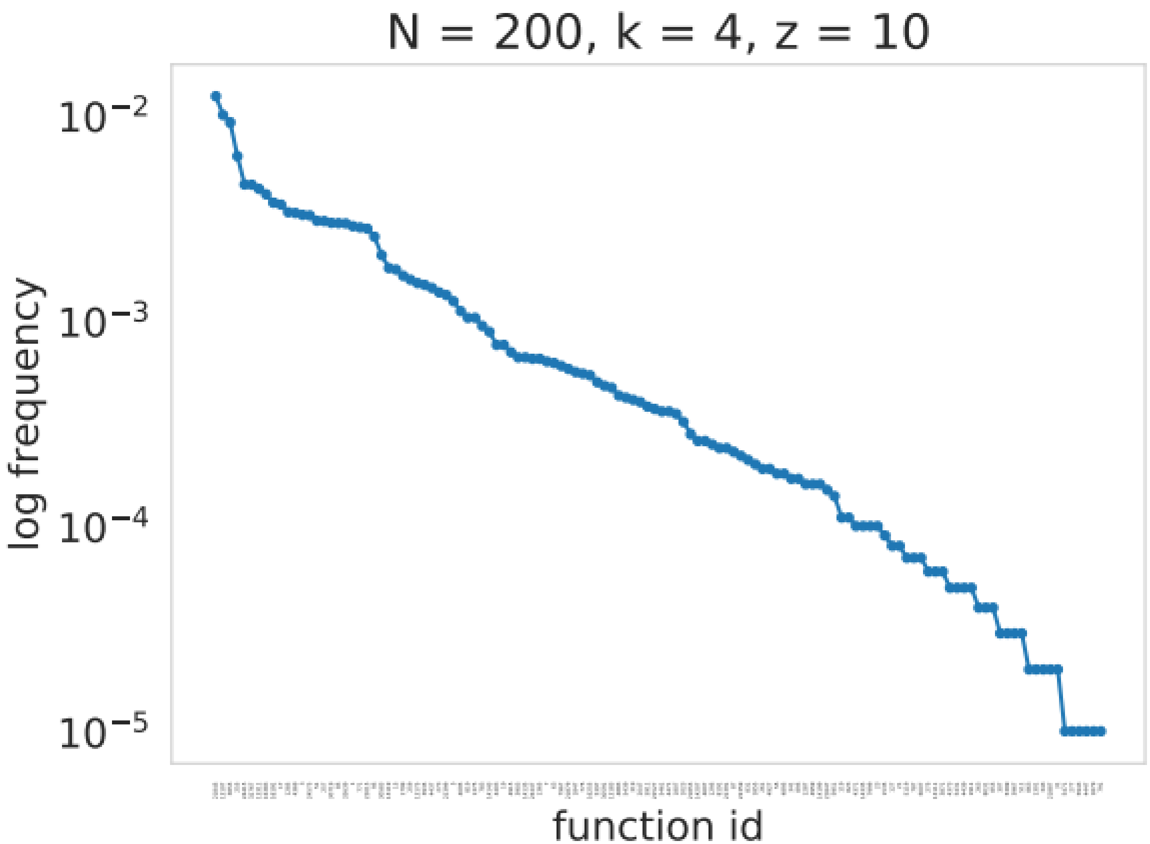
Number of unique functions computed in simulation vs. average degree (z), undergoes a sudden increase at z_c.

Emergence of complex Boolean functions



apparent decreasing exponential
of function frequency

Emergence of complex Boolean functions

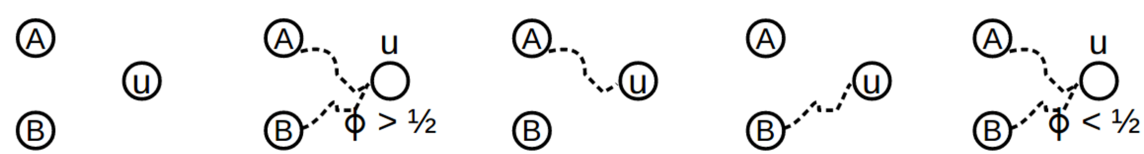


apparent decreasing exponential relation:
167 unique non-zero functions observed (200 trials)

Boolean function probabilities in the LTM

A	B	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

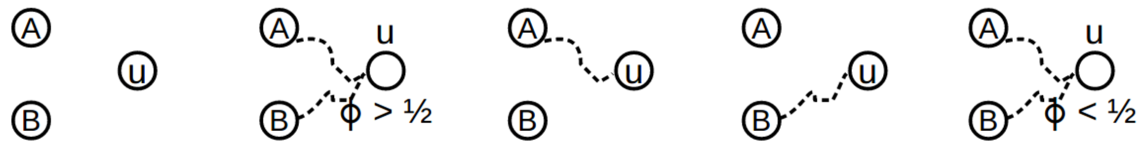
$f_0(a,b) = 0;$ $f_1(a,b) = a \text{ AND } b;$ $f_3(a,b) = a;$ $f_5(a,b) = b;$ $f_7(a,b) = a \text{ OR } b$



simplest sufficient networks to calculate the monotonic increasing functions

Boolean function probabilities in the LTM

$f_0(a,b) = 0;\quad f_1(a,b) = a \text{ AND } b;\quad f_3(a,b) = a;\quad f_5(a,b) = b;\quad f_7(a,b) = a \text{ OR } b$



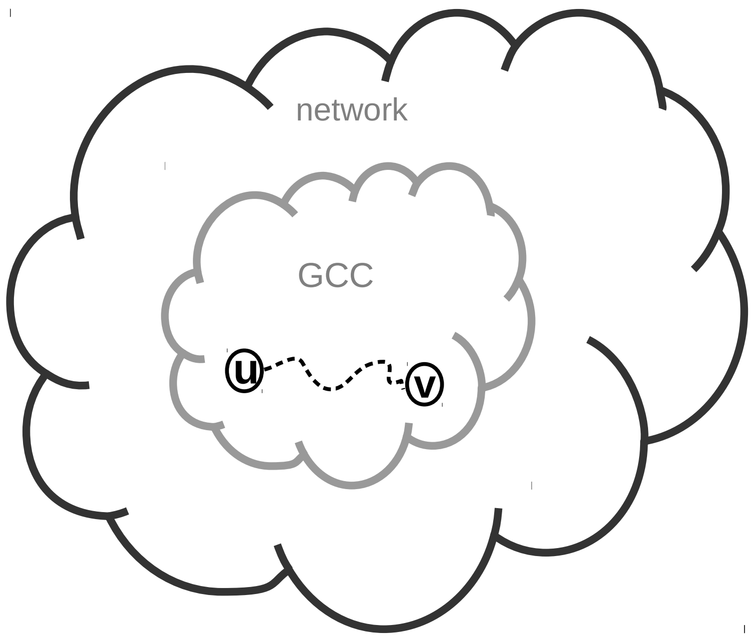
simplest sufficient networks to calculate the monotonic increasing functions

proportionality of monotone function probability:

$$\begin{cases} p(f_0) \propto (1 - p_{path})^2 \\ p(f_1) \propto p_{path}^2 \\ p(f_3) \propto p_{path} \\ p(f_5) \propto p_{path} \\ p(f_7) \propto p_{path}^2 \end{cases}$$

for p_{path} the probability of a path between two nodes

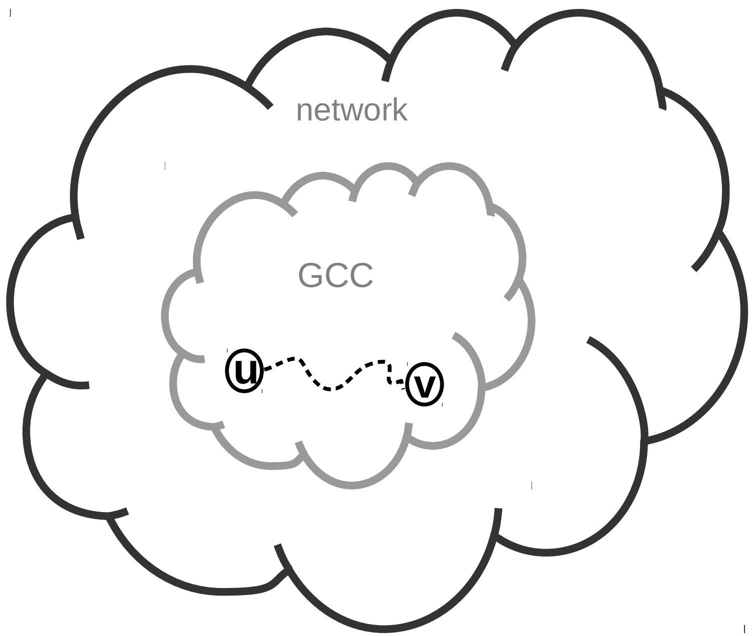
$p(u,v)_{path}$ can be approximated by $p(u,v)_{inGCC}$



if there is a path(u,v),
then u and v are very likely in the
Giant Connected Component (GCC)

as $N \to \infty$ *this is exactly true*

$p(u,v)_{path}$ can be approximated by $p(u,v)_{inGCC}$



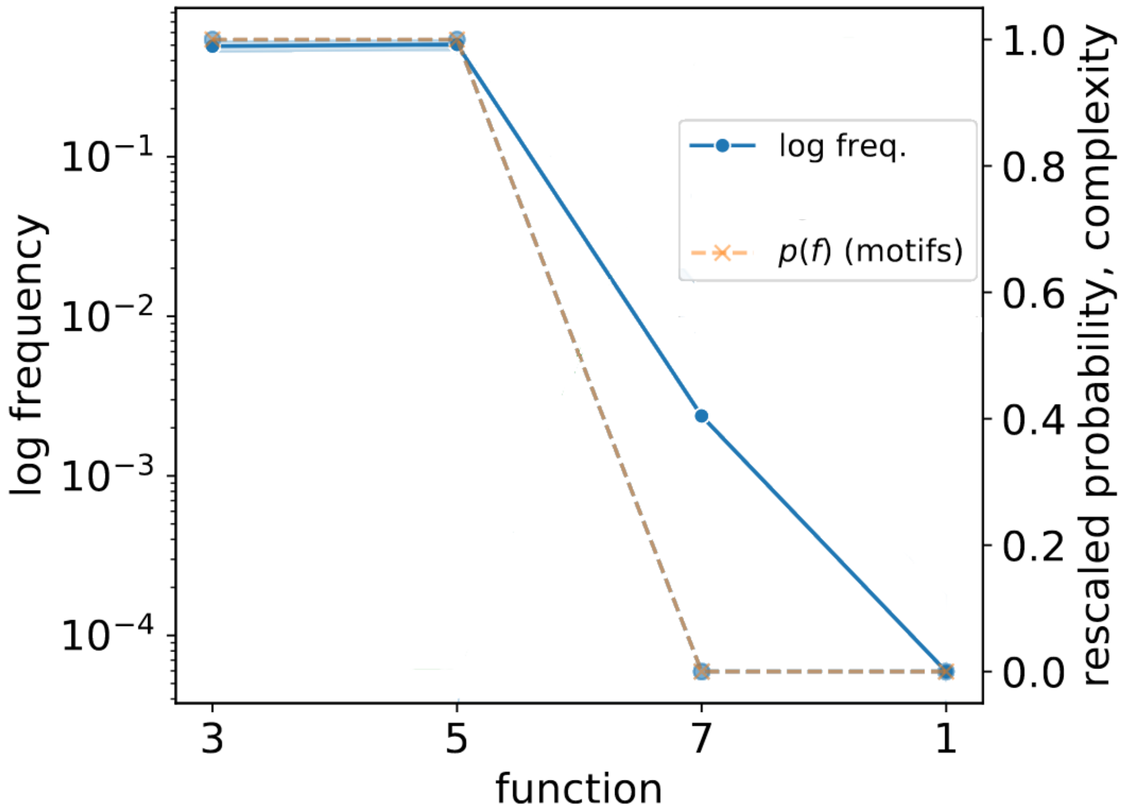
if there is a path(u,v),
then u and v are very likely in the
Giant Connected Component (GCC)

as $N \to \infty$ *this is exactly true*

$$p_{gcc} = v = 1 - e^{-zv}$$

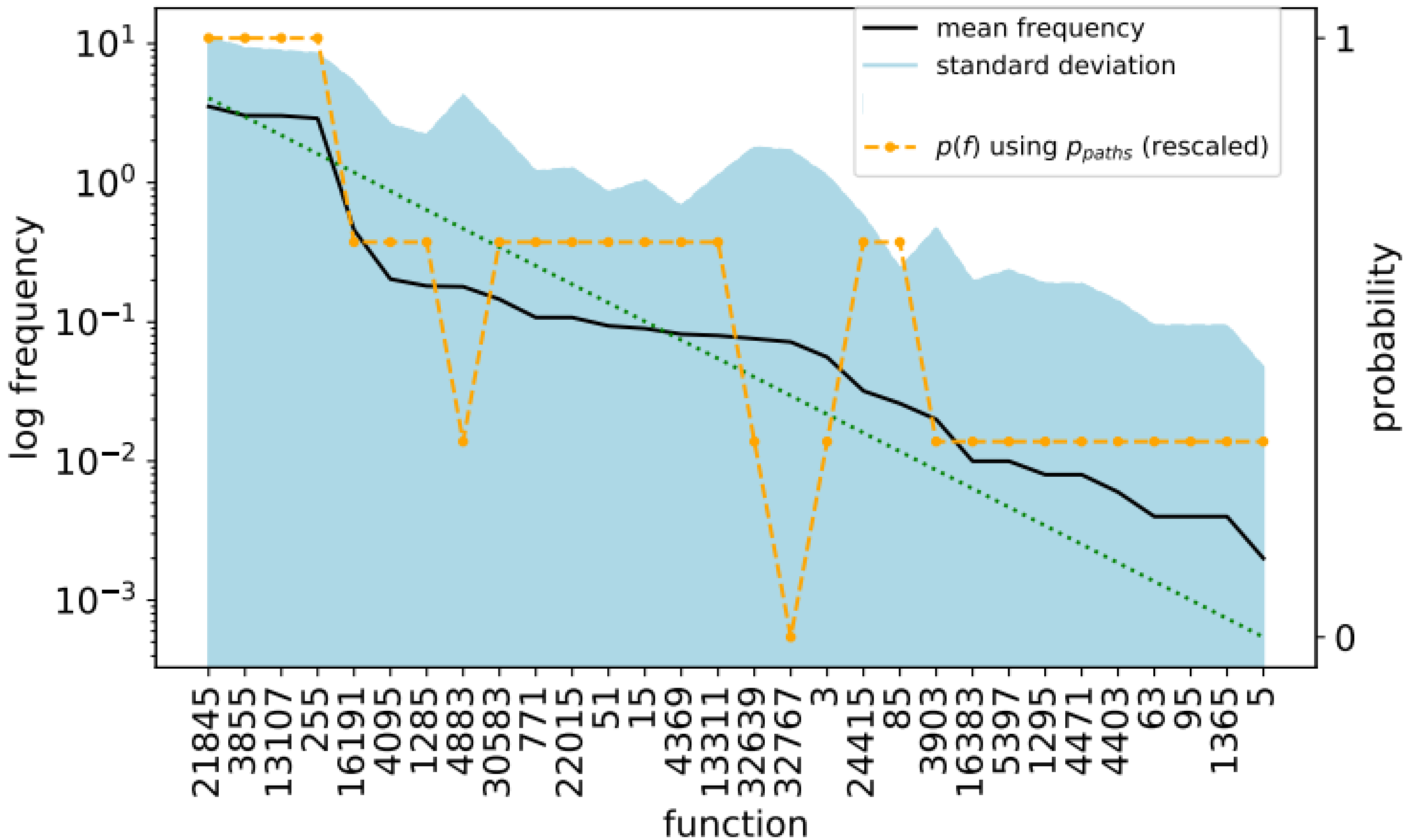
From [Newman 2018] we
have the recursion relation.
Can solve numerically.

Apparent agreement between prediction and frequencies



The frequency corresponds to the probability from motifs.

Apparent agreement between prediction and frequencies

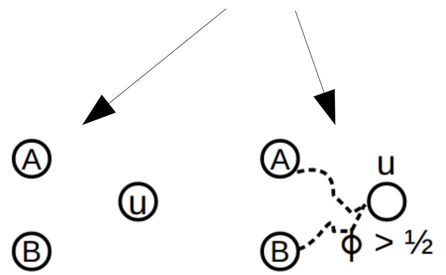


The frequency corresponds to the probability from motifs,
(Pearson correlation 0.74)
and is fit by an exponential with $R^2 = 0.88$

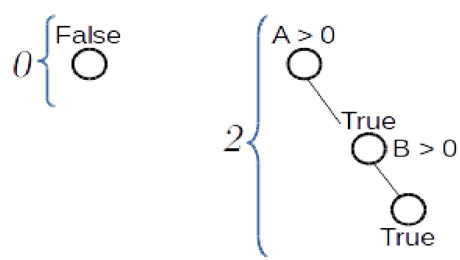
Symmetry and Function Frequency

A	B	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

number of paths required for simplest graph



decision tree complexity (depth of decision tree)



For monotone functions, the number of required paths is the decision tree complexity.

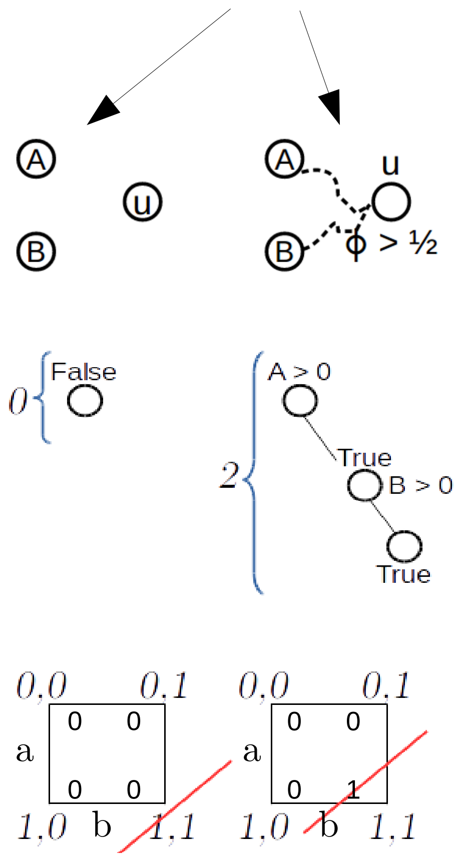
Symmetry and Function Frequency

A	B	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

number of paths required for simplest graph

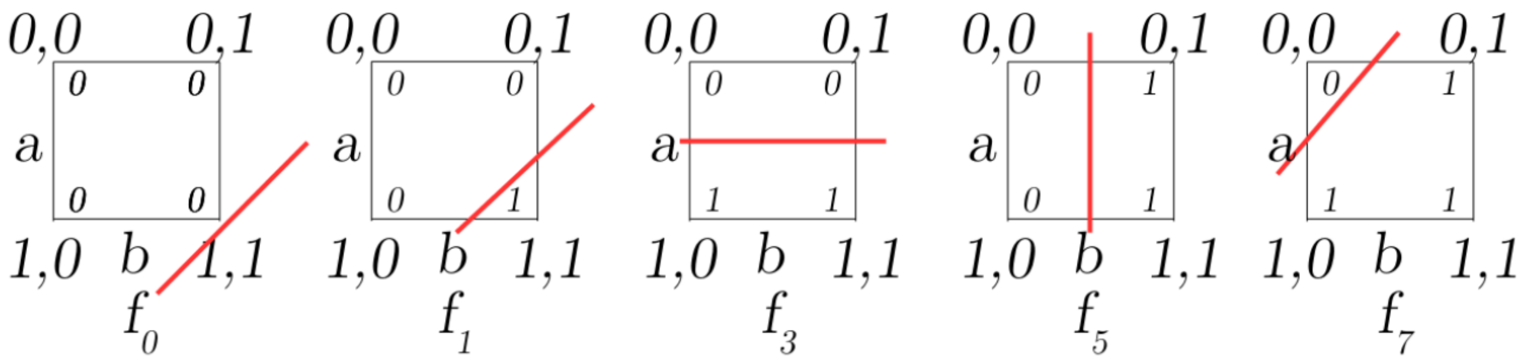
decision tree complexity (depth of decision tree)

reflection symmetries of Hamming Cube



The decision tree complexity is inversely related to the axial reflection symmetries of the Hamming cube.

Hamming Asymmetries →
Decision Tree Complexity



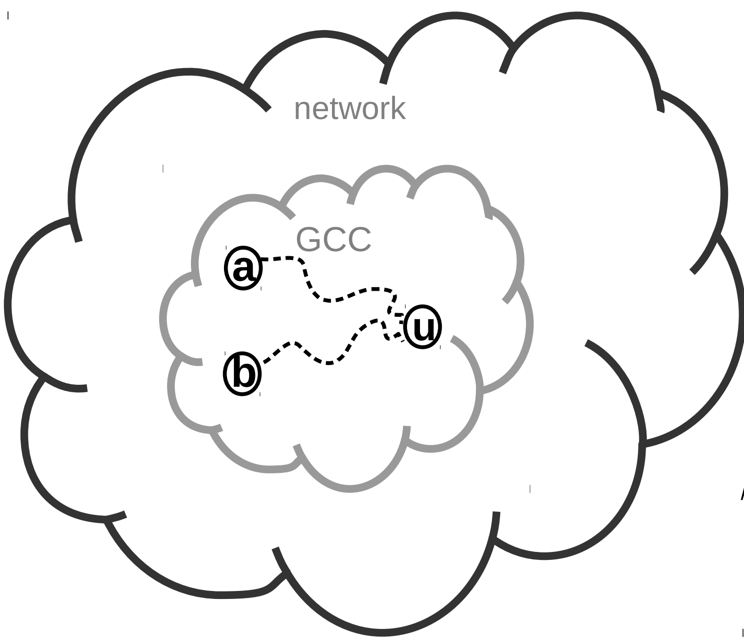
That is, if a function's Hamming cube representation is constant along an axis, it is independent of that axis, giving us

$$C = D - R,$$

where C is decision tree ***c****omplexity,*
*D is the number of axes (****d****imension) of the Hamming cube,*
and R is the number of congruent axial ***r****eflections of the Hamming cube.*

complexity → probability

- *Near criticality, the graph is sparsely connected,*
- *therefore tree-like (no loops)*
- *In a tree,* $N = |E| + 1 \rightarrow N_{gcc} = C + 1$
- *Therefore, the number of nodes in the GCC is one more than the number of paths*



$$p(f) \propto p_{gcc}^{C(f)+1}$$

This is a decreasing exponential distribution

Putting it all together

$$C = D - R$$

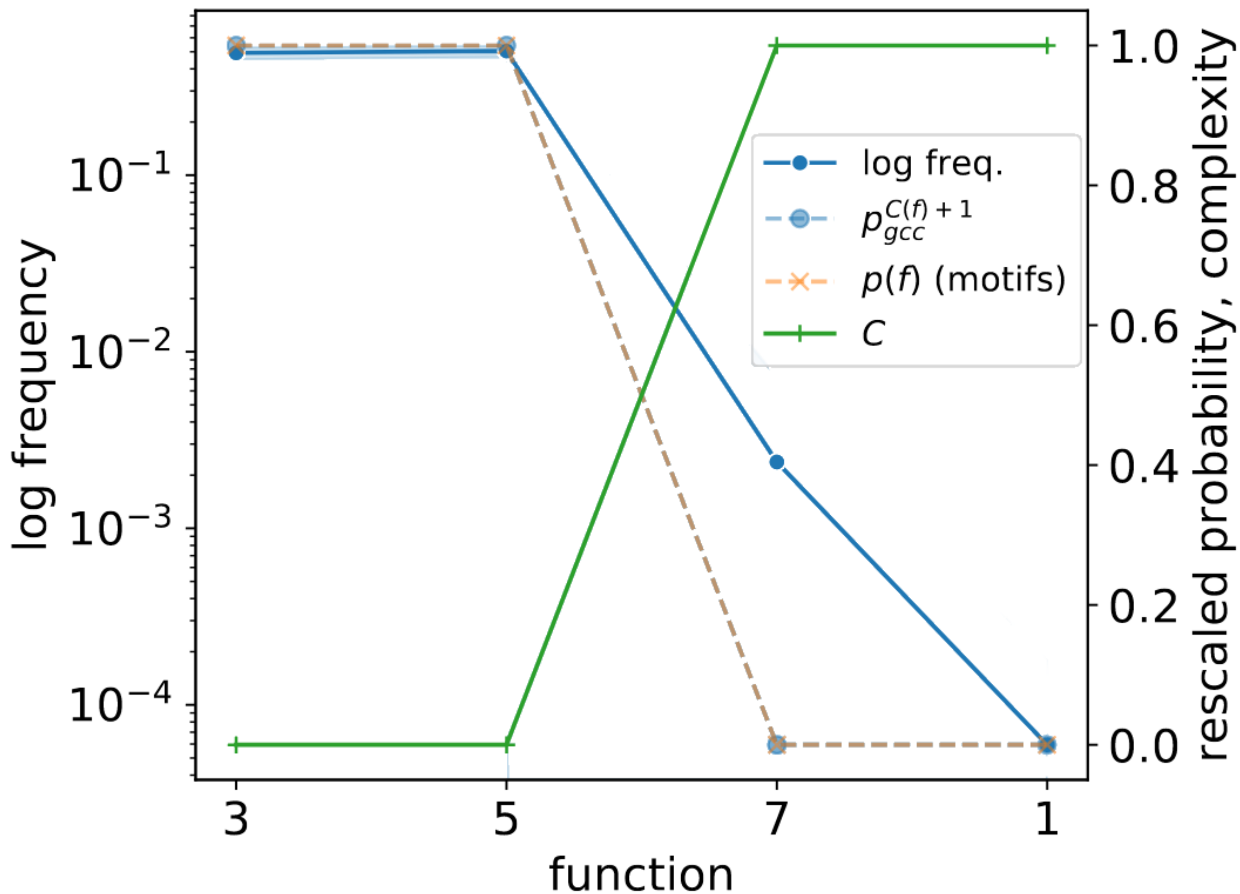
$$N_{gcc} = C + 1$$

$$p(f) \propto p_{gcc}^{C(f)+1}$$

- *Hamming asymmetry of function* $\rightarrow$
- *Decision Tree Complexity* $\rightarrow$
- *number of required network paths* $\rightarrow$
- *number of nodes in GCC* $\rightarrow$
- *probability factor (proportionality)*

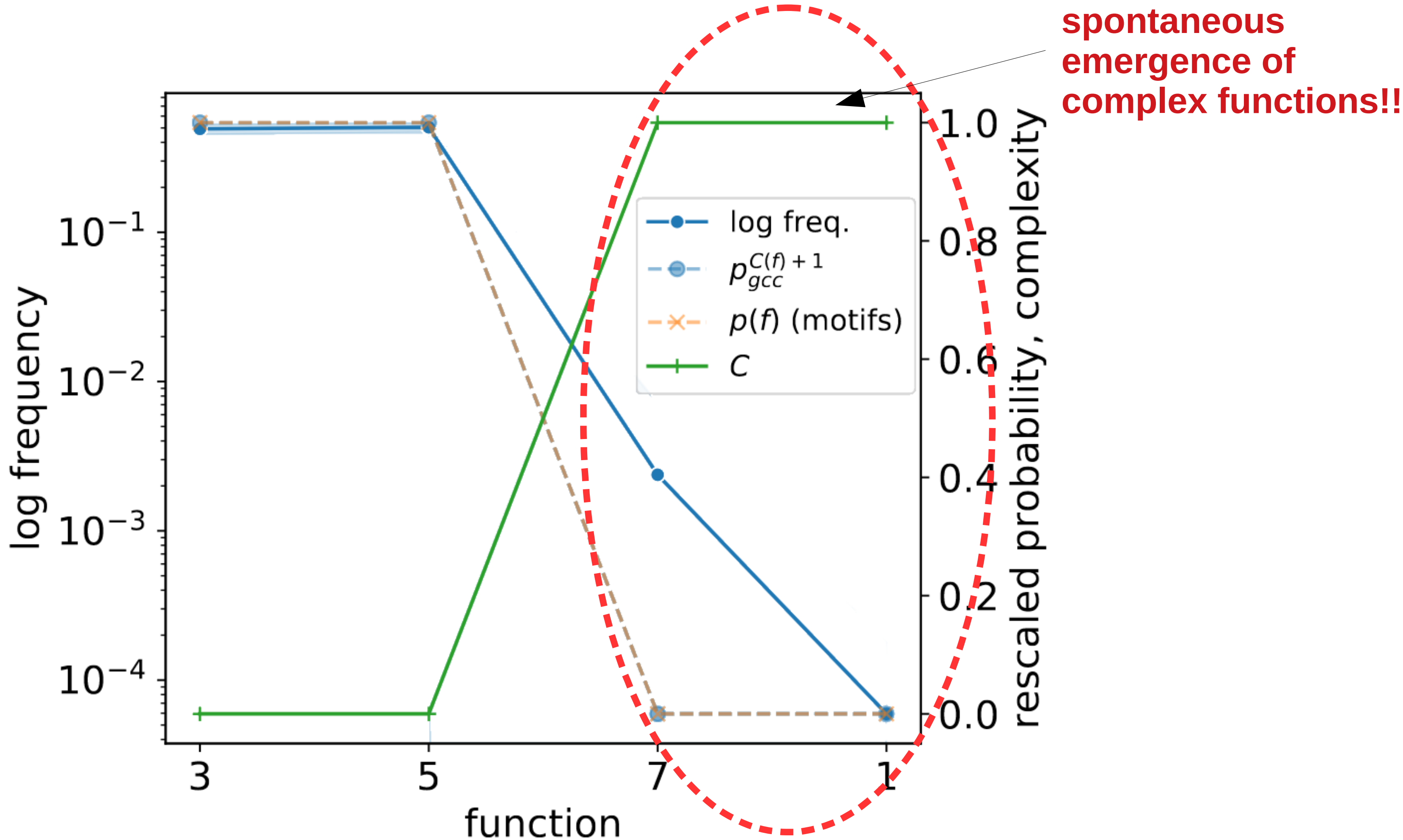
The probability of a monotone function in the LTM is proportional to the number of axial reflection ***asymmetries*** *of its Hamming cube.*

Function Frequency and Rank Ordering



Inverse relationship between function ***complexity*** *and* ***frequency*** $\rightarrow$ *rank ordering, inversely related to complexity.*

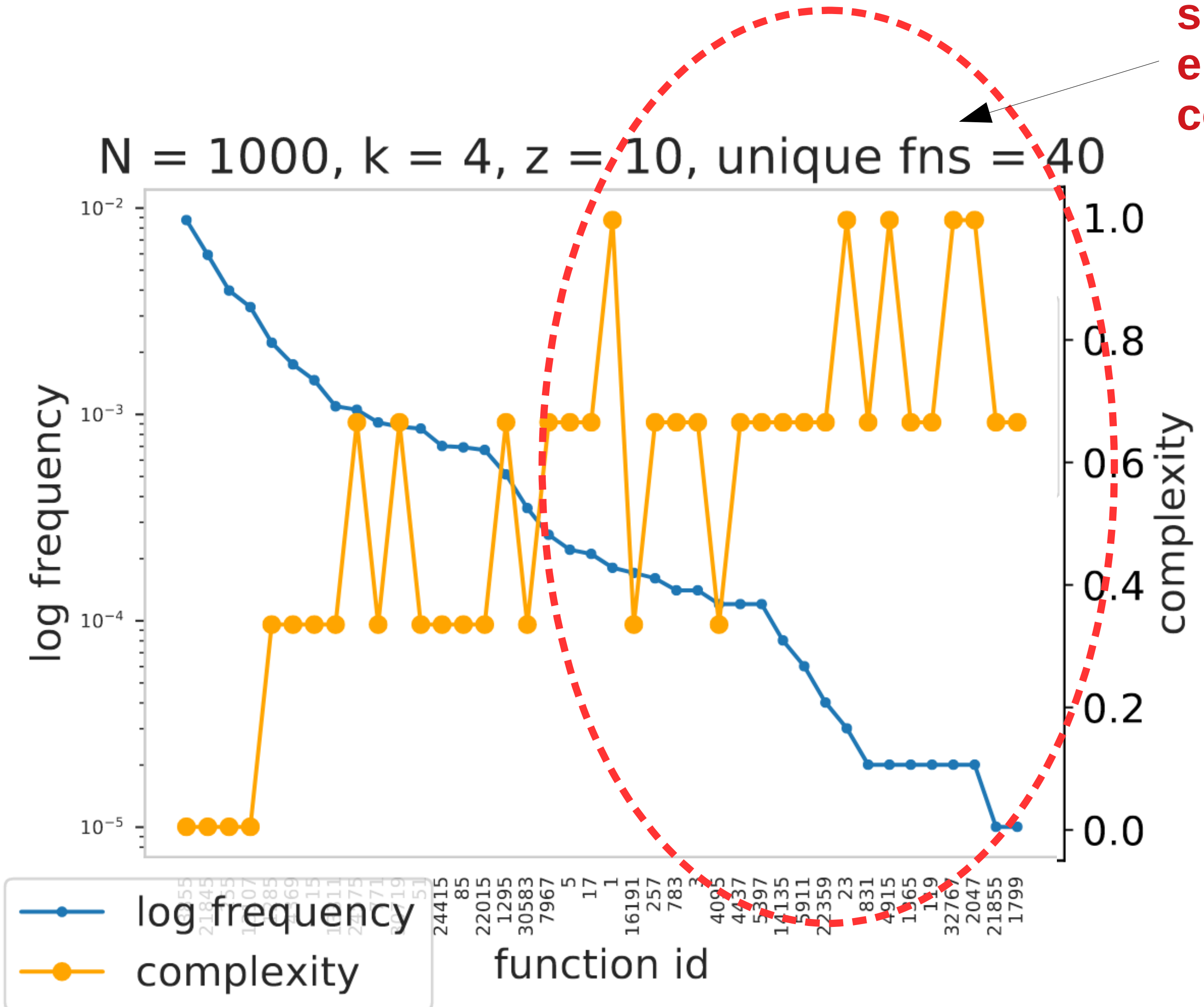
Function Frequency and Rank Ordering



Inverse relationship between function ***complexity*** *and* ***frequency*** $\rightarrow$ *rank ordering, inversely related to complexity.*

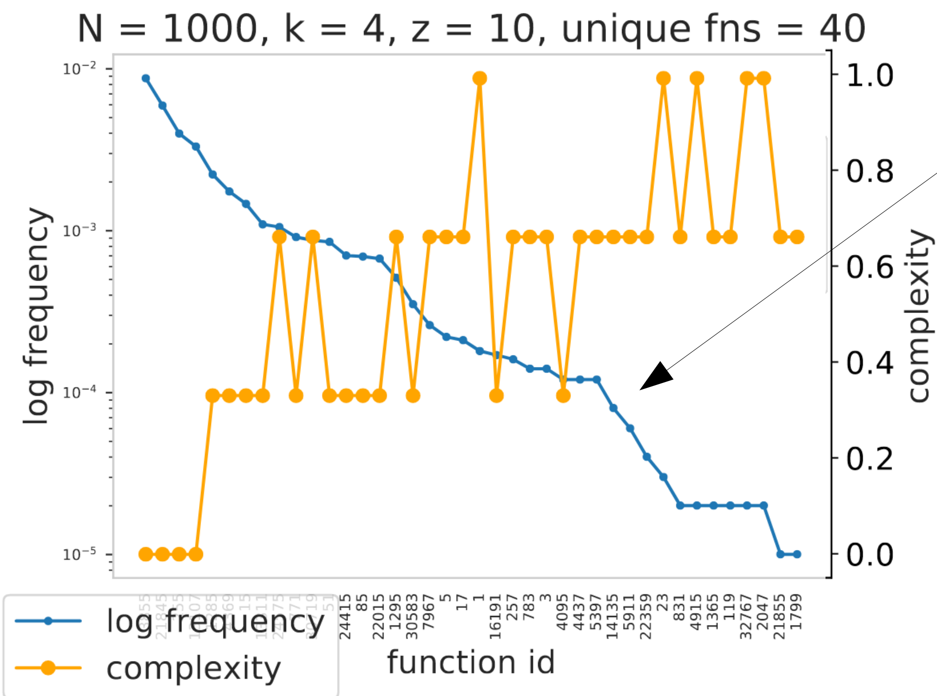
Function Frequency and Rank Ordering

spontaneous emergence of complex functions!!

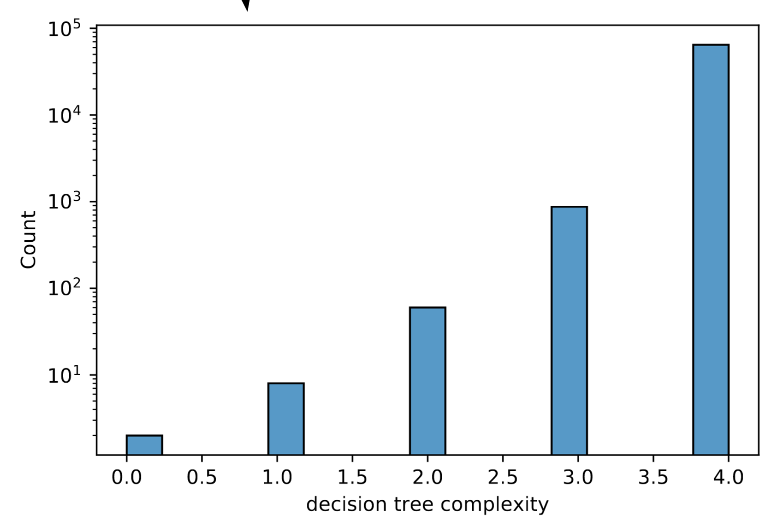


frequency (blue) is fit by exponential, having $r^2 = 0.88$

Function Frequency and Rank Ordering



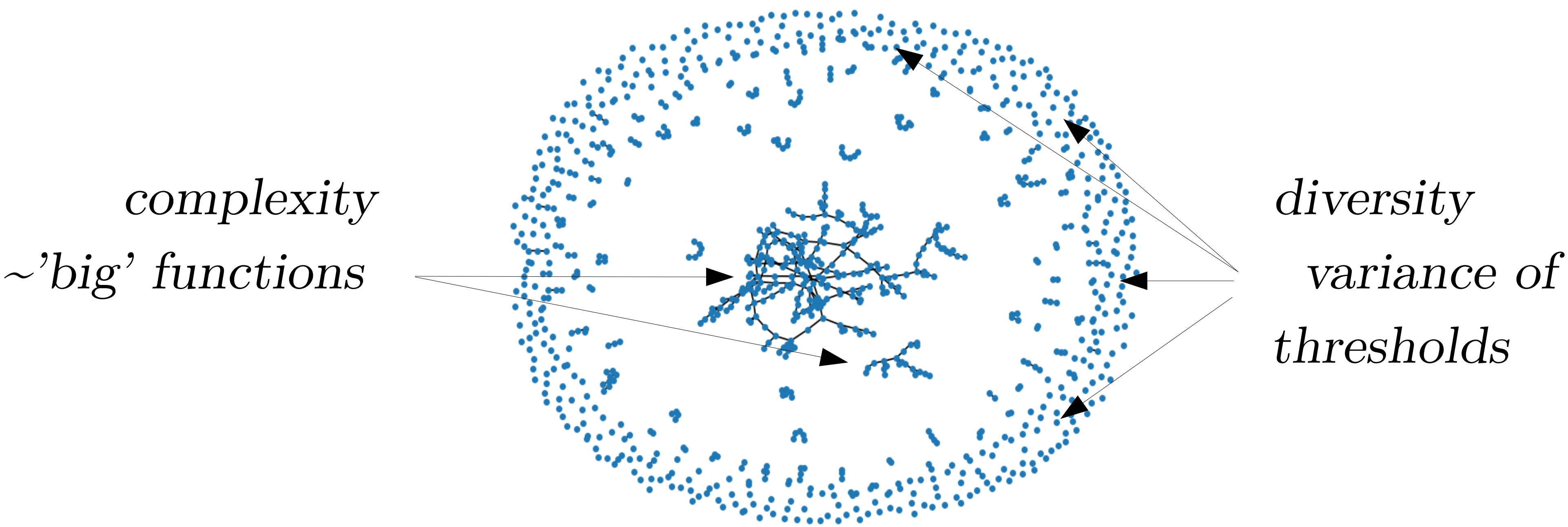
surprising in light of super-exponential availability of complex functions



frequency (blue) is fit by exponential, having $r^2 = 0.88$

Function Frequency and Rank Ordering

Inverse relationship between function ***complexity*** *and* ***frequency*** $\rightarrow$ *rank ordering (inverse to complexity)*

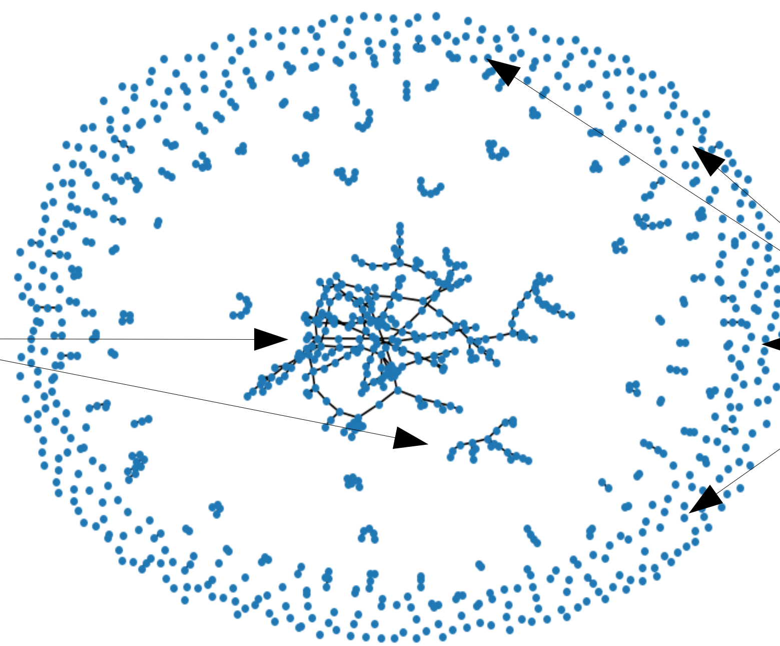


Perhaps ***Pareto-optimal near criticality****:*
best ***trade-off between complexity and diversity*** *when connecting "logic modules" (available connected components).*
(cf. 'criticality hypothesis') [Beggs 2003]

Function Frequency and Rank Ordering

Inverse relationship between function ***complexity*** *and* ***frequency*** $\rightarrow$ *rank ordering (inverse to complexity)*

complexity
~'big' functions



diversity
variance of
thresholds

Perhaps Pareto-optimal ***near criticality****:*
best ***trade-off between complexity and diversity*** *when connecting "logic modules" (available connected components).*
(cf. 'criticality hypothesis') [Beggs 2003]

can rewire and obtain new complex functions?

Boolean function probabilities in the ALTM: Monotone decreasing functions

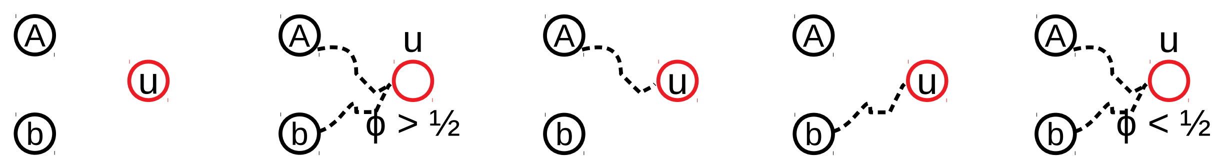
A	B	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	**0**	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	**1**	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	**0**	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	**1**	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

graphs for monotonic-decreasing functions (f_8, f_{10}, f_{12}, f_{14}, f_{15}), (NAND, NOR, ...) are similar to monotone functions (above).

Boolean function probabilities in the ALTM: Monotone decreasing functions

A	B	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

$f_{15}(a,b) = 1;$ $f_{14}(a,b) = a$ NAND $b;$ $f_{12}(a,b) = \neg a;$ $f_{10}(a,b) = \neg b;$ $f_{8}(a,b) = a$ NOR b

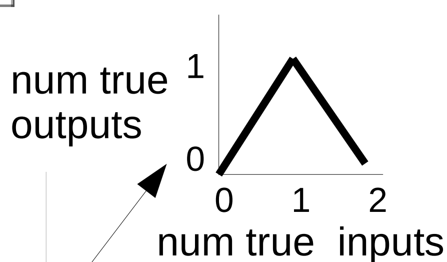


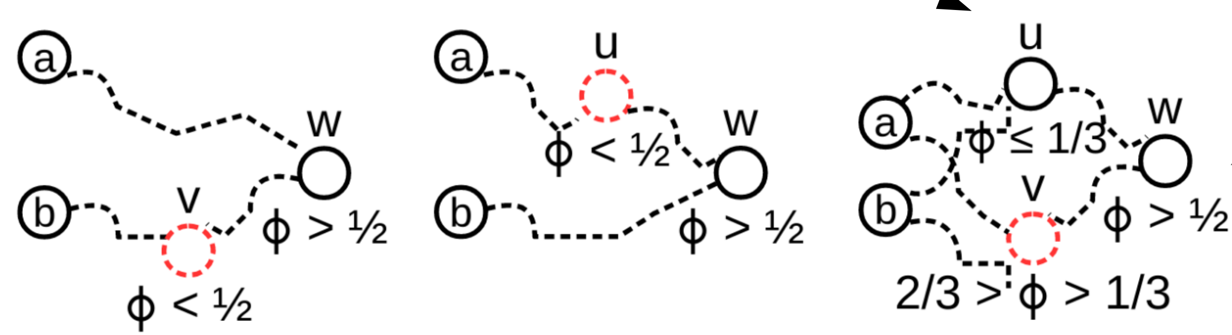
simplest sufficient networks to calculate the monotonic ***decreasing*** *functions*

Boolean function probabilities in the ALTM: non-monotone functions

A	B	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

XOR
not monotone!

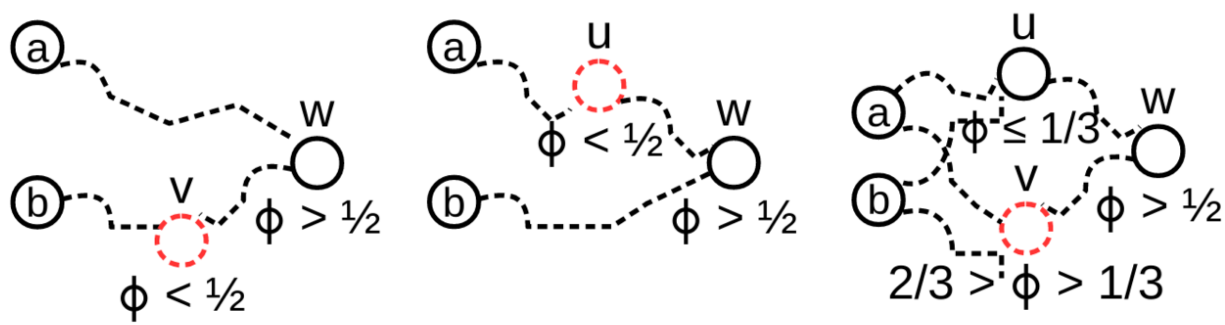




$f_2(a,b) = a \wedge \sim b$ $f_4(a,b) = \sim a \wedge b$ $f_6(a,b) = a \text{ XOR } b$

simplest sufficient networks to calculate non-monotone functions
(also f_9, f_{11}, f_{13})

Boolean function probabilities in the ALTM: non-monotone functions

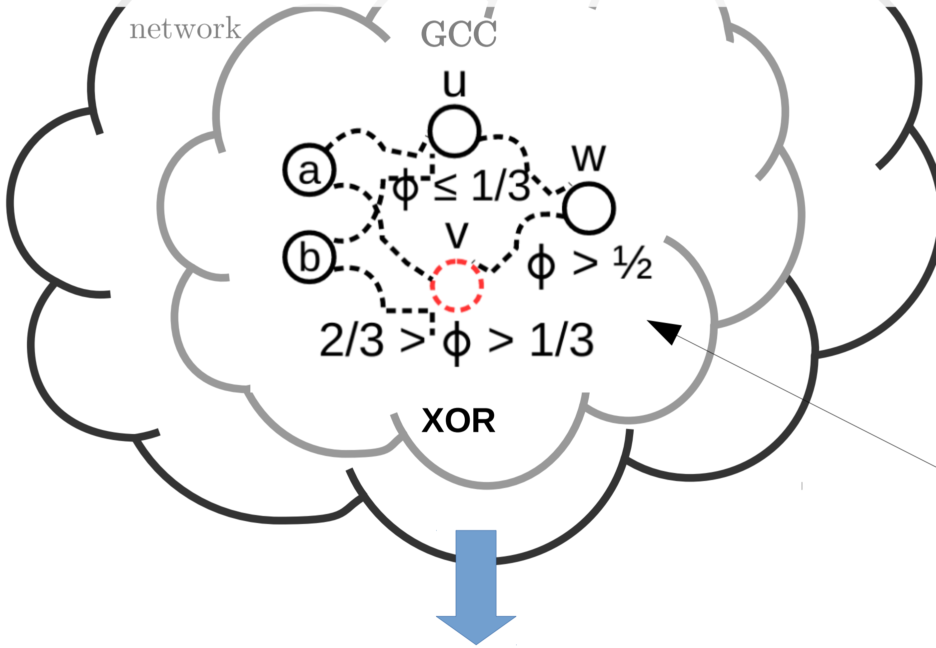


$f_2(a,b) = a \wedge \sim b$ $f_4(a,b) = \sim a \wedge b$ $f_6(a,b) = a \text{ XOR } b$

non-monotonic functions probability:

$$\left\{\begin{array}{l} p(f_2) \propto p_{path}^3 \\ p(f_4) \propto p_{path}^3 \\ p(f_6) \propto p_{path}^6 \end{array}\right.$$

Boolean function probabilities in the ALTM: non-monotone functions



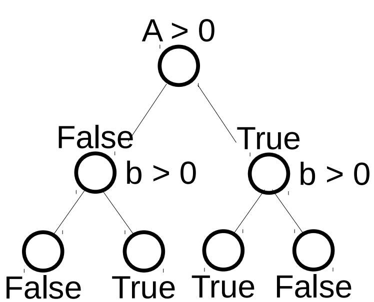
5 nodes in GCC

non-monotonic functions probability: $\left\{ p(f_6) \propto p_{path}^{6} \rightarrow \boldsymbol{p_{gcc}}^{5} \right.$

Boolean function probabilities in the ALTM: non-monotone functions

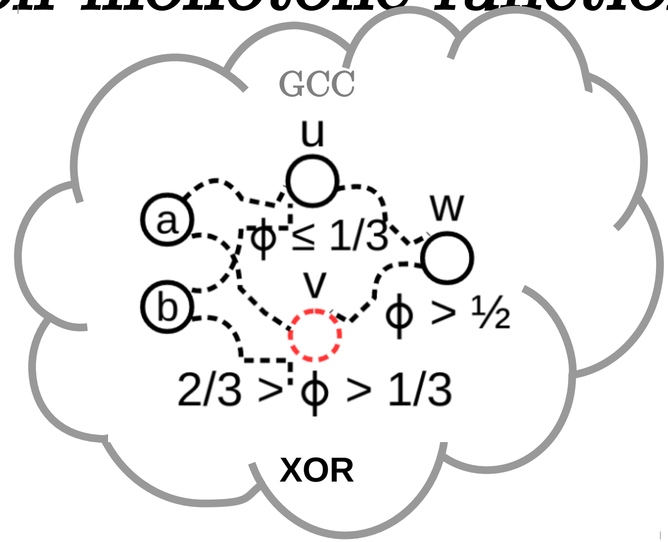
A	B	6
0	0	0
0	1	1
1	0	1
1	1	0

$f_6(a,b) = a \text{ XOR } b$



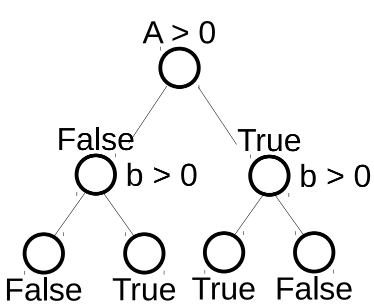
$C = 2 \Rightarrow p(f) \propto p_{gcc}^{C(f)+1} \Rightarrow p_{gcc}{}^{3}$

Boolean function probabilities in the ALTM: non-monotone functions



non-monotonic functions probability: $\left\{ p(f_6) \propto p_{path}^6 \rightarrow \boldsymbol{p_{gcc}}^{5} \right.$

$\neq$

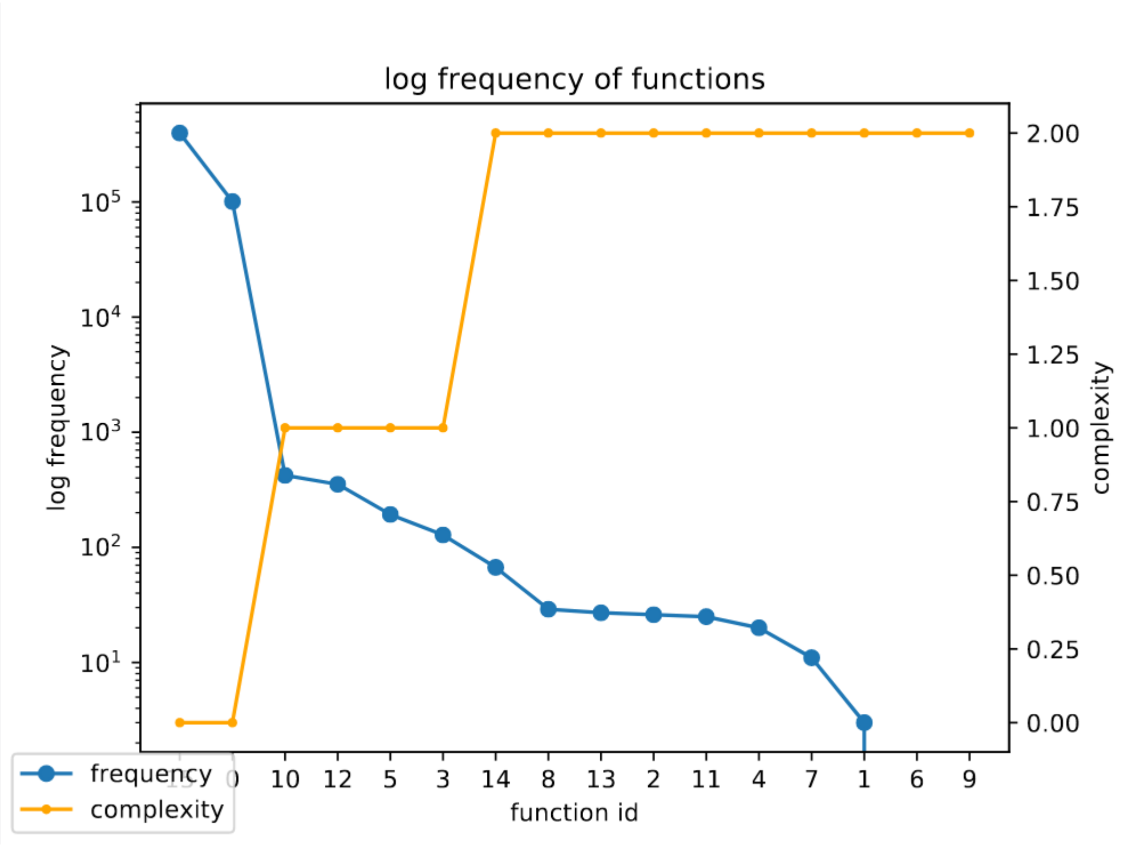


$C = 2$

$$p(f) \propto p_{gcc}^{C(f)+1}$$

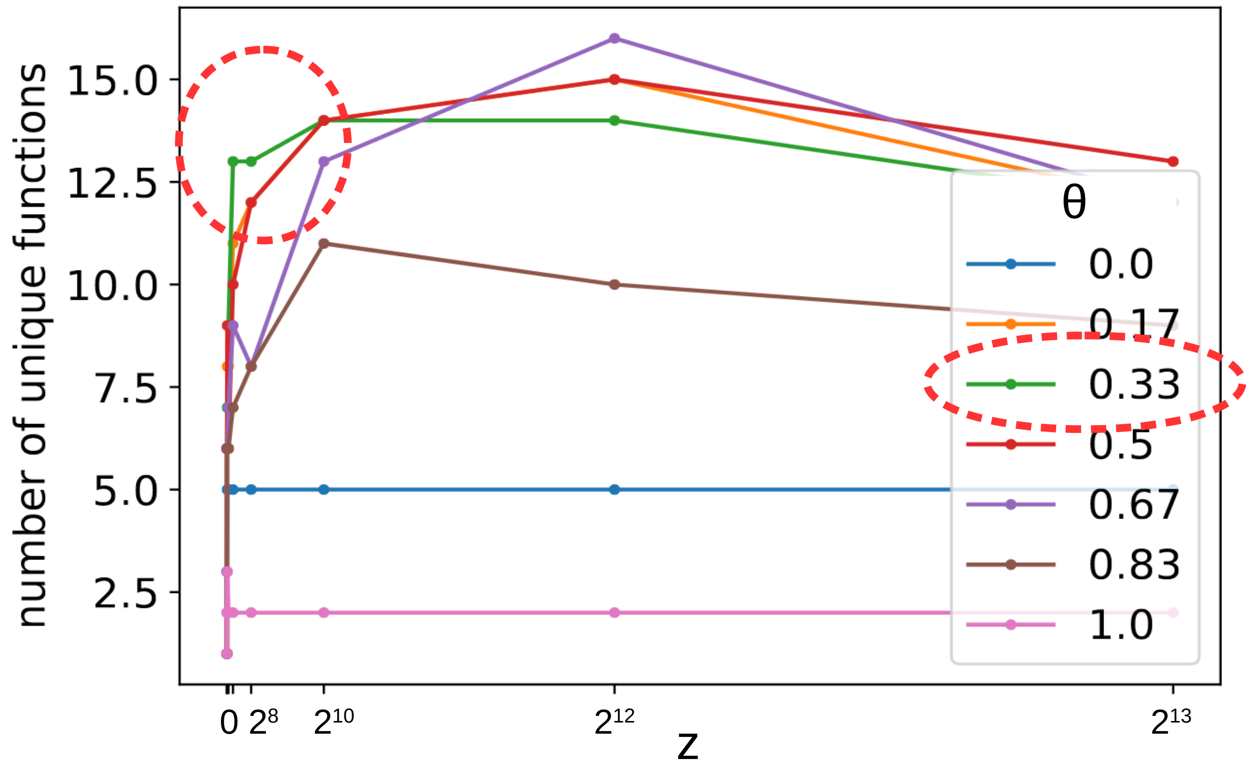
$\boldsymbol{p_{gcc}}^{3}$

Boolean function probabilities in the ALTM: non-monotone functions



Nevertheless, decision tree complexity does still inversely relate to frequency.

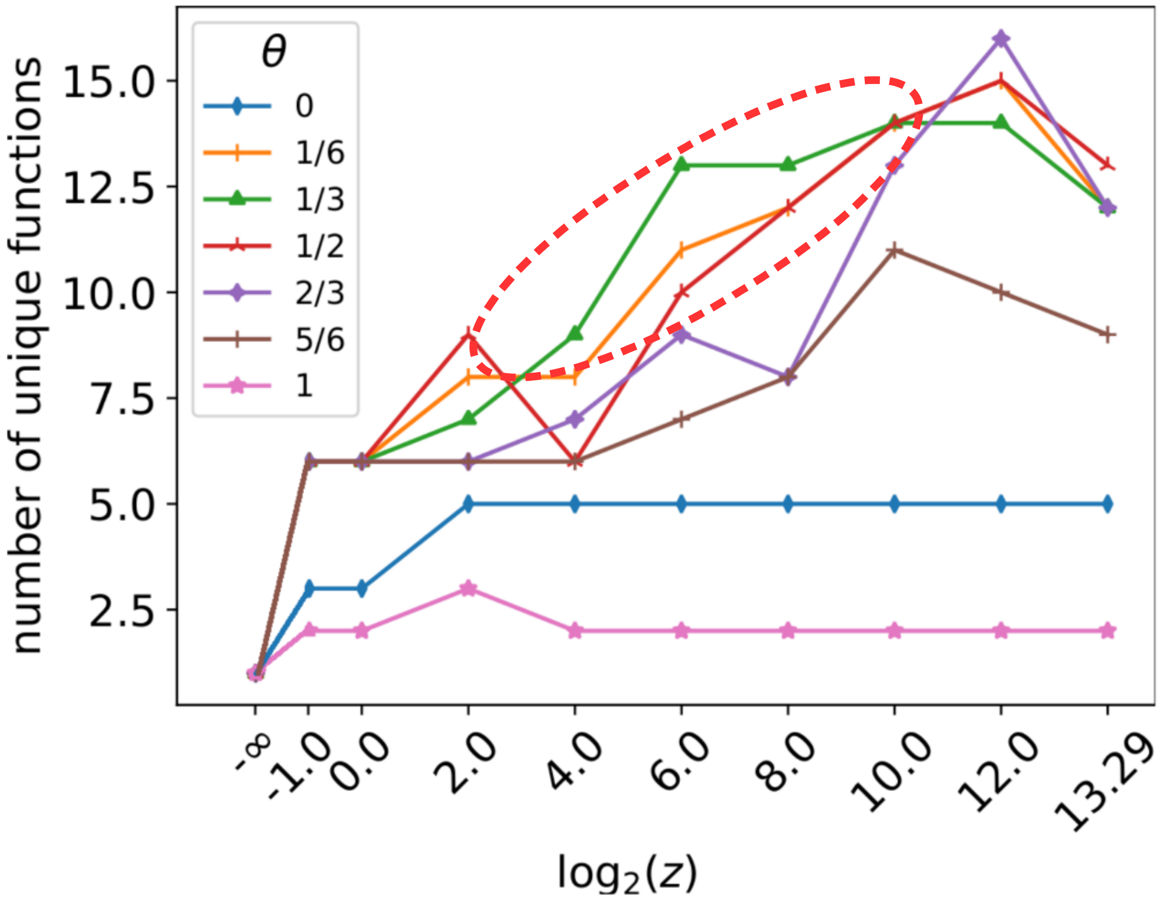
Antagonism / Inhibition ~ 1/3
Maximizes functions near criticality



θ = 1/3 for z in [2^3, 2^{10}] *maximizes number of unique functions observed*

- *c.f.:* ***'criticality hypothesis'*** $\rightarrow$ *information processing optimized in brain near criticality*

Antagonism / Inhibition ~ 1/3
Maximizes functions near criticality

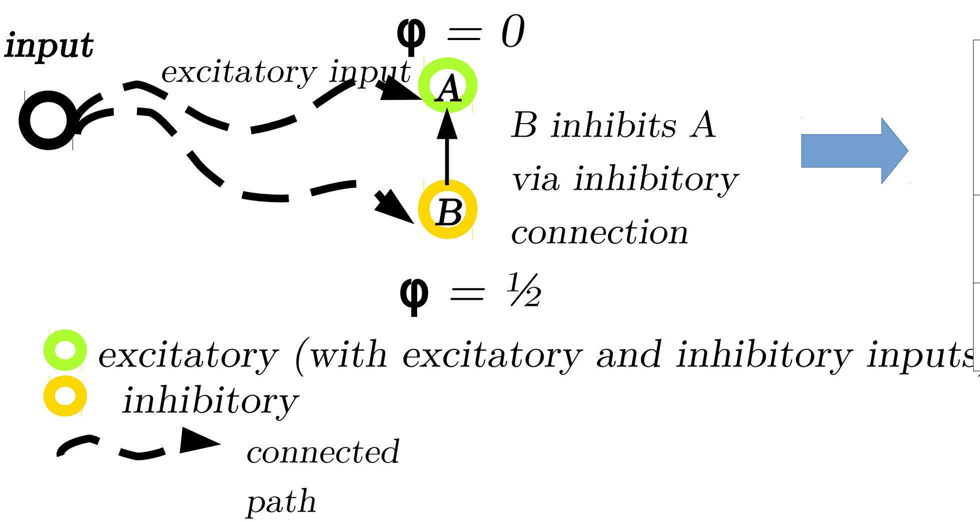


θ = 1/3 for z in [2^3, 2^{10}] *maximizes number of unique functions observed*

- *c.f. [Capano et al., 2015] → info. processing optimized near 30% inhibition*
- *30% inhibition prevalent in biology*

Interchangeability of
LTM Antagonism and McCulloch-Pitts Inhibition

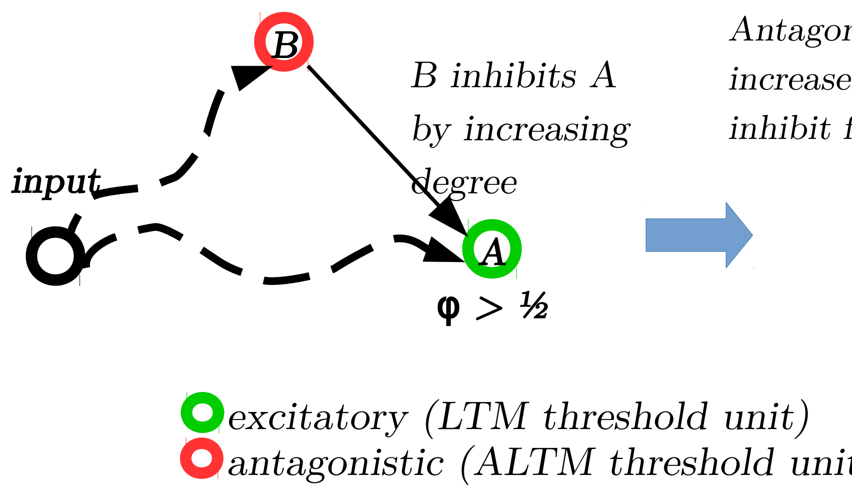
1. Inhibition to Antagonism



A antagonistic to the input

input	A	B
0	1	0
1	0	1

2. Antagonism to Inhibition



Antagonistic neighbor(s) increase degree and inhibit firing

B is inhibitory to A

input	A	B
0	0	1
1	0	0

Summary

- *Network cascades are percolation (<u>ubiquitous</u> and naturally occurring)*
- *Cascades compute logic*
- *sub-networks compute logic as* logic motifs *or* automata
- *Antagonism (= inhibition) yields universal computation*
- *Boolean function space is* huge*, how do large networks navigate it (learn)?*
- ***Random networks of threshold units yield a rank ordering of complexity***
  - ***i.e. complex Boolean functions emerge spontaneously***
- *Percolation lets us predict proportionality of function probabilities*
- *For monotone functions, there is an apparent relationship between*
    - *function frequency*
    - *decision tree complexity*
    - *Boolean function symmetry*
- *Results for 1/3 antagonism coincide with findings elsewhere and in biology.*

- *Symmetry-breaking in the network (formation of GCC) yields complexity of functionality*

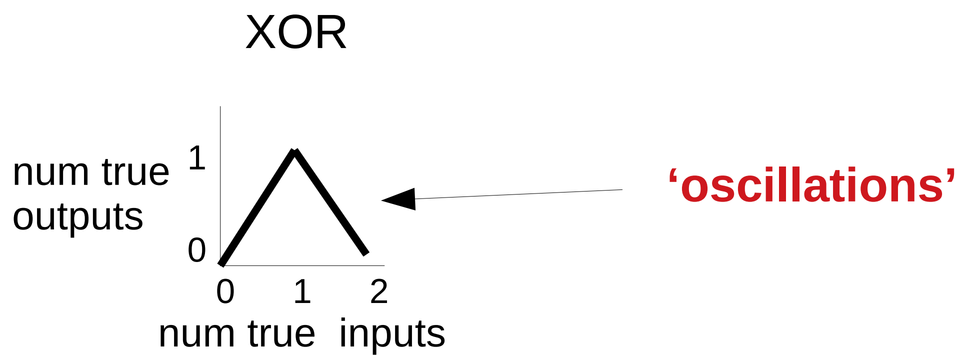
Future Work

- *Better complexity measures for **non-monotone functions** - Boolean Fourier analysis?*

$f_6(a,b)$ = a XOR b

A	B
0	0
0	1
1	0
1	1

6
0
1
1
0

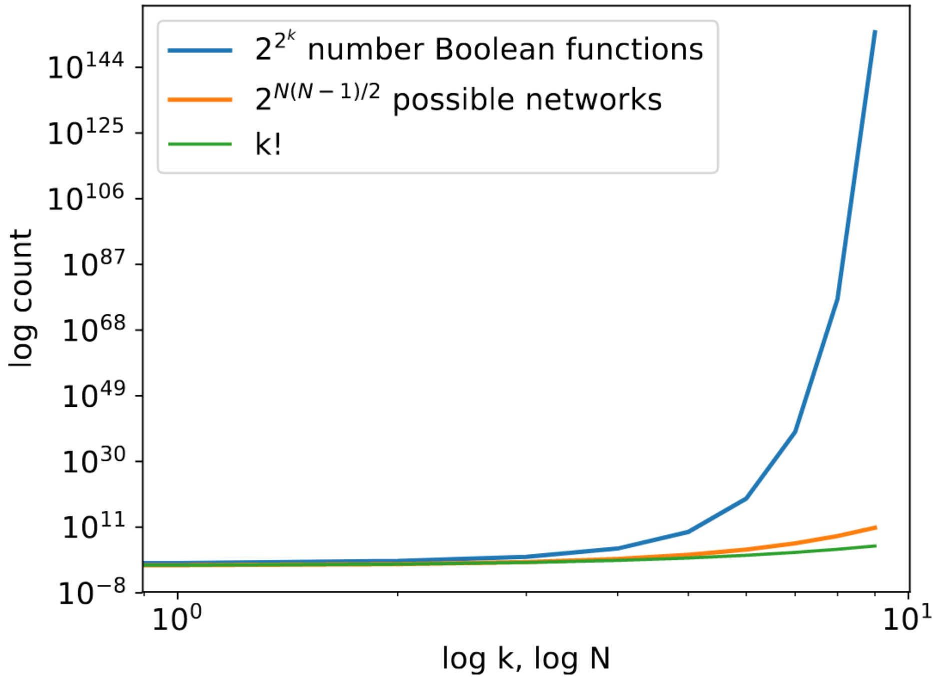


Future Work

- *generalize predictions to $k \gg 2$ inputs and much larger networks ($N \sim 10^9$ nodes)*
  - *learning in large combinatorial spaces*

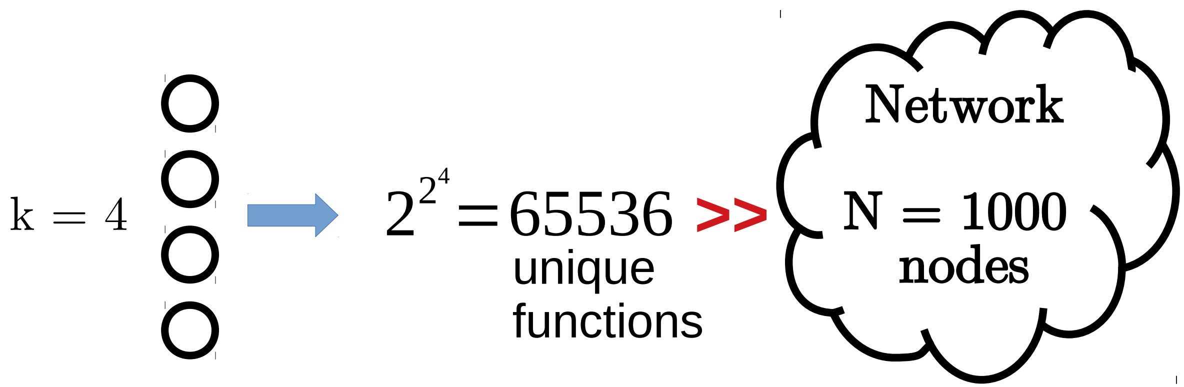
!!!?? **far too big! →**

need abstraction (symmetry?)



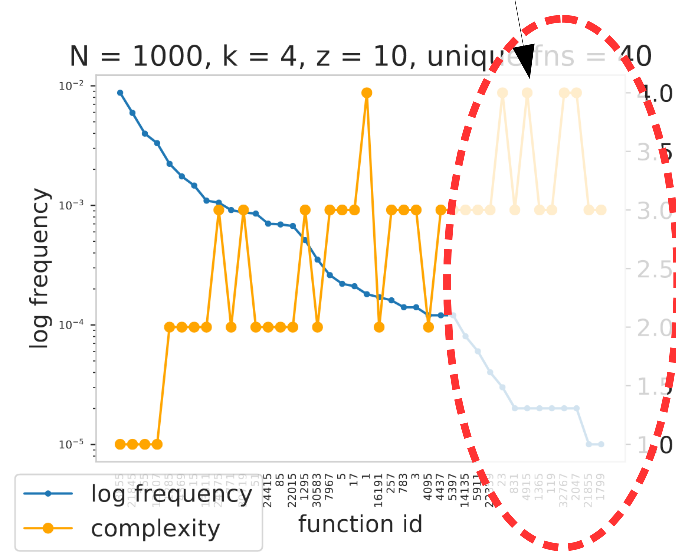
Future Work

- *modularity as related to network's complexity capacity for k inputs*



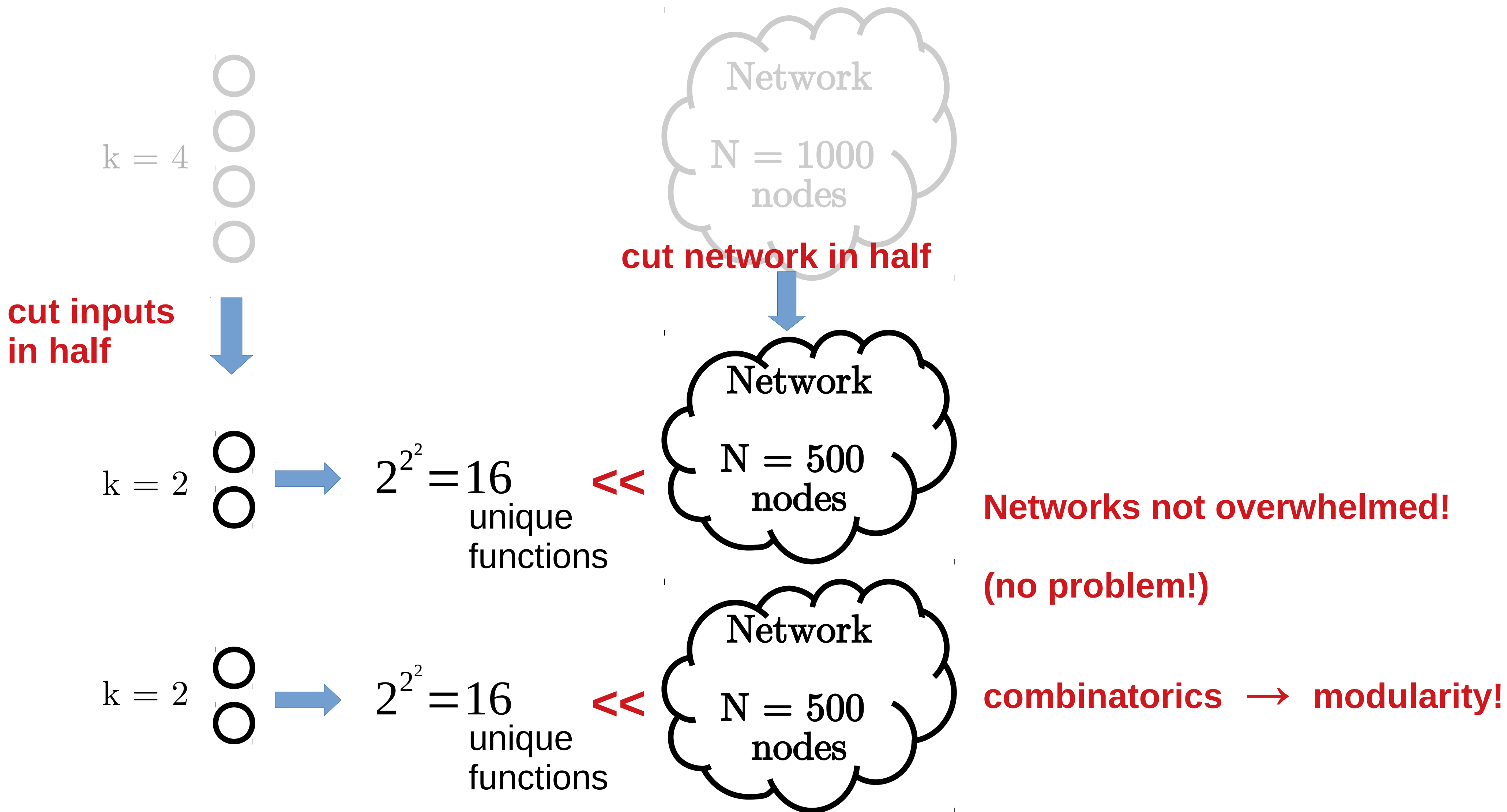
Network overwhelmed!

(can't compute complex functions)



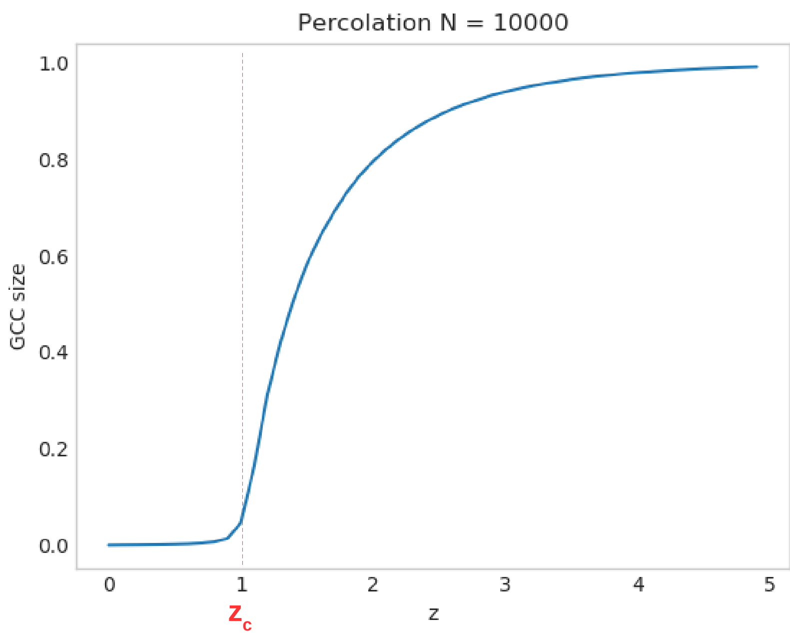
Future Work

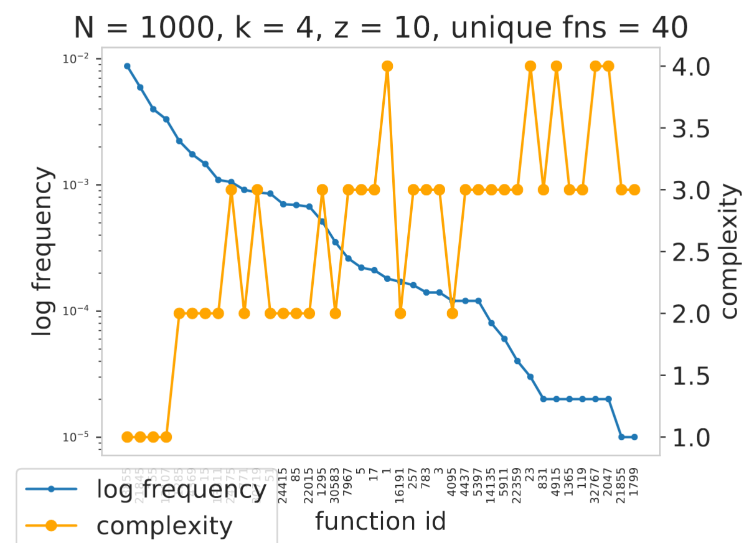
- *modularity as related to network's complexity capacity for k inputs*



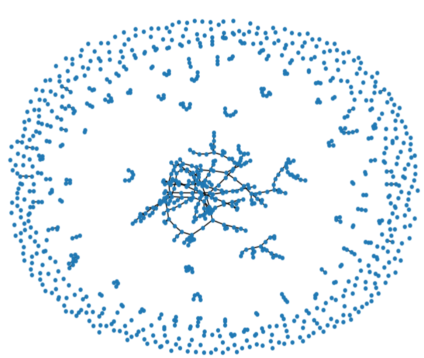
Future Work

- *symmetry of network ↔ symmetry of functions*
- ***general theory of structure and function?*** *(!!)*





symmetry-breaking phase transition

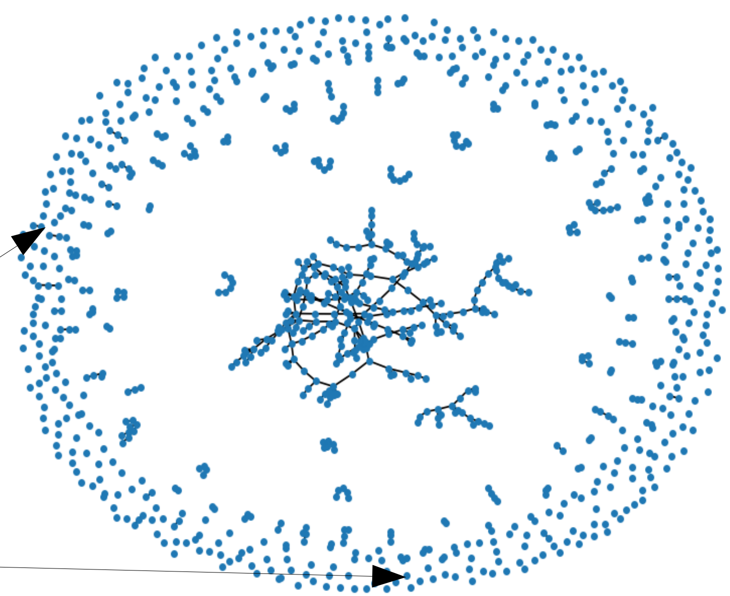


emergence of **asymmetric** (complex) functions

Future Work

- *network as* ***information engine*** $\rightarrow$
  general conservation law of information or complexity ??
  (cf. Liouville theorem – conservation of information,
  Noether's theorem – symmetries correspond to conservation)

A	B	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1



Input perturbations have complexity, information content

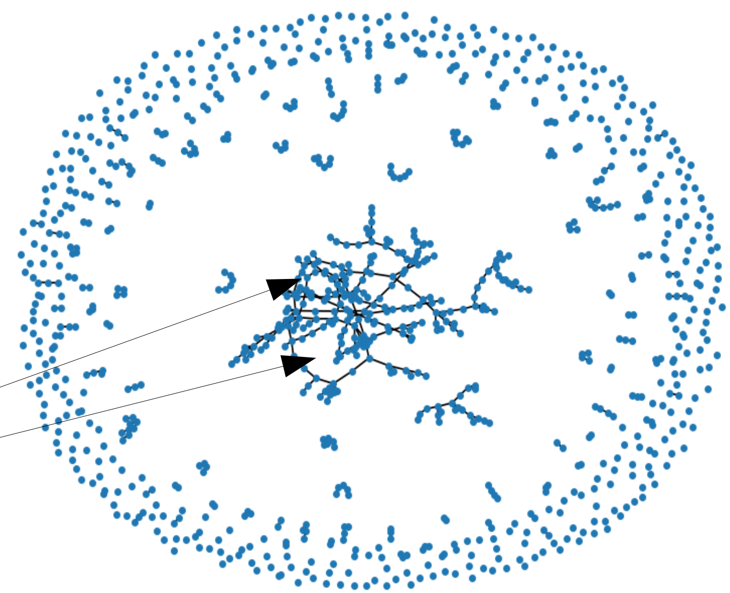
most nodes lose complexity (or information)

percolation:
scale-invariance
of component size
$p(x) \sim cx^{-\beta}$

Future Work

- *network as **information engine** –*
  general conservation law of information or complexity ??
  (cf. Liouville's, Noether's theorems?)

A	B	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1



Input perturbations have complexity, information content

a few nodes increase complexity or information
(contain **interaction** information)

– recall decreasing exponential rank - ordering

percolation:
scale-invariance
of component size
$p(x) \sim cx^{-\beta}$

Future Work

- *information processing in terms of* ***entropy***
  - *(cf. Lizier, Prokopenko; Beggs, Plenz et al.)*

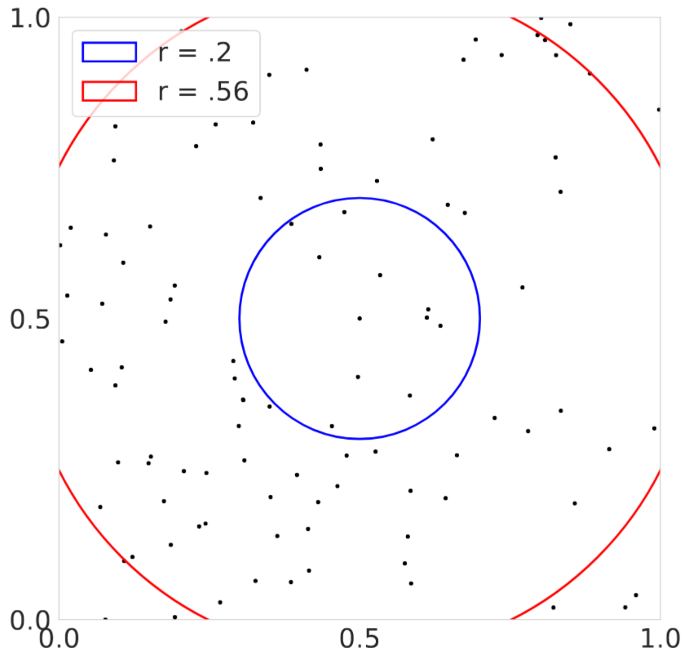
A	B	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

$$T_{X\to Y} = I(Y_t; X_{t-1:t-L} \mid Y_{t-1:t-L})$$

e.g. ***transfer entropy***
(conditional mutual entropy
between inputs and outputs)

Future Work

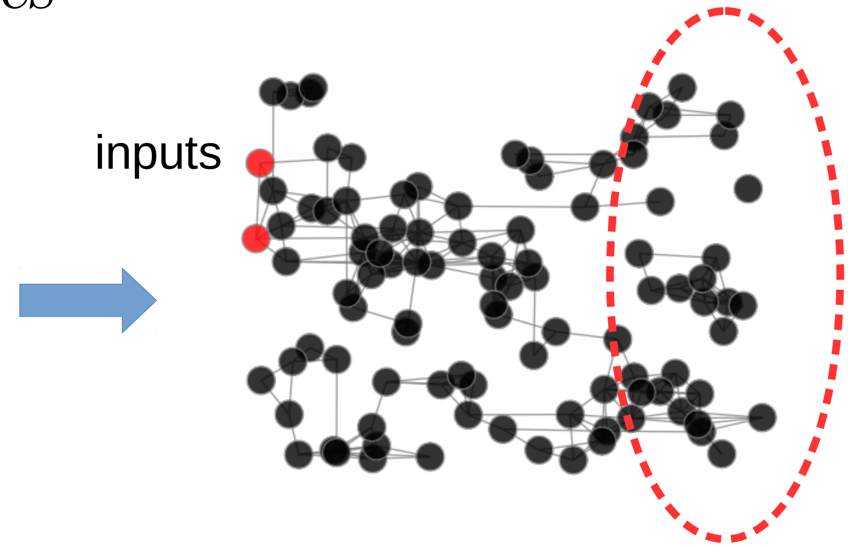
- *geographic (Euclidean) constraints and other network topologies*



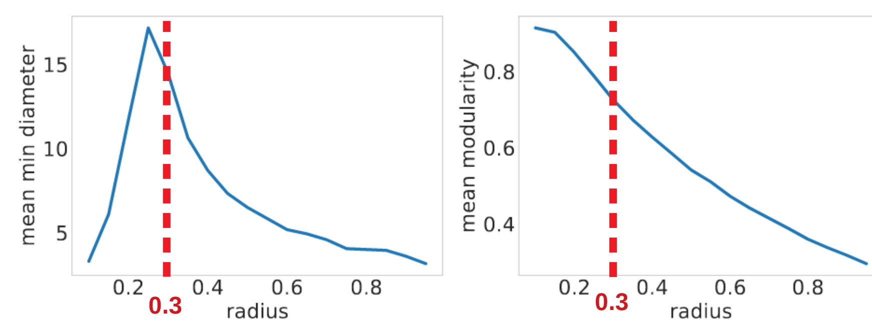
Random Geometric Network
(connections are
restricted by radius)

**are nodes at
farthest distance
most complex??**

inputs



**convenient
for physical
(e.g. brain)
modules!**

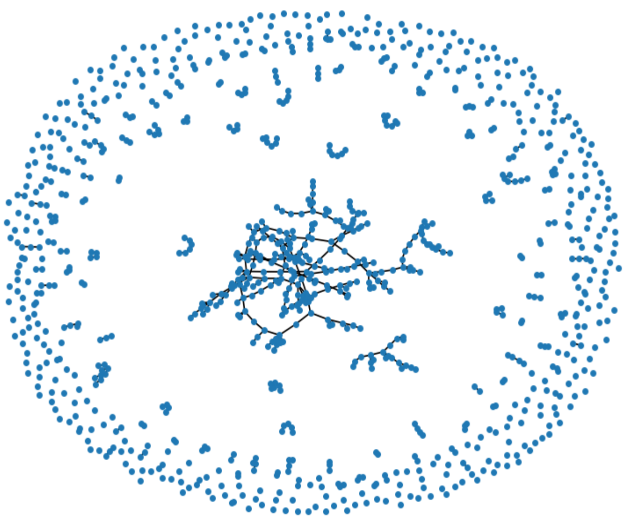


networks are **modular** and
NOT small-world at restricted radius
(e.g. radius = 0.3)

Future Work

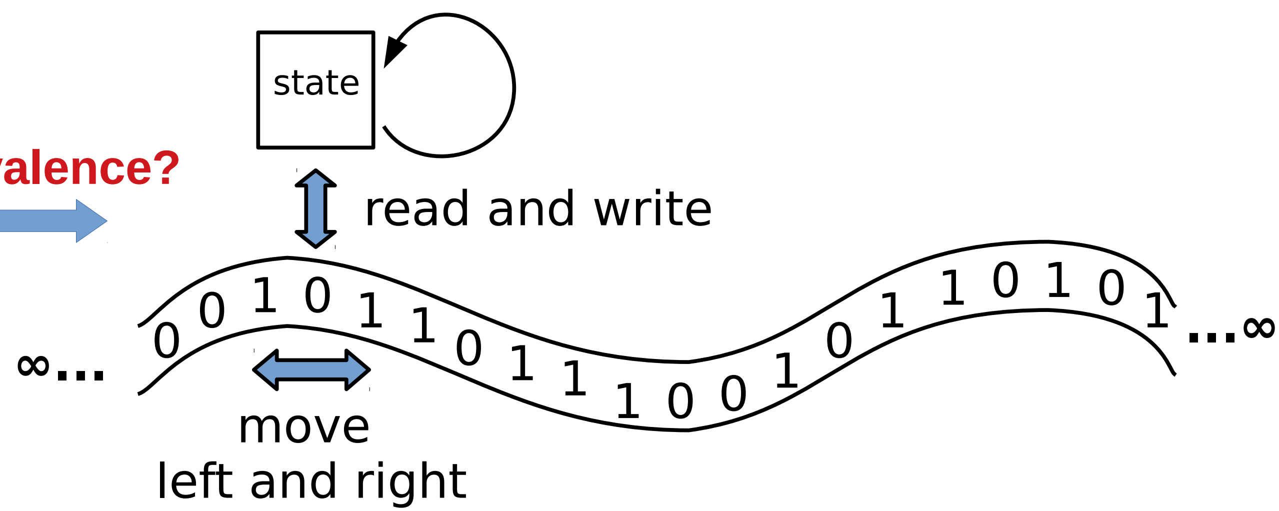
- *Turing completeness*

ALTM



equivalence?

Turing Machine



Rewire → state dynamics

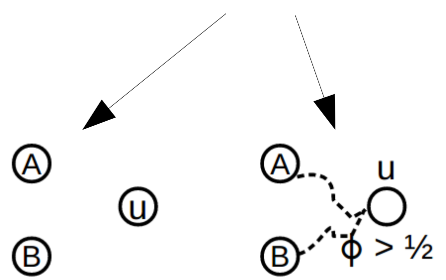
(especially near criticality)

Future Work

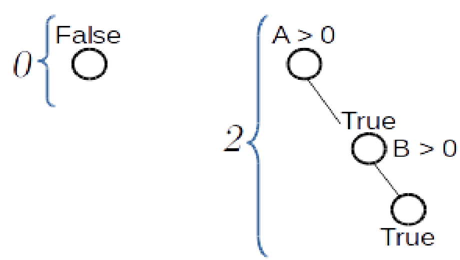
- *Consciousness [Tononi]*

A	B	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

number of paths required for simplest graph



decision tree complexity (depth of decision tree)



number of paths required

= **decision tree complexity**

= **how integrative a node is**

= **consciousness** [Tononi]

References

- *Wilkerson, Galen J., and Sotiris Moschoyiannis.*

  "Logic and learning in network cascades." Network Science (2021): 1-18.

- *Beggs, John M., and Dietmar Plenz. "Neuronal avalanches in neocortical circuits."*

  Journal of neuroscience 23.35 (2003): 11167-11177.

- *Watts, D. J. "A simple model of global cascades on random networks."*

  Proceedings of the National Academy of Sciences 99.9 (2002): 5766-5771.

- *Capano, Vittorio, Hans J. Herrmann, and Lucilla De Arcangelis. "Optimal percentage of inhibitory synapses in multi-task learning." Scientific reports 5.1 (2015): 1-5.*

- *Rojas, Raúl. Neural networks: a systematic introduction. Springer Science & Business Media, 2013.*

- *R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, Network motifs: simple building blocks of complex networks, Science 298, 824 (2002).*

- *Newman, Mark. Networks. Oxford university press, 2018.*

- *Von Neumann, J. "Probabilistic logics and the synthesis of reliable organisms from unreliable components." Automata Studies (1956): 43-98.*

- *Christensen, Kim, and Nicholas R. Moloney. Complexity and Criticality. Vol. 1. World Scientific Publishing Company, 2005.*

- *van der Hofstad, Percolation and Random Graphs, https://www.win.tue.nl/~rhofstad/percolation_randomgraphs_rev.pdf*

- *Lizier, Joseph T., Mikhail Prokopenko, and Albert Y. Zomaya. "Local information transfer as a spatiotemporal filter for complex systems." Physical Review E 77.2 (2008): 026110.*

- *Tononi, Giulio, et al. "Integrated information theory: from consciousness to its physical substrate." Nature Reviews Neuroscience 17.7 (2016): 450-461.*

Thank you!

Correspondence:

g.wilkerson21@imperial.ac.uk

Acknowledgements

Funding: This work was supported partly by EIT Digital IVZW, through the Real-Time Flow Project under Grant 18387-SGA201,

and partly by the EPSRC project AGELink under Grant EP/R511791/1.