What's new in coding?
When classical codes meet modern ideas.

QIP 2016

Rüdiger Urbanke

Friday, January 15th, 2016

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Standard Setup

“information source”

“channel”

Standard Setup

“information source” “channel”

Y

C={000, 010, 101, 111} block code

codeword

n … blocklength

	Algebraic:	Iterative/Codes on Graphs:	Polar:
Examples:			
Based on:			
Decoding:			
+ :			
— :			
Applications:			

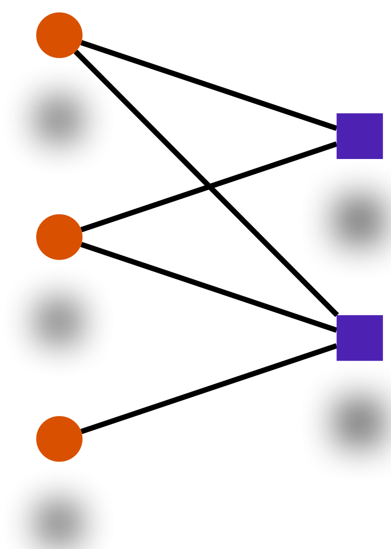
	Algebraic:	Iterative/Codes on Graphs:	Polar:
Examples:	Reed-Muller BCH Reed-Solomon Lattices		
Based on:			
Decoding:			
+ :			
— :			
Applications:			

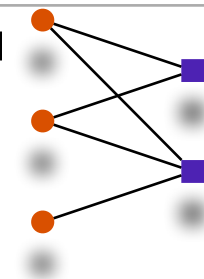
	Algebraic:	Iterative/Codes on Graphs:	Polar:
Examples:	Reed-Muller BCH Reed-Solomon Lattices		
Based on:	packing in Hamming or Euclidean space 		
Decoding:			
+ :			
— :			
Applications:			

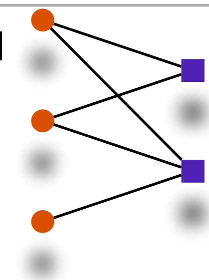
	Algebraic:	Iterative/Codes on Graphs:	Polar:
Examples:	Reed-Muller BCH Reed-Solomon Lattices		
Based on:	packing in Hamming or Euclidean space 		
Decoding:	algebraic sphere decoder		
+ :			
— :			
Applications:			

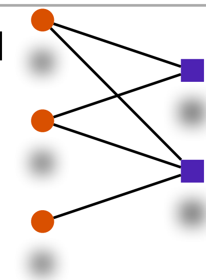
	Algebraic:	Iterative/Codes on Graphs:	Polar:
Examples:	Reed-Muller BCH Reed-Solomon Lattices		
Based on:	packing in Hamming or Euclidean space 		
Decoding:	algebraic sphere decoder		
+ :	simple high throughput low complexity		
— :	does not achieve capacity		
Applications:			

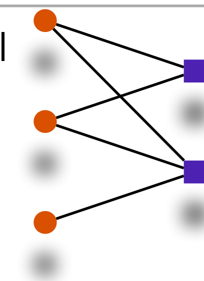
	Algebraic:	Iterative/Codes on Graphs:	Polar:
Examples:	Reed-Muller BCH Reed-Solomon Lattices		
Based on:	packing in Hamming or Euclidean space 		
Decoding:	algebraic sphere decoder		
+ :	simple high throughput low complexity		
— :	does not achieve capacity		
Applications:	space CD, DVD hard disks optical		

	Algebraic:	Iterative/Codes on Graphs:	Polar:
Examples:	Reed-Muller BCH Reed-Solomon Lattices	Convolutional Turbo LDPC LDGM 	
Based on:	packing in Hamming or Euclidean space 		
Decoding:	algebraic sphere decoder		
+ :	simple high throughput low complexity		
— :	does not achieve capacity		
Applications:	space CD, DVD hard disks optical		

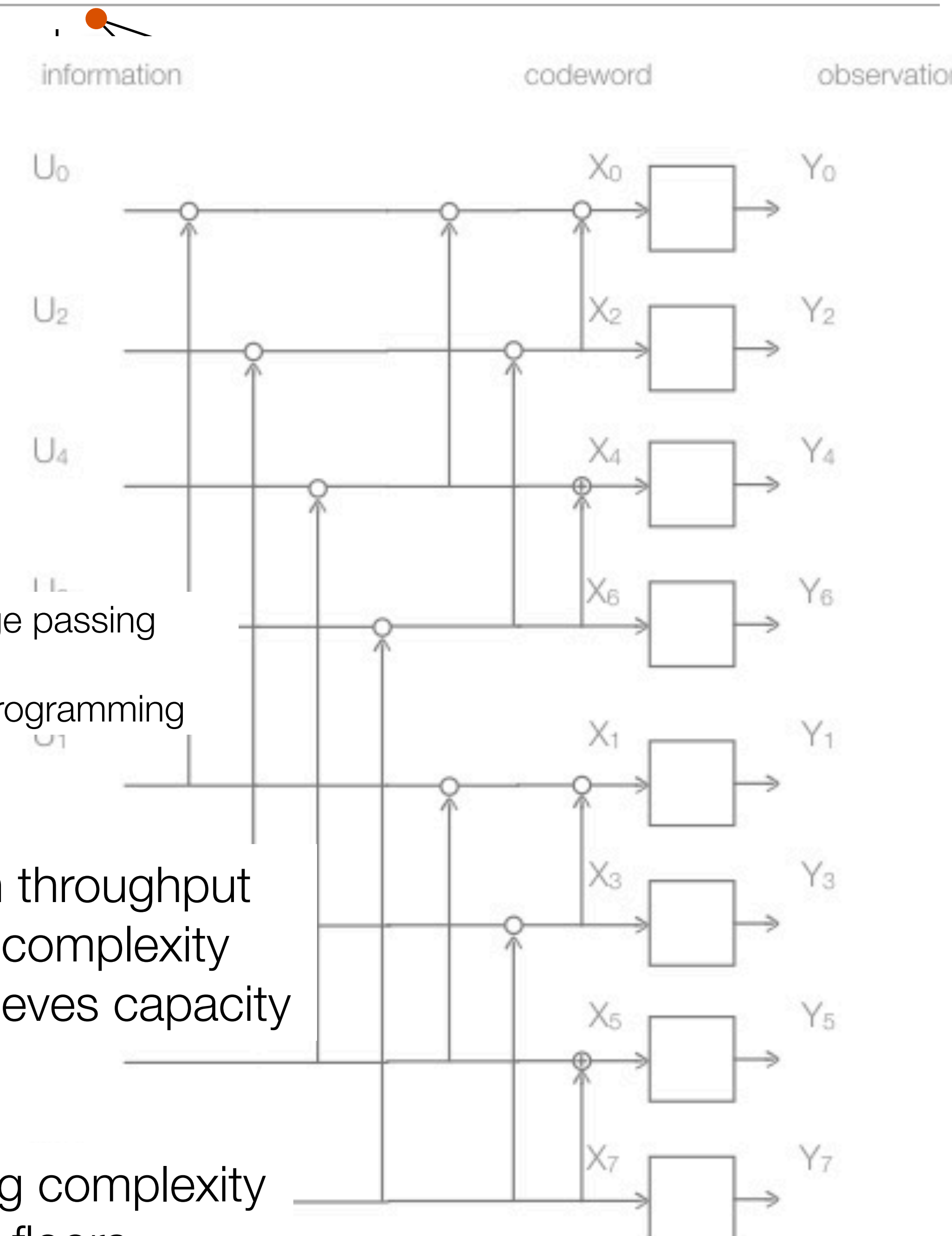
	Algebraic:	Iterative/Codes on Graphs:	Polar:
Examples:	Reed-Muller BCH Reed-Solomon Lattices	Convolutional Turbo LDPC LDGM 	
Based on:	packing in Hamming or Euclidean space 	factor graph approximation of bit-MAP decoding	
Decoding:	algebraic sphere decoder		
+ :	simple high throughput low complexity		
— :	does not achieve capacity		
Applications:	space CD, DVD hard disks optical		

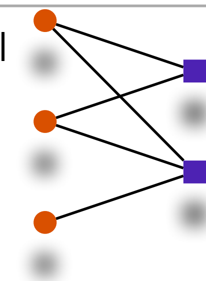
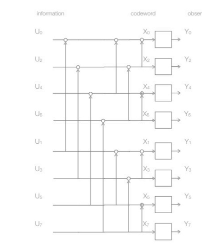
	Algebraic:	Iterative/Codes on Graphs:	Polar:
Examples:	Reed-Muller BCH Reed-Solomon Lattices	Convolutional Turbo LDPC LDGM 	
Based on:	packing in Hamming or Euclidean space 	factor graph approximation of bit-MAP decoding	
Decoding:	algebraic sphere decoder	message passing flipping linear programming	
+ :	simple high throughput low complexity		
— :	does not achieve capacity		
Applications:	space CD, DVD hard disks optical		

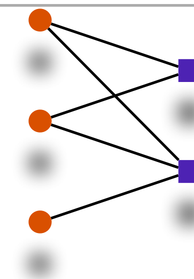
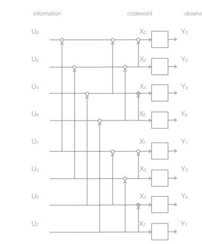
	Algebraic:	Iterative/Codes on Graphs:	Polar:
Examples:	Reed-Muller BCH Reed-Solomon Lattices	Convolutional Turbo LDPC LDGM 	
Based on:	packing in Hamming or Euclidean space 	factor graph approximation of bit-MAP decoding	
Decoding:	algebraic sphere decoder	message passing flipping linear programming	
+ :	simple high throughput low complexity	high throughput low complexity achieves capacity	
— :	does not achieve capacity	wiring complexity error floors	
Applications:	space CD, DVD hard disks optical		

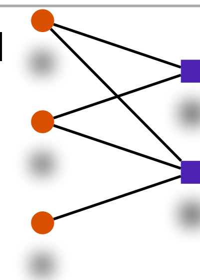
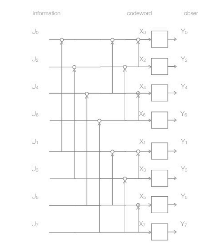
	Algebraic:	Iterative/Codes on Graphs:	Polar:
Examples:	Reed-Muller BCH Reed-Solomon Lattices	Convolutional Turbo LDPC LDGM 	
Based on:	packing in Hamming or Euclidean space 	factor graph approximation of bit-MAP decoding	
Decoding:	algebraic sphere decoder	message passing flipping linear programming	
+ :	simple high throughput low complexity	high throughput low complexity achieves capacity	
— :	does not achieve capacity	wiring complexity error floors	
Applications:	space CD, DVD hard disks optical	mobile WiFi optical power-line	

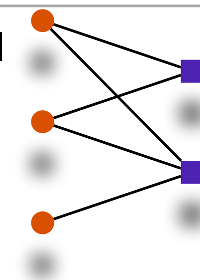
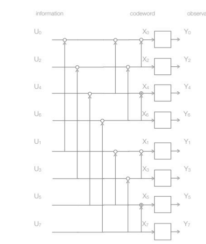
	Algebraic:	Iterative/Codes on Graphs:	Polar:
Examples:	Reed-Muller BCH Reed-Solomon Lattices	Convoluti Turbo LDPC LDGM	
Based on:	packing in Hamming or Euclidean space 	fac ap of de	
Decoding:	algebraic sphere decoder	message passing flipping linear programming	
+ :	simple high throughput low complexity	high throughput low complexity achieves capacity	
— :	does not achieve capacity	wiring complexity error floors	
Applications:	space CD, DVD hard disks optical	mobile WiFi optical power-line	

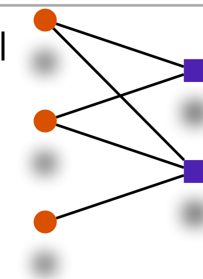
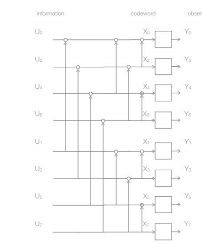


	Algebraic:	Iterative/Codes on Graphs:	Polar:
Examples:	Reed-Muller BCH Reed-Solomon Lattices	Convolutional Turbo LDPC LDGM 	
Based on:	packing in Hamming or Euclidean space 	factor graph approximation of bit-MAP decoding	chain rule of mutual information
Decoding:	algebraic sphere decoder	message passing flipping linear programming	
+ :	simple high throughput low complexity	high throughput low complexity achieves capacity	
— :	does not achieve capacity	wiring complexity error floors	
Applications:	space CD, DVD hard disks optical	mobile WiFi optical power-line	

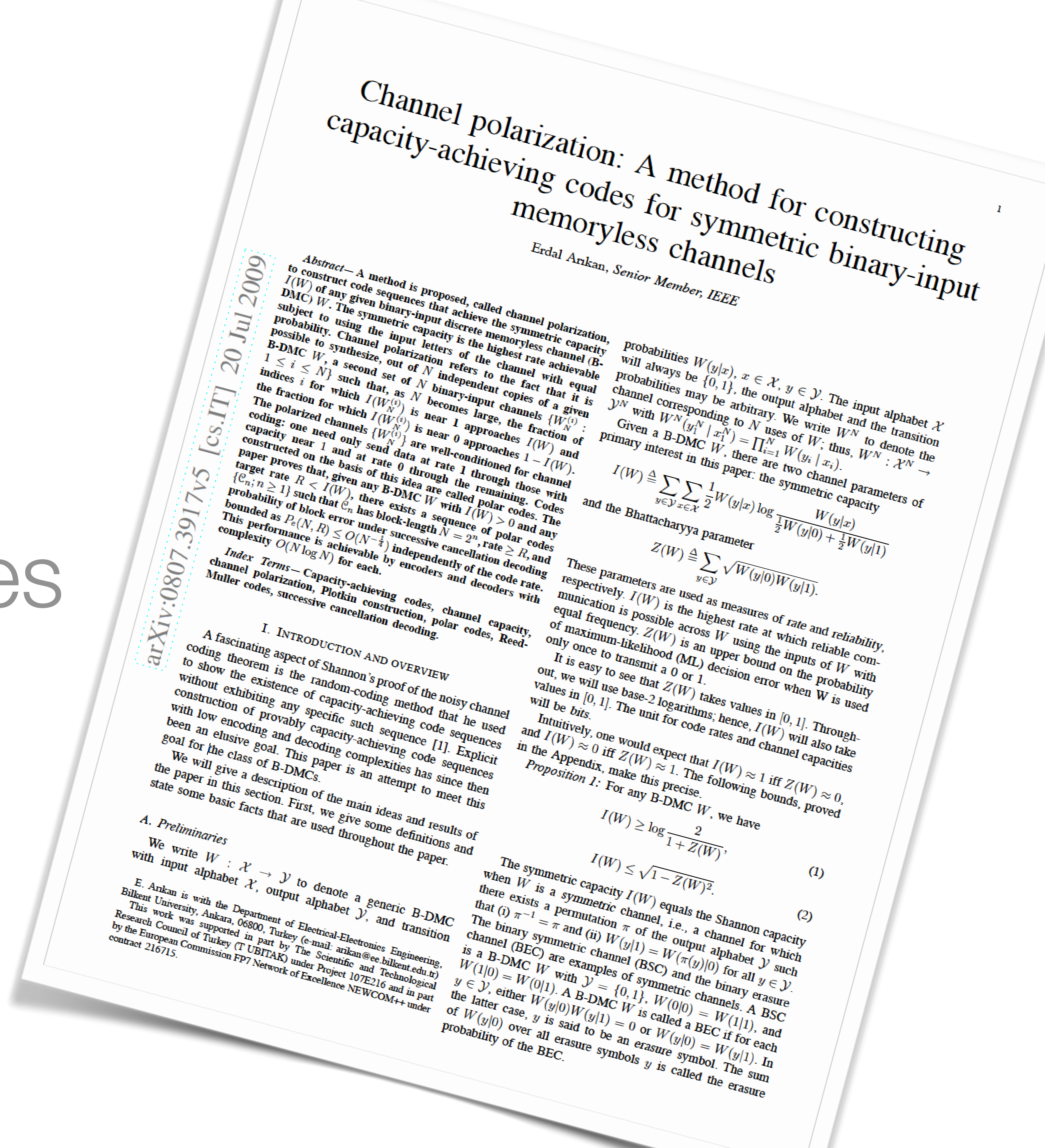
	Algebraic:	Iterative/Codes on Graphs:	Polar:
Examples:	Reed-Muller BCH Reed-Solomon Lattices	Convolutional Turbo LDPC LDGM 	
Based on:	packing in Hamming or Euclidean space 	factor graph approximation of bit-MAP decoding	chain rule of mutual information
Decoding:	algebraic sphere decoder	message passing flipping linear programming	successive
+ :	simple high throughput low complexity	high throughput low complexity achieves capacity	
— :	does not achieve capacity	wiring complexity error floors	
Applications:	space CD, DVD hard disks optical	mobile WiFi optical power-line	

	Algebraic:	Iterative/Codes on Graphs:	Polar:
Examples:	Reed-Muller BCH Reed-Solomon Lattices	Convolutional Turbo LDPC LDGM 	
Based on:	packing in Hamming or Euclidean space 	factor graph approximation of bit-MAP decoding	chain rule of mutual information
Decoding:	algebraic sphere decoder	message passing flipping linear programming	successive
+ :	simple high throughput low complexity	high throughput low complexity achieves capacity	elegant low complexity low energy achieves capacity
— :	does not achieve capacity	wiring complexity error floors	finite length not universal
Applications:	space CD, DVD hard disks optical	mobile WiFi optical power-line	

	Algebraic:	Iterative/Codes on Graphs:	Polar:
Examples:	Reed-Muller BCH Reed-Solomon Lattices	Convolutional Turbo LDPC LDGM 	
Based on:	packing in Hamming or Euclidean space 	factor graph approximation of bit-MAP decoding	chain rule of mutual information
Decoding:	algebraic sphere decoder	message passing flipping linear programming	successive
+ :	simple high throughput low complexity	high throughput low complexity achieves capacity	elegant low complexity low energy achieves capacity
— :	does not achieve capacity	wiring complexity error floors	finite length not universal
Applications:	space CD, DVD hard disks optical	mobile WiFi optical power-line	future wireless?

	Algebraic:	Iterative/Codes on Graphs:	Polar:
Examples:	Reed-Muller BCH Reed-Solomon Lattices	Convolutional Turbo LDPC LDGM 	
Based on:	packing in Hamming or Euclidean space 	factor graph approximation of bit-MAP decoding	chain rule of mutual information
Decoding:	algebraic sphere decoder	message passing flipping linear programming	successive
+ :	simple high throughput low complexity	high throughput low complexity achieves capacity	elegant low complexity low energy achieves capacity
— :	does not achieve capacity	wiring complexity error floors	finite length not universal
Applications:	space CD, DVD hard disks optical	mobile WiFi optical power-line	future wireless?

Polar Codes



Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels

Erdal Arıkan, *Senior Member, IEEE*

arXiv:0807.3917v5 [cs.IT] 20 Jul 2009

1

Abstract— **A method is proposed, called channel polarization, to construct code sequences that achieve the symmetric capacity $I(W)$ of any given binary-input discrete memoryless channel (B-DMC) W. The symmetric capacity is the highest rate achievable subject to using the input letters of the channel with equal probability. Channel polarization refers to the fact that it is possible to synthesize, out of N independent copies of a given B-DMC W, a second set of N binary-input channels $\{W_N^{(i)} : 1 \le i \le N\}$ such that, as N becomes large, the fraction of indices i for which $I(W_N^{(i)})$ is near 1 approaches $I(W)$ and the fraction for which $I(W_N^{(i)})$ is near 0 approaches $1 - I(W)$. The polarized channels $\{W_N^{(i)}\}$ are well-conditioned for channel coding: one need only send data at rate 1 through those with capacity near 1 and at rate 0 through the remaining. Codes constructed on the basis of this idea are called polar codes. The paper proves that, given any B-DMC W with $I(W) > 0$ and any target rate $R < I(W)$, there exists a sequence of polar codes $\{\mathcal{C}_n; n \ge 1\}$ such that $\mathcal{C}_n$ has block-length $N = 2^n$, rate $\ge R$, and probability of block error under successive cancellation decoding bounded as $P_e(N, R) \le O(N^{-\frac{1}{4}})$ independently of the code rate. This performance is achievable by encoders and decoders with complexity $O(N \log N)$ for each.**

Index Terms— **Capacity-achieving codes, channel capacity, channel polarization, Plotkin construction, polar codes, Reed-Muller codes, successive cancellation decoding.**

I. Introduction and Overview

A fascinating aspect of Shannon's proof of the noisy channel coding theorem is the random-coding method that he used to show the existence of capacity-achieving code sequences without exhibiting any specific such sequence [1]. Explicit construction of provably capacity-achieving code sequences with low encoding and decoding complexities has since then been an elusive goal. This paper is an attempt to meet this goal for the class of B-DMCs.

We will give a description of the main ideas and results of the paper in this section. First, we give some definitions and state some basic facts that are used throughout the paper.

A. *Preliminaries*

We write $W : \mathcal{X} \to \mathcal{Y}$ to denote a generic B-DMC with input alphabet $\mathcal{X}$, output alphabet $\mathcal{Y}$, and transition probabilities $W(y|x)$, $x \in \mathcal{X}$, $y \in \mathcal{Y}$. The input alphabet $\mathcal{X}$ will always be $\{0, 1\}$, the output alphabet and the transition probabilities may be arbitrary. We write W^N to denote the channel corresponding to N uses of W; thus, $W^N : \mathcal{X}^N \to \mathcal{Y}^N$ with $W^N(y_1^N \mid x_1^N) = \prod_{i=1}^N W(y_i \mid x_i)$.

Given a B-DMC W, there are two channel parameters of primary interest in this paper: the symmetric capacity

$$I(W) \triangleq \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} \frac{1}{2} W(y|x) \log \frac{W(y|x)}{\frac{1}{2}W(y|0) + \frac{1}{2}W(y|1)}$$

and the Bhattacharyya parameter

$$Z(W) \triangleq \sum_{y \in \mathcal{Y}} \sqrt{W(y|0)W(y|1)}.$$

These parameters are used as measures of *rate* and *reliability*, respectively. $I(W)$ is the highest rate at which reliable communication is possible across W using the inputs of W with equal frequency. $Z(W)$ is an upper bound on the probability of maximum-likelihood (ML) decision error when W is used only once to transmit a 0 or 1.

It is easy to see that $Z(W)$ takes values in $[0, 1]$. Throughout, we will use base-2 logarithms; hence, $I(W)$ will also take values in $[0, 1]$. The unit for code rates and channel capacities will be *bits*.

Intuitively, one would expect that $I(W) \approx 1$ iff $Z(W) \approx 0$, and $I(W) \approx 0$ iff $Z(W) \approx 1$. The following bounds, proved in the Appendix, make this precise.

Proposition 1: For any B-DMC W, we have

$$I(W) \ge \log \frac{2}{1 + Z(W)}, \tag{1}$$

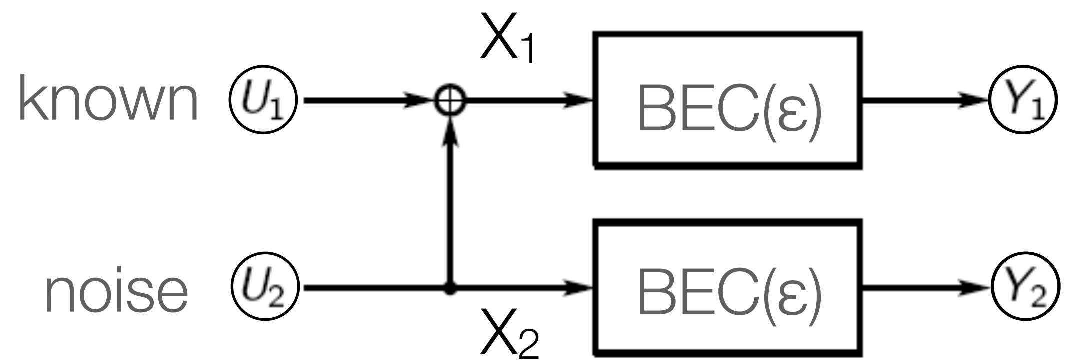
$$I(W) \le \sqrt{1 - Z(W)^2}. \tag{2}$$

The symmetric capacity $I(W)$ equals the Shannon capacity when W is a symmetric channel, i.e., a channel for which there exists a permutation π of the output alphabet $\mathcal{Y}$ such that (i) $\pi^{-1} = \pi$ and (ii) $W(y|1) = W(\pi(y)|0)$ for all $y \in \mathcal{Y}$. The binary symmetric channel (BSC) and the binary erasure channel (BEC) are examples of symmetric channels. A BSC is a B-DMC W with $\mathcal{Y} = \{0, 1\}$, $W(0|0) = W(1|1)$, and $W(1|0) = W(0|1)$. A B-DMC W is called a BEC if for each $y \in \mathcal{Y}$, either $W(y|0)W(y|1) = 0$ or $W(y|0) = W(y|1)$. In the latter case, y is said to be an erasure symbol. The sum of $W(y|0)$ over all erasure symbols y is called the erasure probability of the BEC.

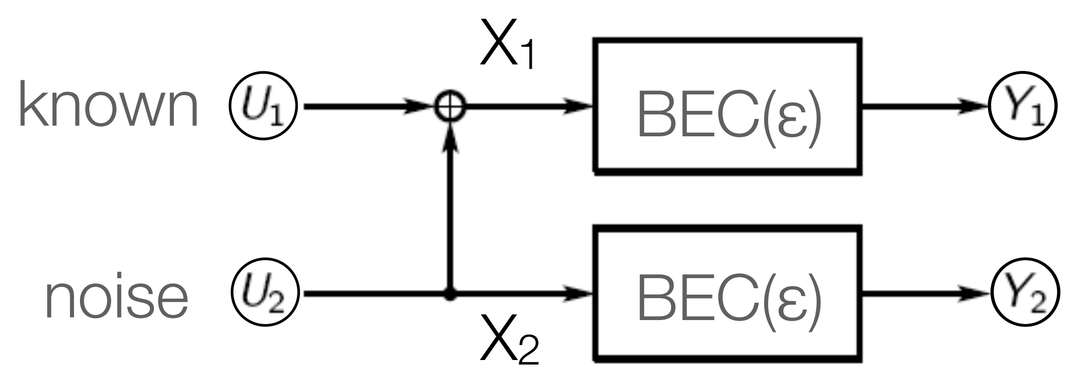
E. Arıkan is with the Department of Electrical-Electronics Engineering, Bilkent University, Ankara, 06800, Turkey (e-mail: arikan@ee.bilkent.edu.tr)

This work was supported in part by The Scientific and Technological Research Council of Turkey (TÜBITAK) under Project 107E216 and in part by the European Commission FP7 Network of Excellence NEWCOM++ under contract 216715.

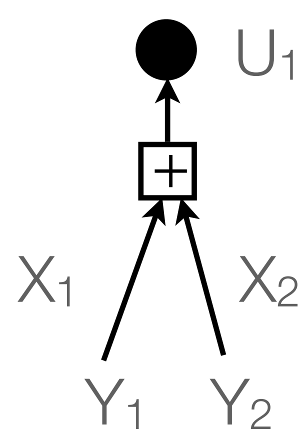
Channel Polarisation



Channel Polarisation



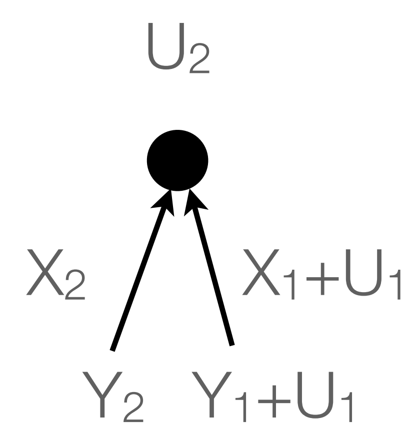
$U_1 = X_1 + X_2$; observe Y_1 and Y_2 $U_2 = X_2$; $U_2 = X_1 + U_1$; observe Y_1 and Y_2



$1-(1-\varepsilon)^2$
much worse

ε

parity-check node



ε^2 much better

ε

repetition code

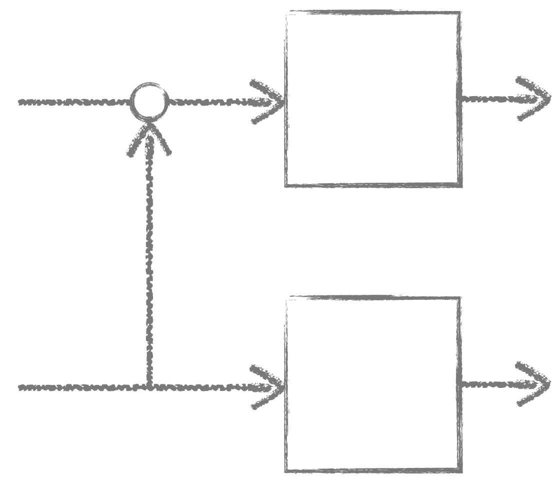
total capacity = $(1-\varepsilon)^2+1-\varepsilon^2=2(1-\varepsilon)$

$$I(U_1, U_2; Y_1, Y_2) = I(X_1, X_2; Y_1, Y_2) = I(X_1; Y_1) + I(X_2; Y_2) = 2I(W)$$

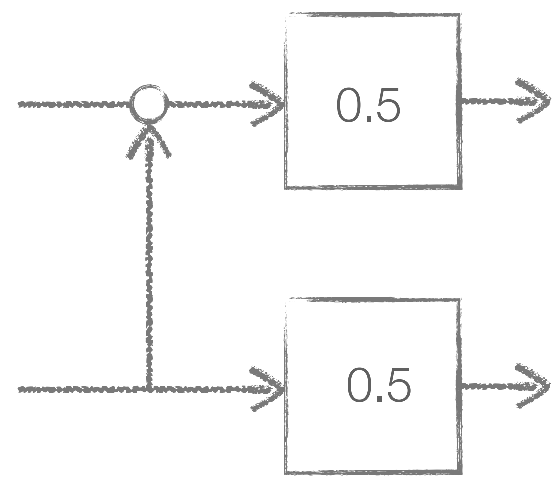
$$I(U_1, U_2; Y_1, Y_2) = I(U_1; Y_1, Y_2) + I(U_2; Y_1, Y_2 | U_1)$$

Polar Construction

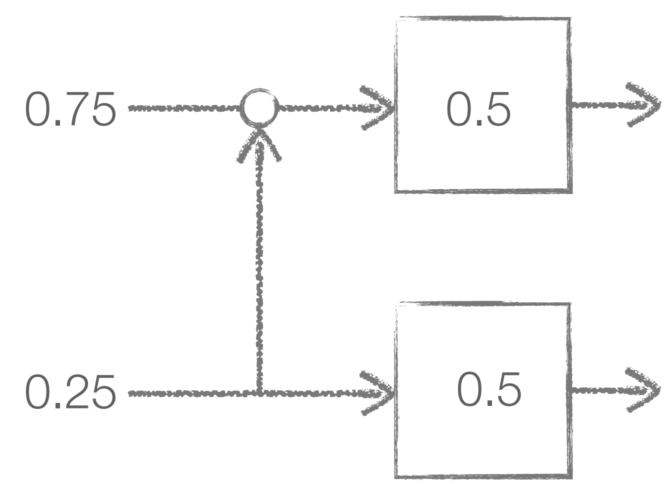
Polar Construction



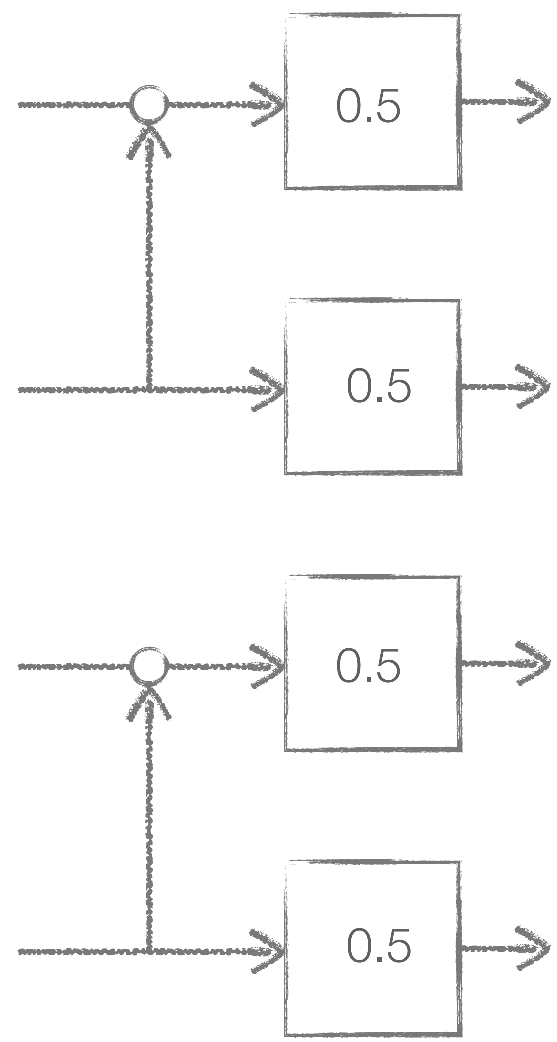
Polar Construction



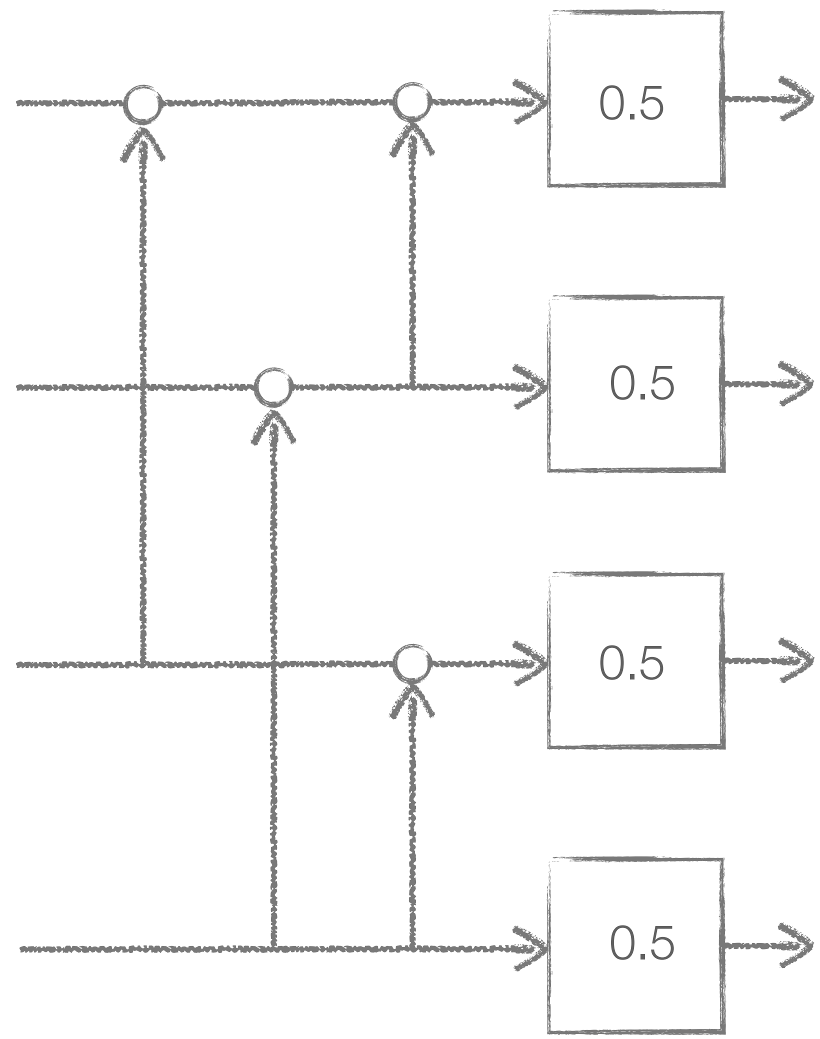
Polar Construction



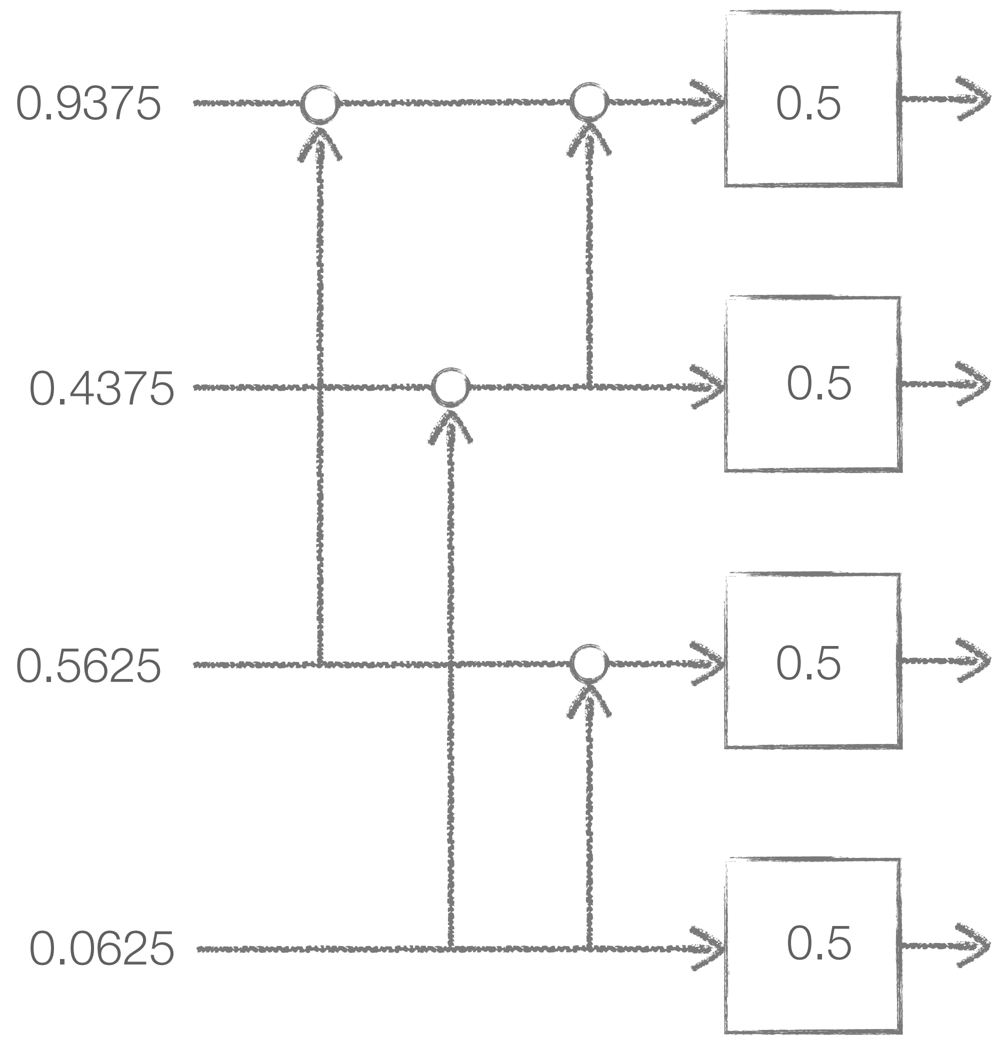
Polar Construction



Polar Construction



Polar Construction

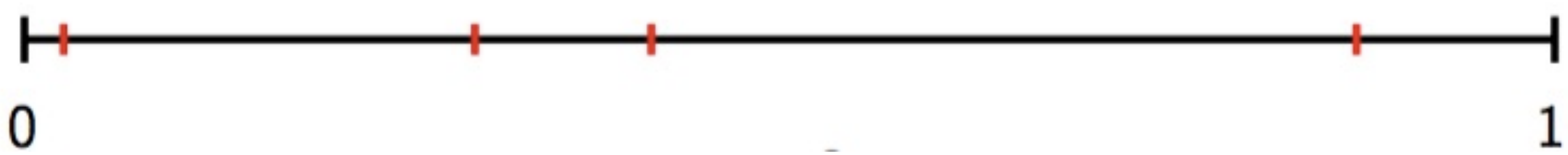


Channel Polarization

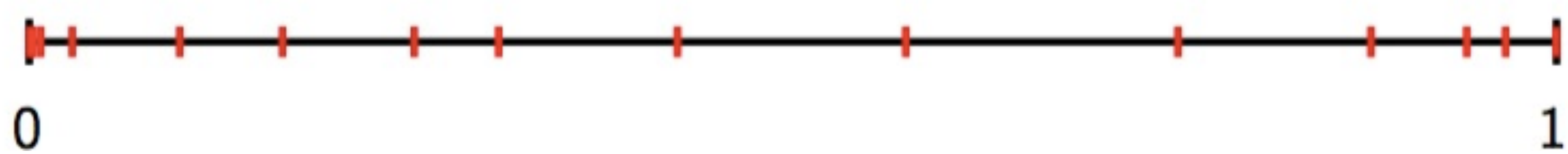
Channel Polarization

Channel Polarization

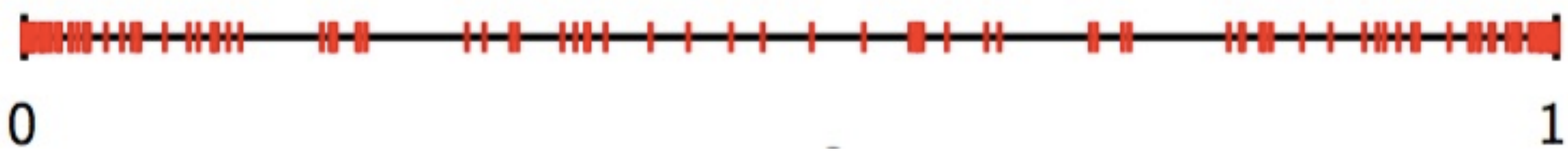
Channel Polarization



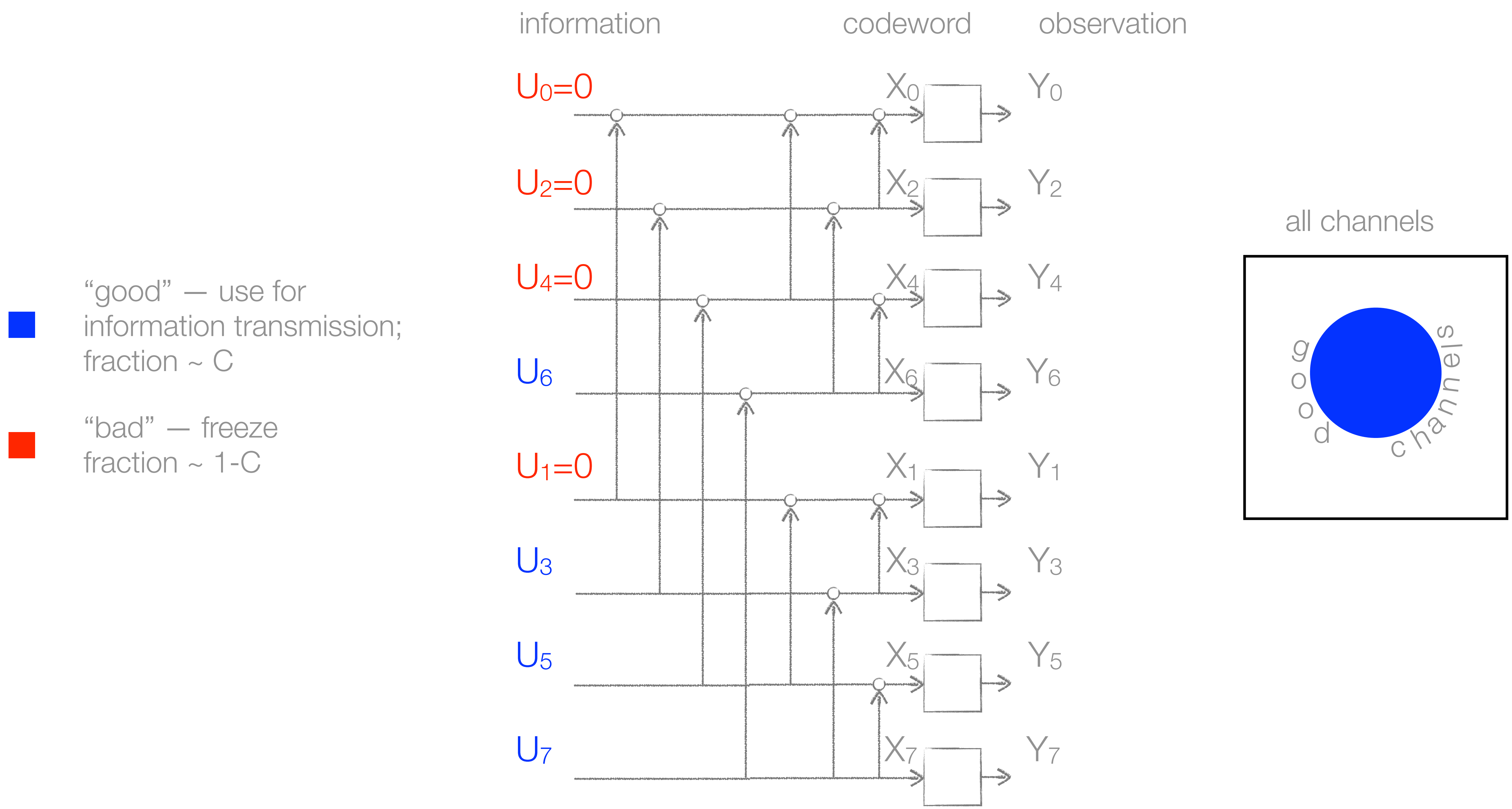
Channel Polarization



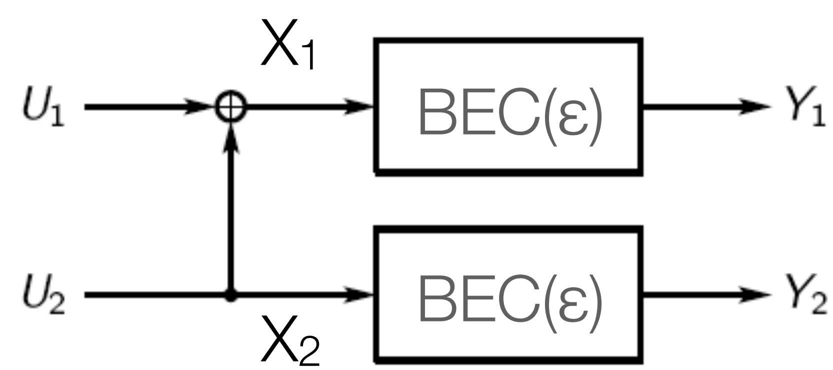
Channel Polarization



Polar Code — Polarization Effect and Set of Good Channels



Polar Codes — Key Ideas



$$I(U_1, U_2; Y_1, Y_2) = I(X_1, X_2; Y_1, Y_2) = I(X_1; Y_1) + I(X_2; Y_2) = 2I(W)$$

$$I(U_1, U_2; Y_1, Y_2) = I(U_1; Y_1, Y_2) + I(U_2; Y_1, Y_2 | U_1)$$

strict polarisation at each step unless extreme points are reached

Spatially Coupled Codes

Spatially Coupled Ensembles Universally Achieve Capacity Under Belief Propagation

Shrinivas Kudekar, Tom Richardson, *Fellow, IEEE*, and Rüdiger L. Urbanke

***Abstract*—We investigate spatially coupled code ensembles. For transmission over the binary erasure channel, it was recently shown that spatial coupling increases the *belief propagation* threshold of the ensemble to essentially the *maximum a priori* threshold of the underlying component ensemble. This explains why convolutional LDPC ensembles, originally introduced by Felström and Zigangirov, perform so well over this channel. We show that the equivalent result holds true for transmission over general binary-input memoryless output-symmetric channels. More precisely, given a desired error probability and a gap to capacity, we can construct a spatially coupled ensemble that fulfills these constraints *universally* on this class of channels under belief propagation decoding. In fact, most *codes* in this ensemble have this property. The quantifier *universal* refers to the *single* ensemble/code that is good for *all* channels but we assume that the channel is known at the receiver. The key technical result is a proof that, under belief-propagation decoding, spatially coupled ensembles achieve essentially the *area threshold* of the underlying uncoupled ensemble. We conclude by discussing some interesting open problems.**

***Index Terms*—Belief propagation (BP), capacity-achieving codes, channel coding, convolutional low-density parity-check (LDPC) codes, iterative decoding, LDPC codes, spatial coupling, spatially coupled codes, threshold saturation.**

I. Introduction

A. Historical Perspective

EVER since the publication of Shannon's seminal paper [1] and the introduction of the first coding schemes by Hamming [2] and Golay [3], coding theory has been concerned with finding low-delay and low-complexity capacity-achieving schemes. The interested reader can find an excellent historical review in [4]. Let us just briefly mention some of the highlights before focusing on those parts that are the most relevant for our purpose.

In the first 50 years, coding theory focused on the construction of *algebraic* coding schemes and algorithms that were capable of exploiting the algebraic structure. Two early highlights of this line of research were the introduction of the Bose–Chaudhuri–Hocquenghem (BCH) codes [5], [6] as well as the Reed–Solomon (RS) codes [7]. Berlekamp devised an efficient decoding algorithm [8], and this algorithm was then interpreted by Massey as an algorithm for finding the shortest feedback-shift register that generates a given sequence [9]. More recently, Sudan introduced a list decoding algorithm for RS codes that decodes beyond the guaranteed error-correcting radius [10]. Guruswami and Sudan improved upon this algorithm [11] and Koetter and Vardy showed how to handle soft information [12].

Another important branch started with the introduction of convolutional codes [13] by Elias and the introduction of *sequential decoding* algorithm by Wozencraft [14]. Viterbi introduced the Viterbi algorithm [15]. It was shown to be optimal by Forney [16] and Omura [17] and to be eminently *practical* by Heller [18], [19].

An important development in transmission over the continuous input, band-limited, additive white Gaussian noise channel was the invention of the *lattice codes*. It was shown in [20]–[24] that lattice codes achieve the Shannon capacity. A breakthrough in bandwidth-limited communications came about when Ungerboeck [25]–[27] invented a technique to combine coding and modulation. Ungerboeck's technique ushered in a new era of fast modems. The technique, called *trellis-coded modulation* (TCM), offered significant coding gains without compromising bandwidth efficiency by mapping binary code symbols, generated by a convolutional encoder, to a larger (nonbinary) signal constellation. In [28] and [29], Forney showed that lattice codes, as well as TCM schemes, might be generated by the same basic elements and the generalized technique was termed *coset-coding*.

Coming back to binary linear codes, in 1993, Berrou *et al.* [30] proposed *turbo* codes. These codes attain near-Shannon limit performance under low-complexity iterative decoding. Their remarkable performance leads to a flurry of research on the "turbo" principle. Around the same time, Spielman in his thesis [31], [32] and MacKay and Neal in [33]–[36], independently rediscovered low-density parity-check (LDPC) codes and iterative decoding, both introduced in Gallager's remarkable thesis [37]. Wiberg showed [38] that both turbo codes and LDPC codes fall under the umbrella of codes based on sparse graphs and that their iterative decoding algorithms are special cases of the *sum-product* algorithm. This line of research was formalized by Kschischang *et al.* who introduced the notion of *factor graphs* [39].

The next breakthrough in the design of codes (based on sparse graphs) came with the idea of using *irregular* LDPC codes by

Manuscript received April 28, 2012; revised January 17, 2013; accepted March 26, 2013. Date of publication September 05, 2013; date of current version November 19, 2013. This work was supported in part by the U.S. Department of Energy, in part by Los Alamos National Laboratory under Contract DE-AC52-06NA25396, and in part by NMC via the NSF collaborative Grant CCF-0829945 on "Harnessing Statistical Physics for Computing and Communications." R. L. Urbanke was supported by the European project STAMINA 265496. This paper was presented at the 2012 IEEE International Symposium on Information Theory.

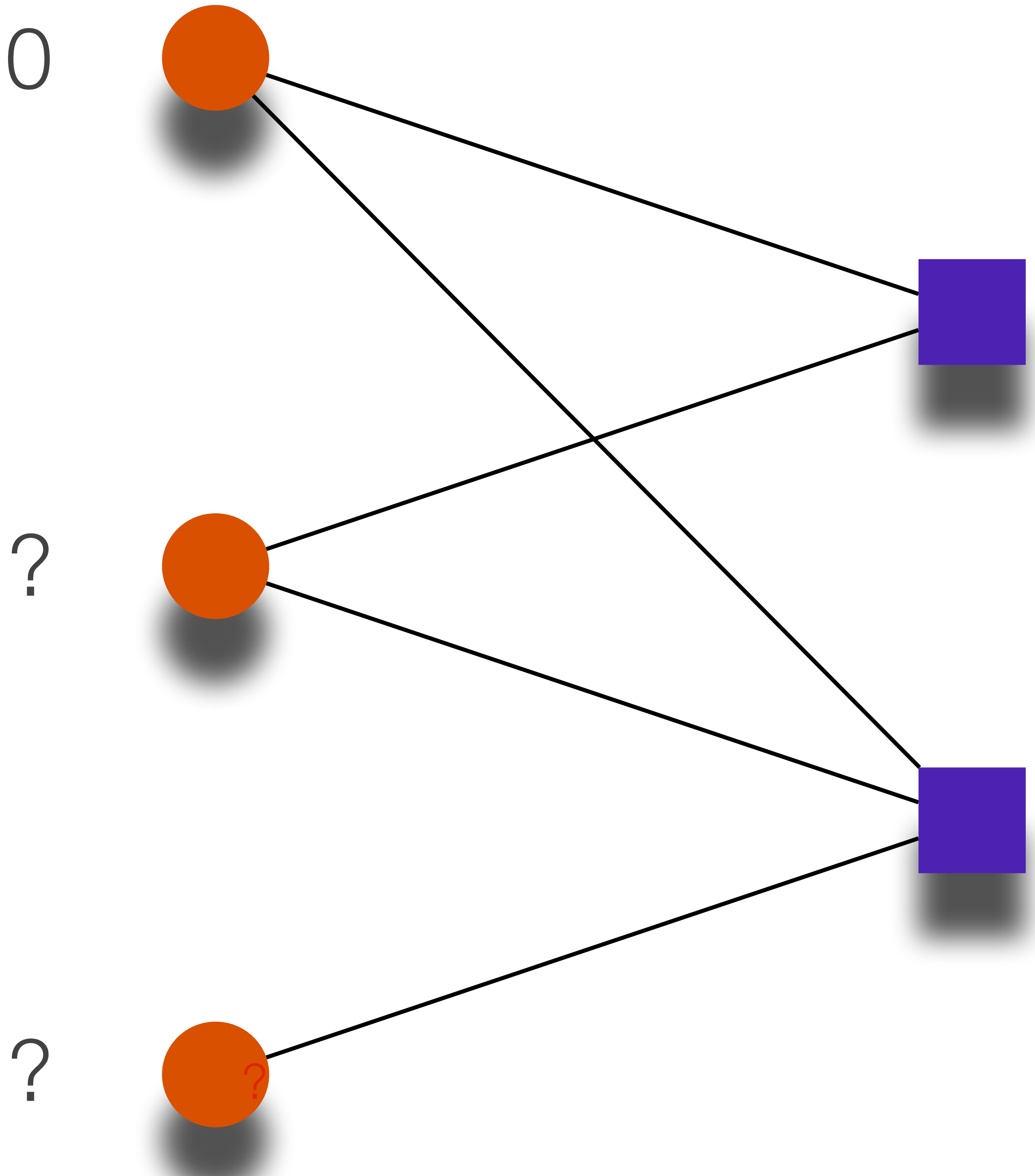
S. Kudekar and T. Richardson are with Qualcomm, Bridgewater, NJ 08807 USA (e-mail: skudekar@qti.qualcomm.com; tomr@qti.qualcomm.com).

R. L. Urbanke is with the School of Computer and Communication Sciences EPFL, Lausanne, Vaud, Switzerland (e-mail: ruediger.urbanke@epfl.ch).

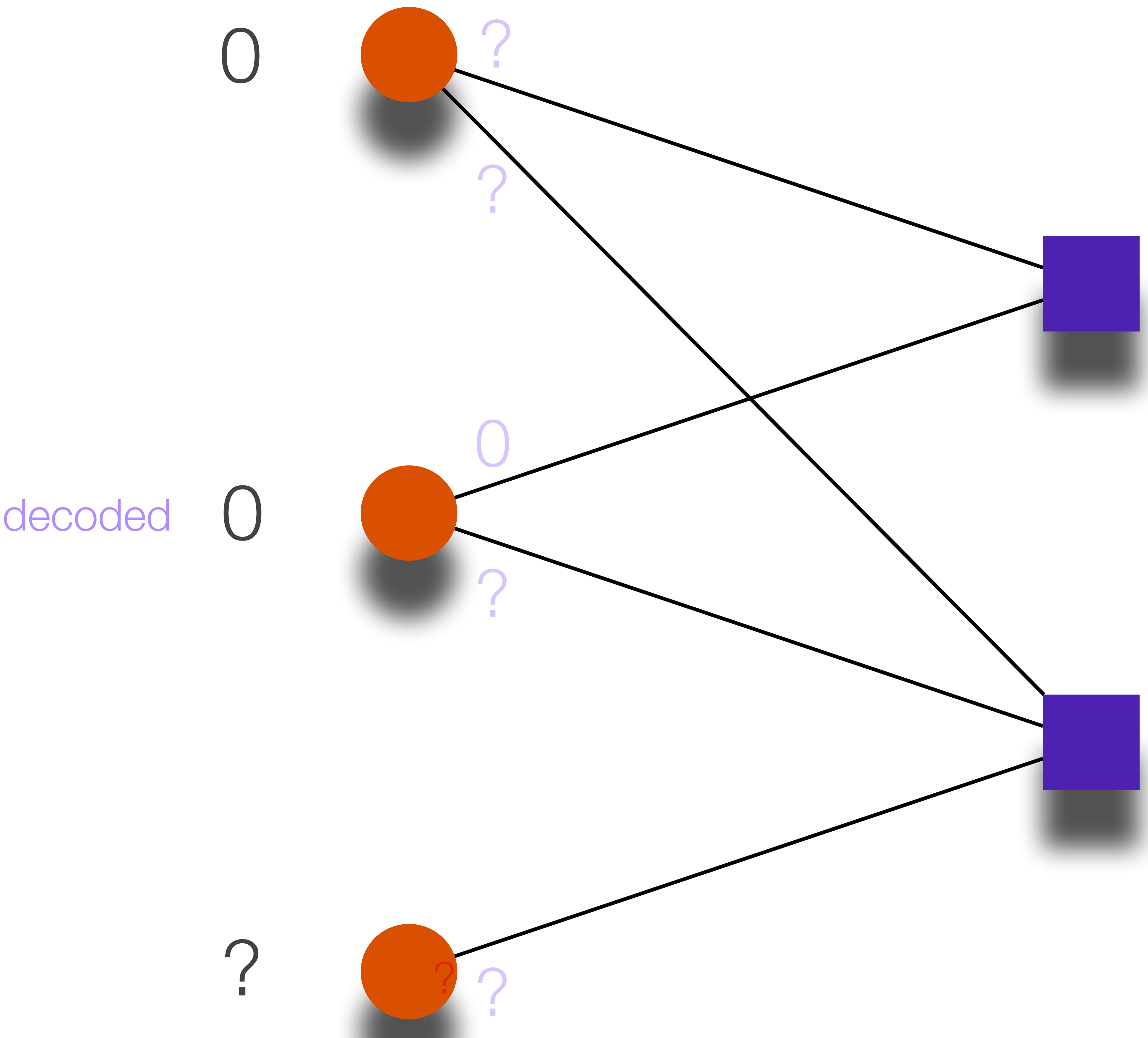
Communicated by E. Arıkan, Associate Editor for Coding Theory.

Digital Object Identifier 10.1109/TIT.2013.2280915

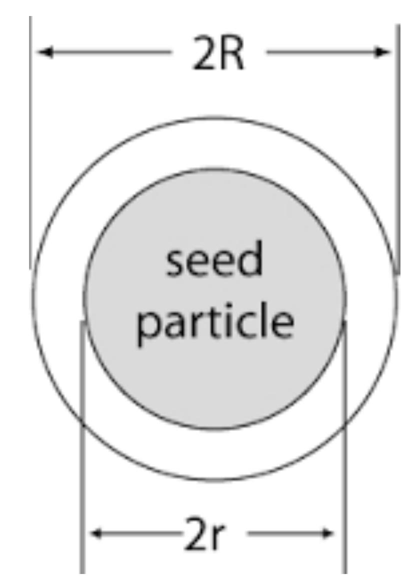
LDPC Codes and Message Passing



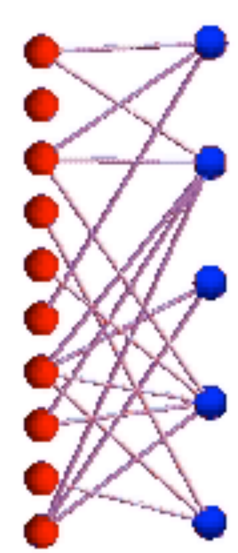
LDPC Codes and Message Passing

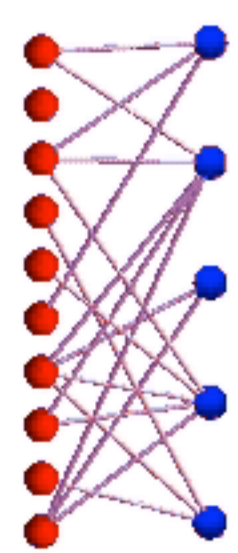


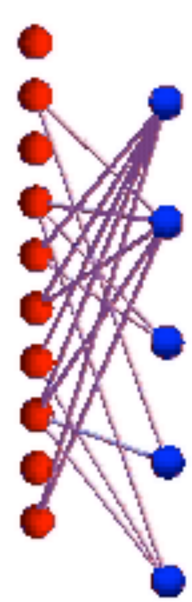
Metastability

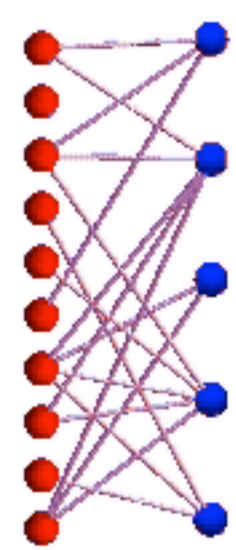


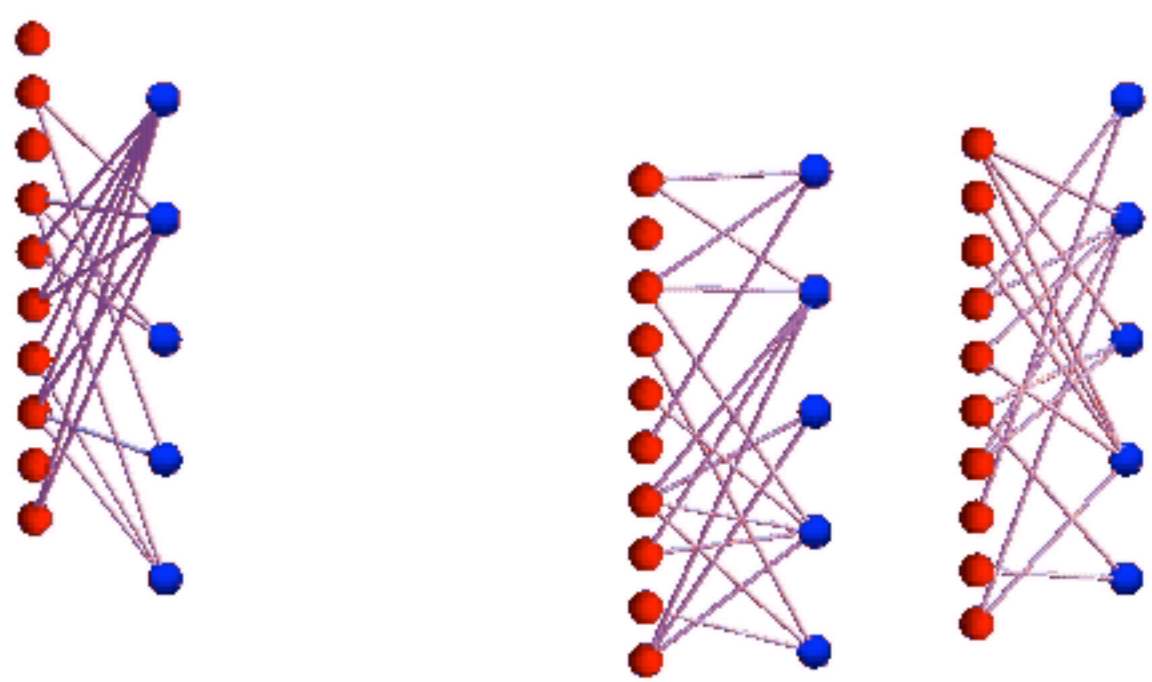
Sodium acetate, $C_2H_3NaO_2$,

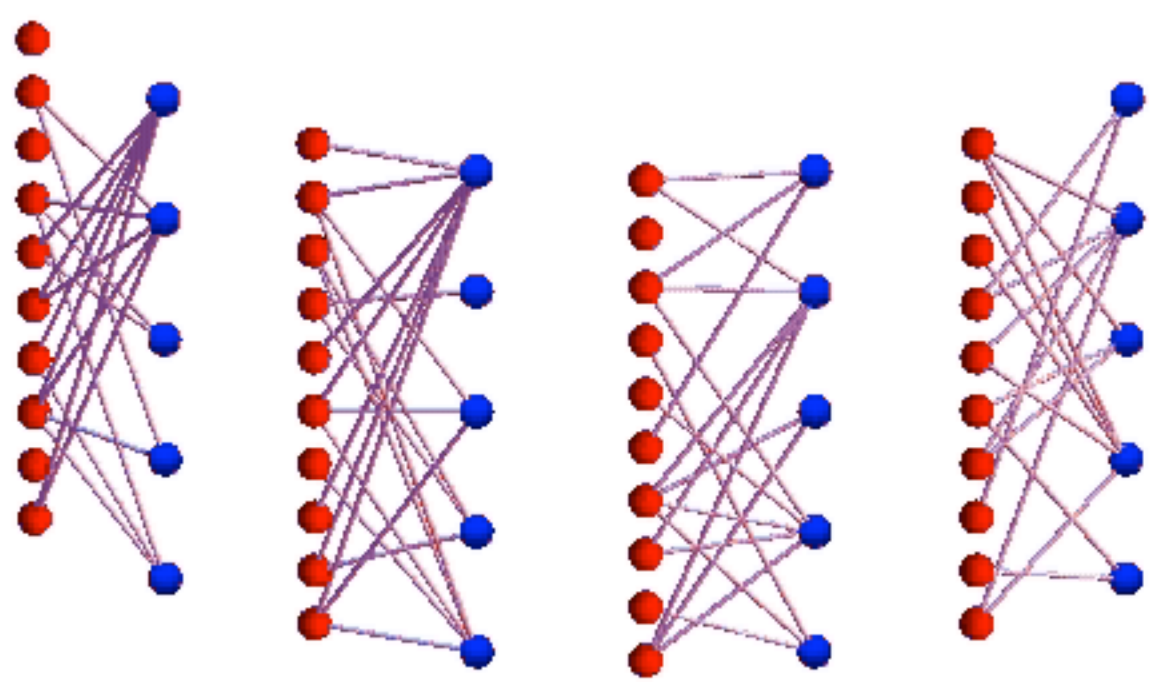


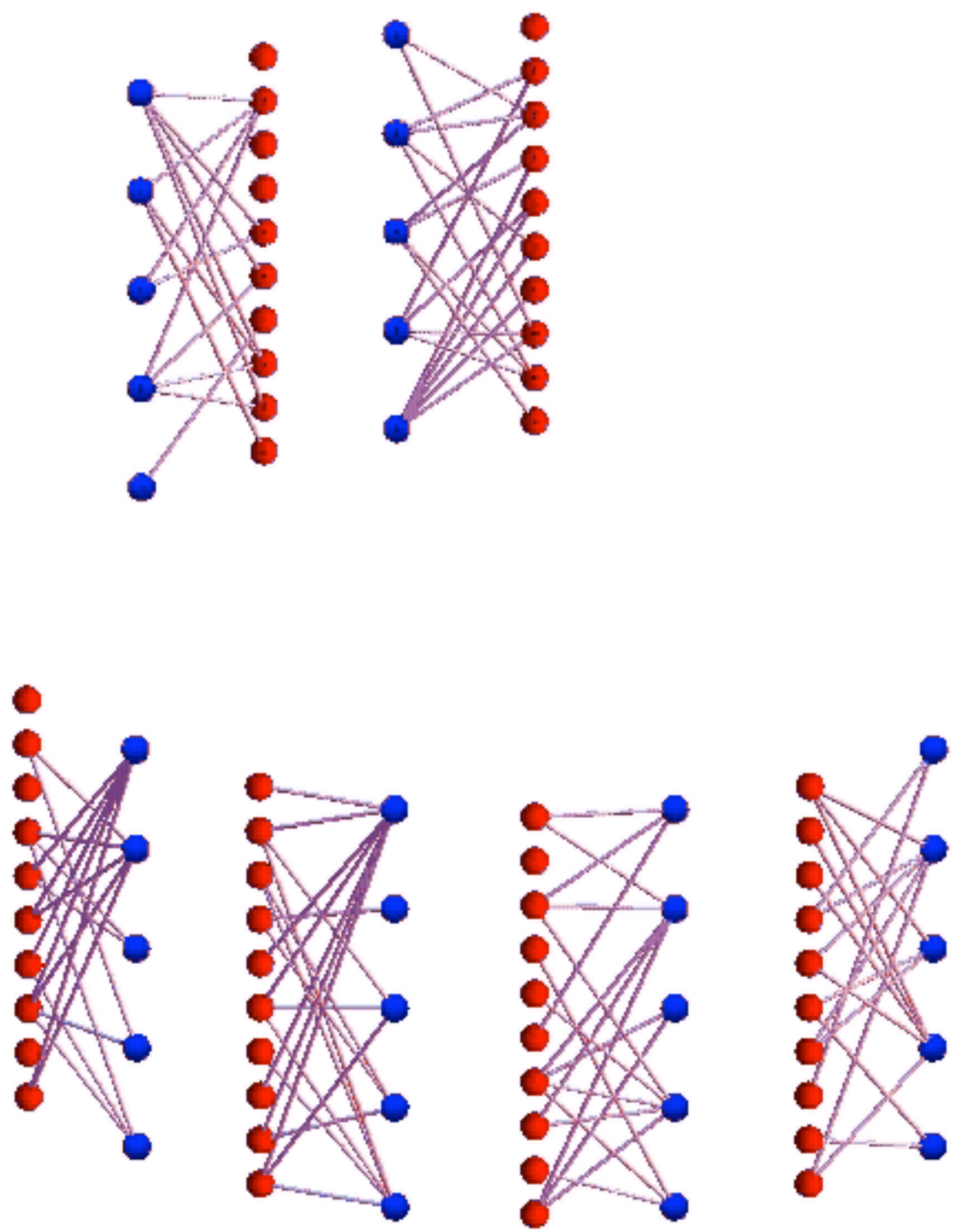


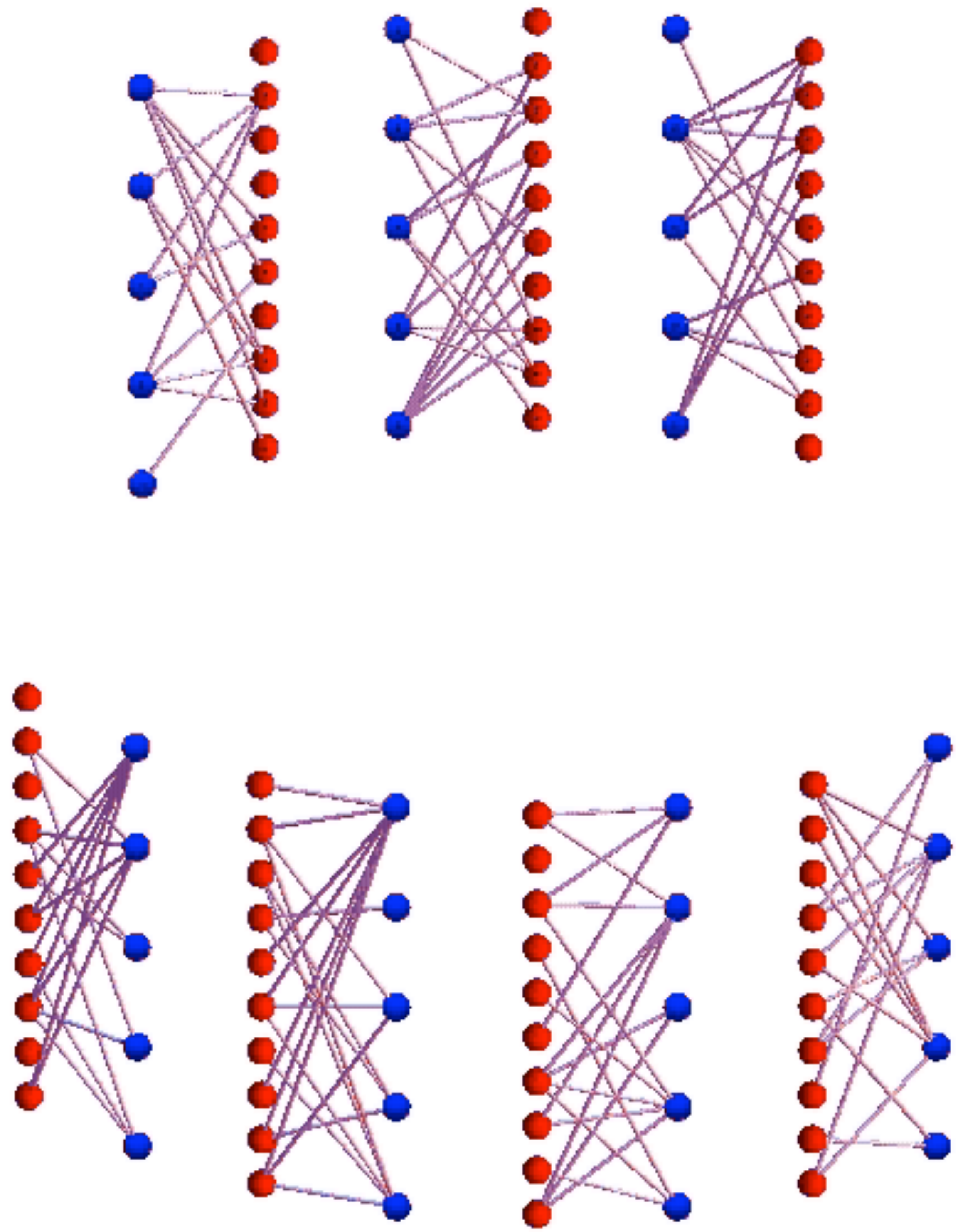


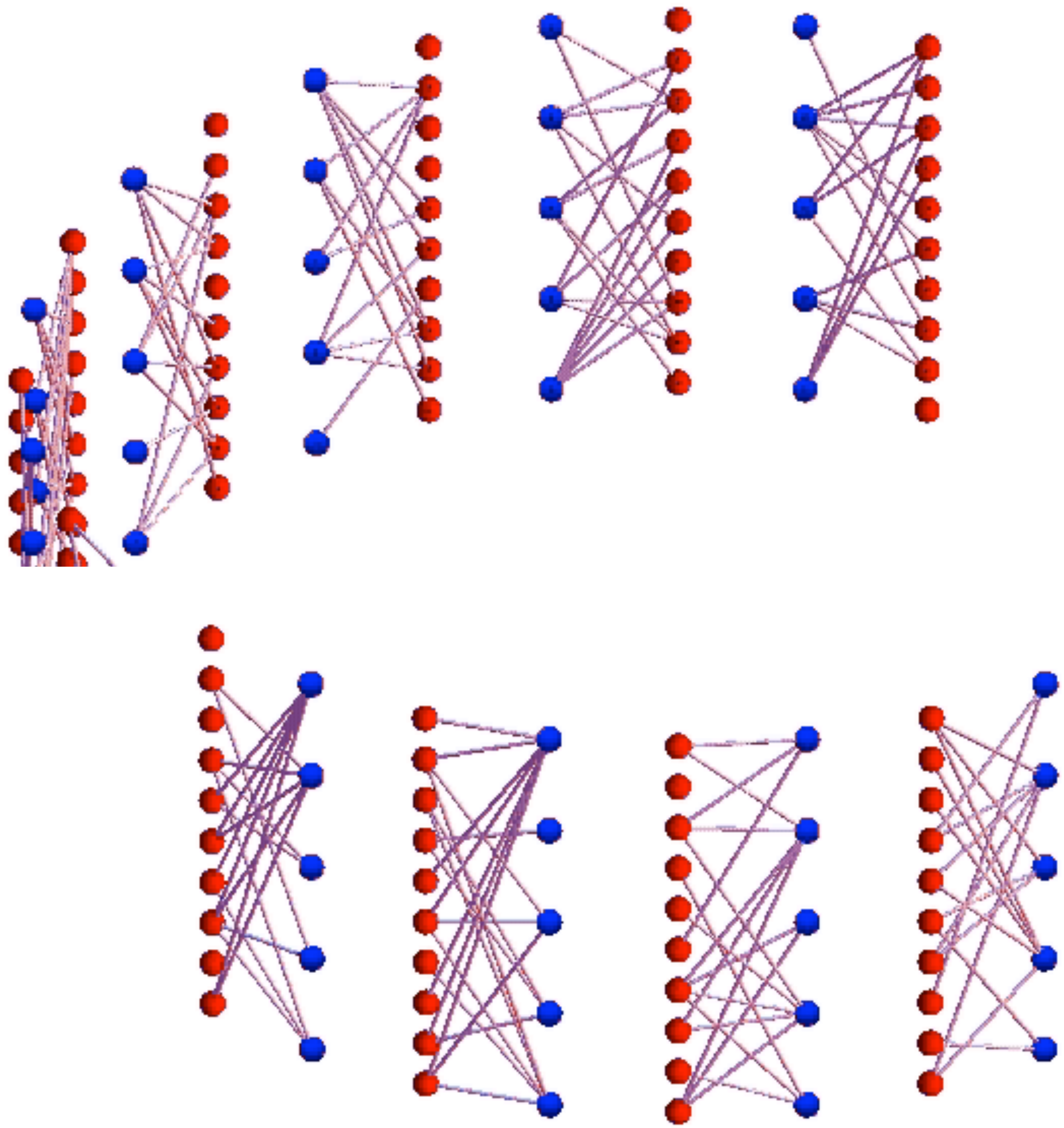


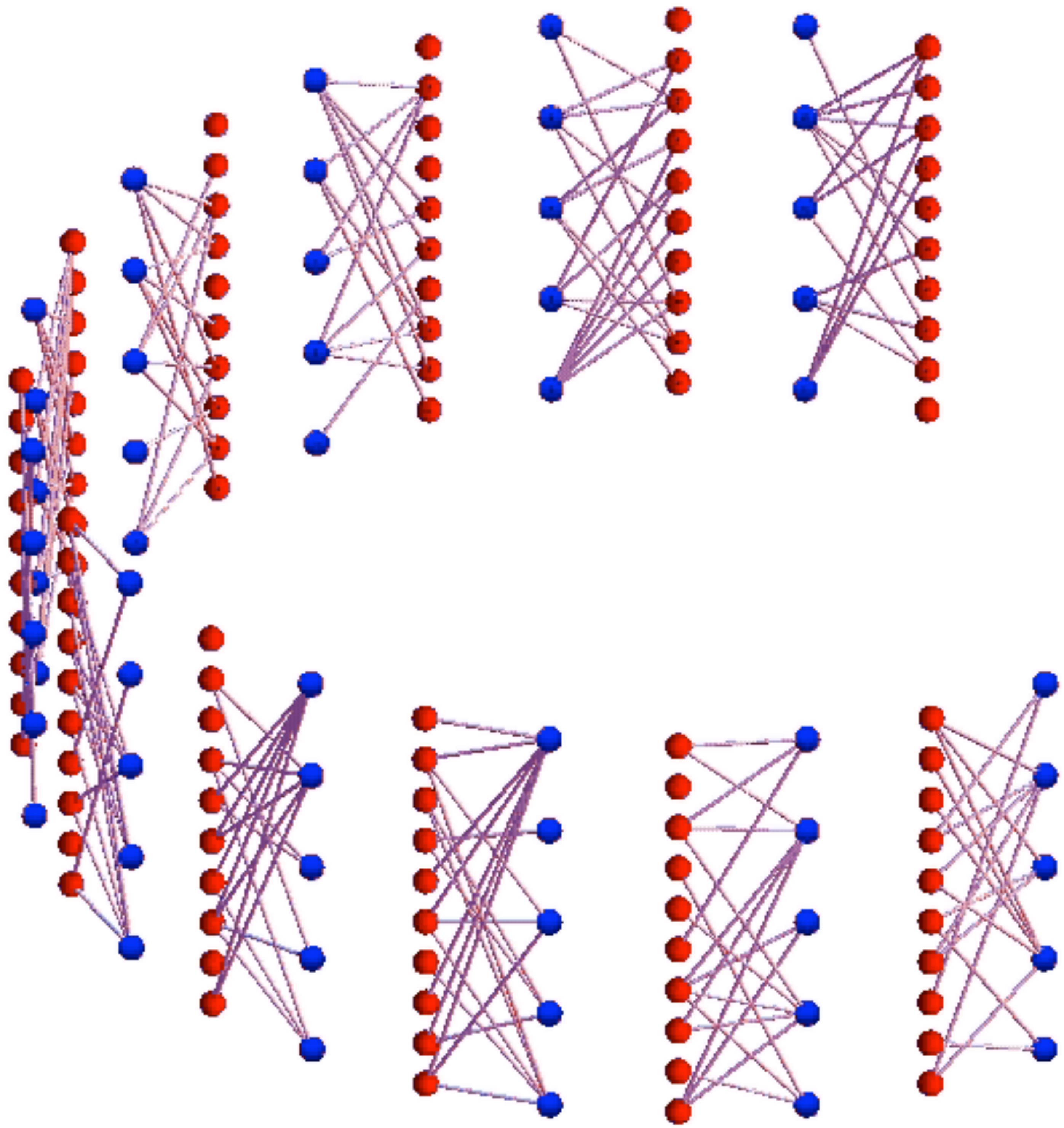


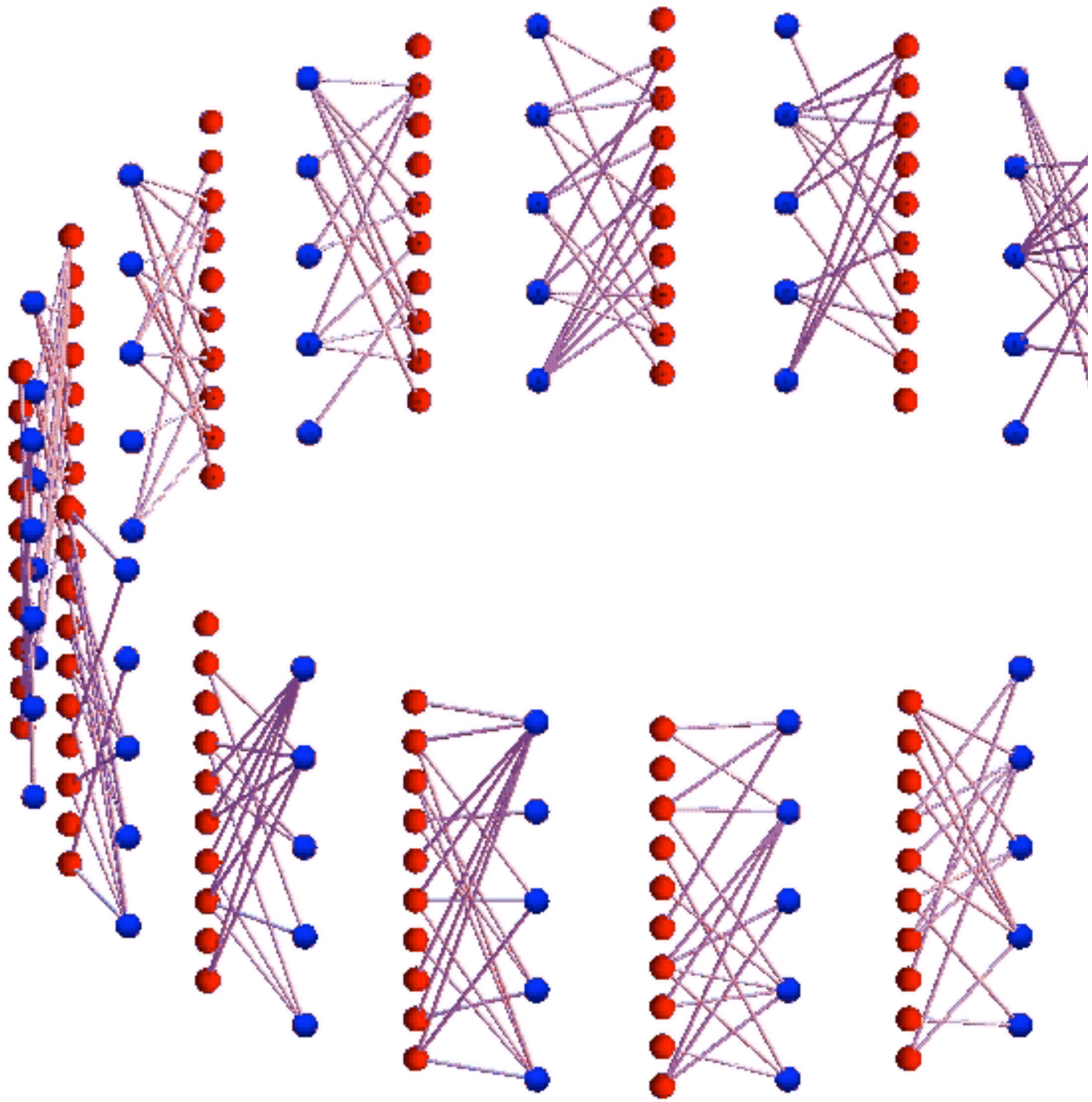


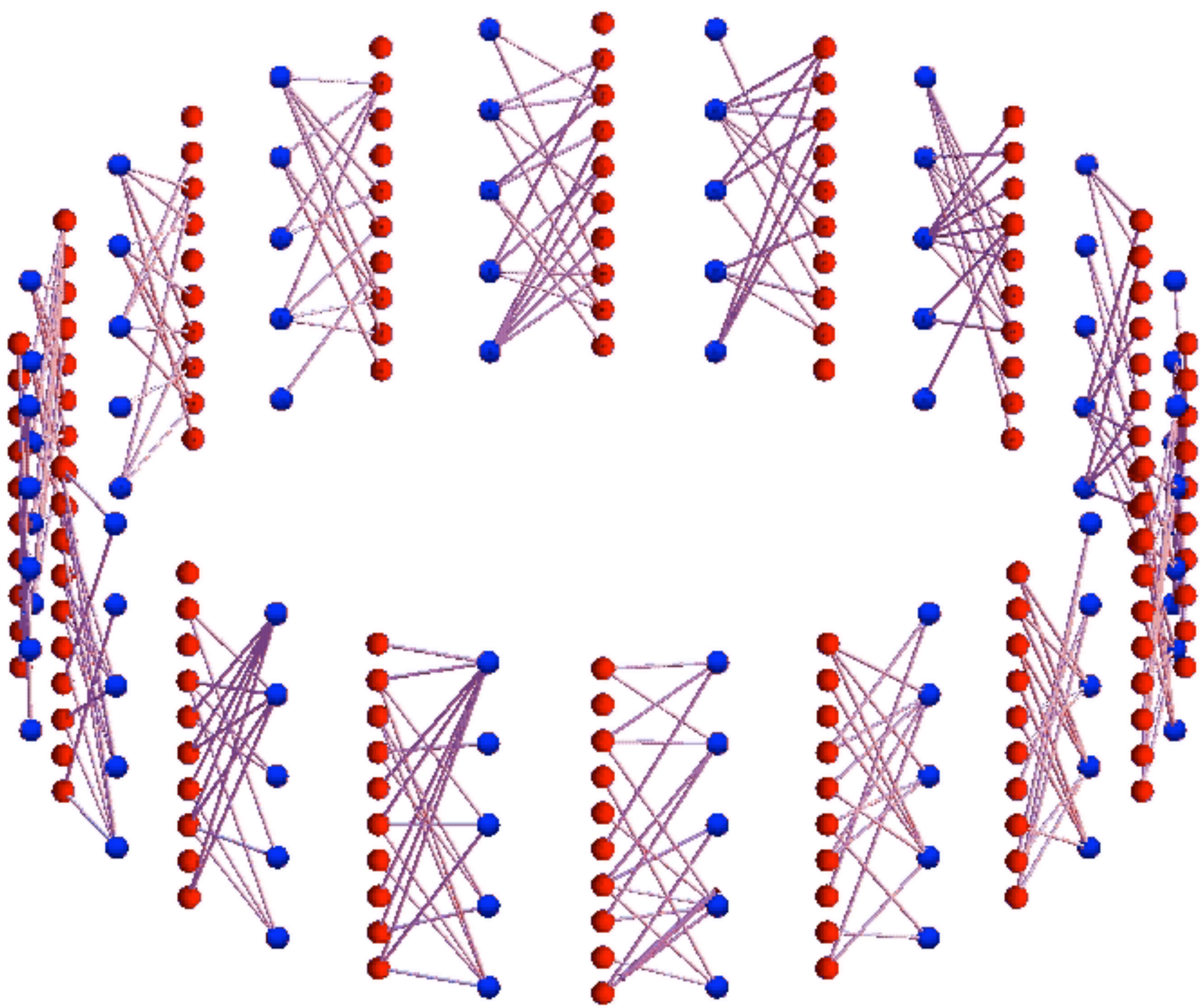


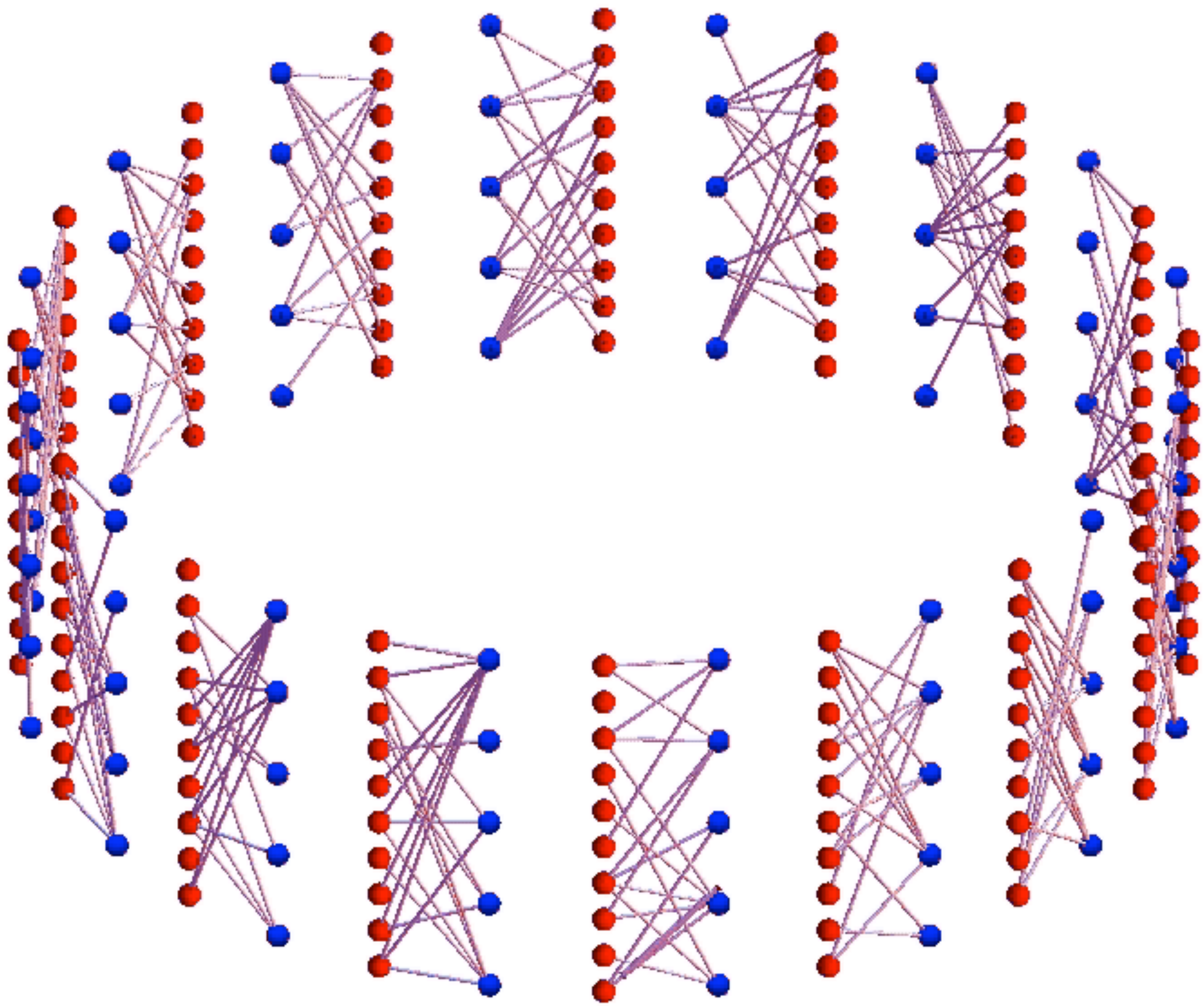


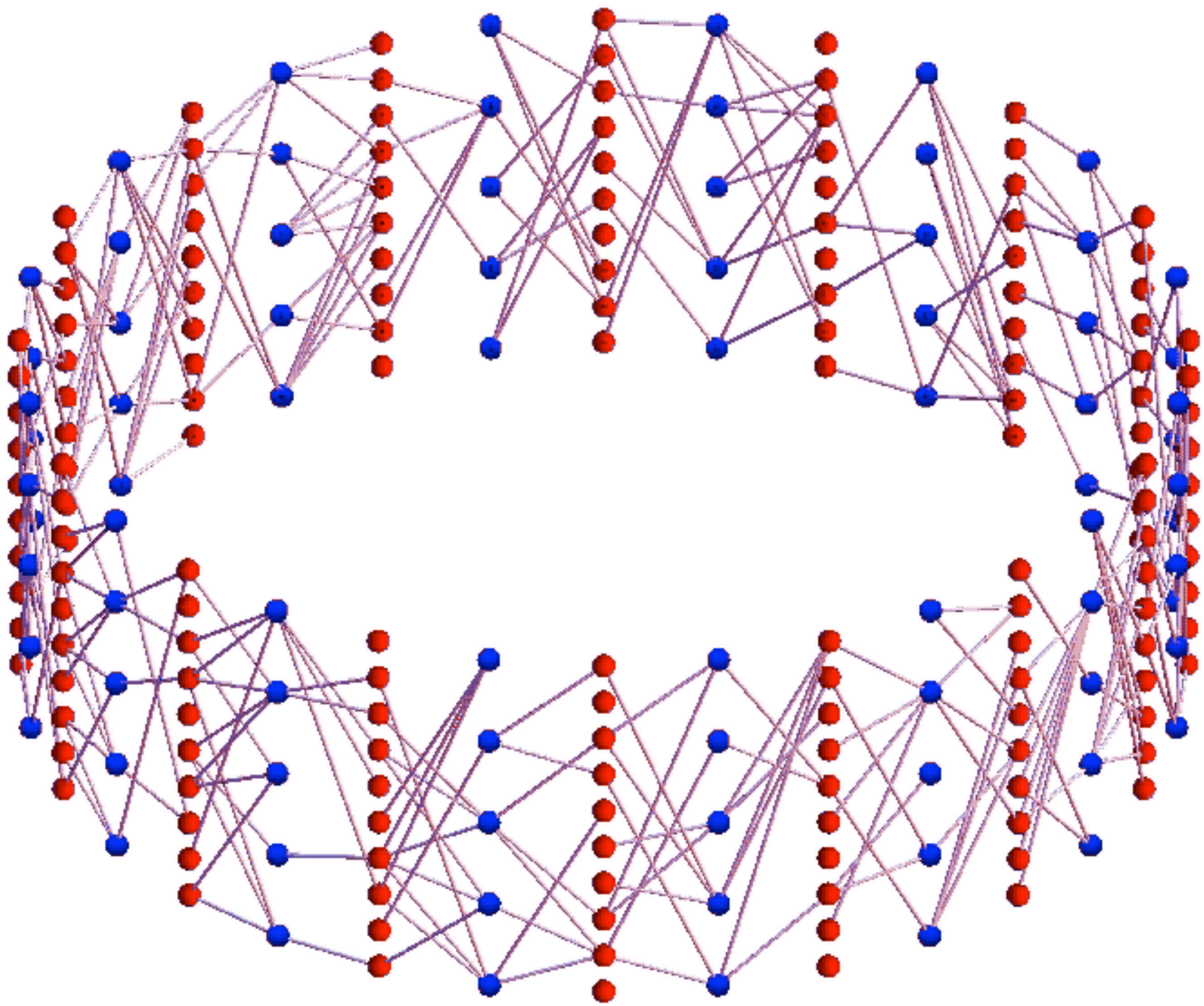


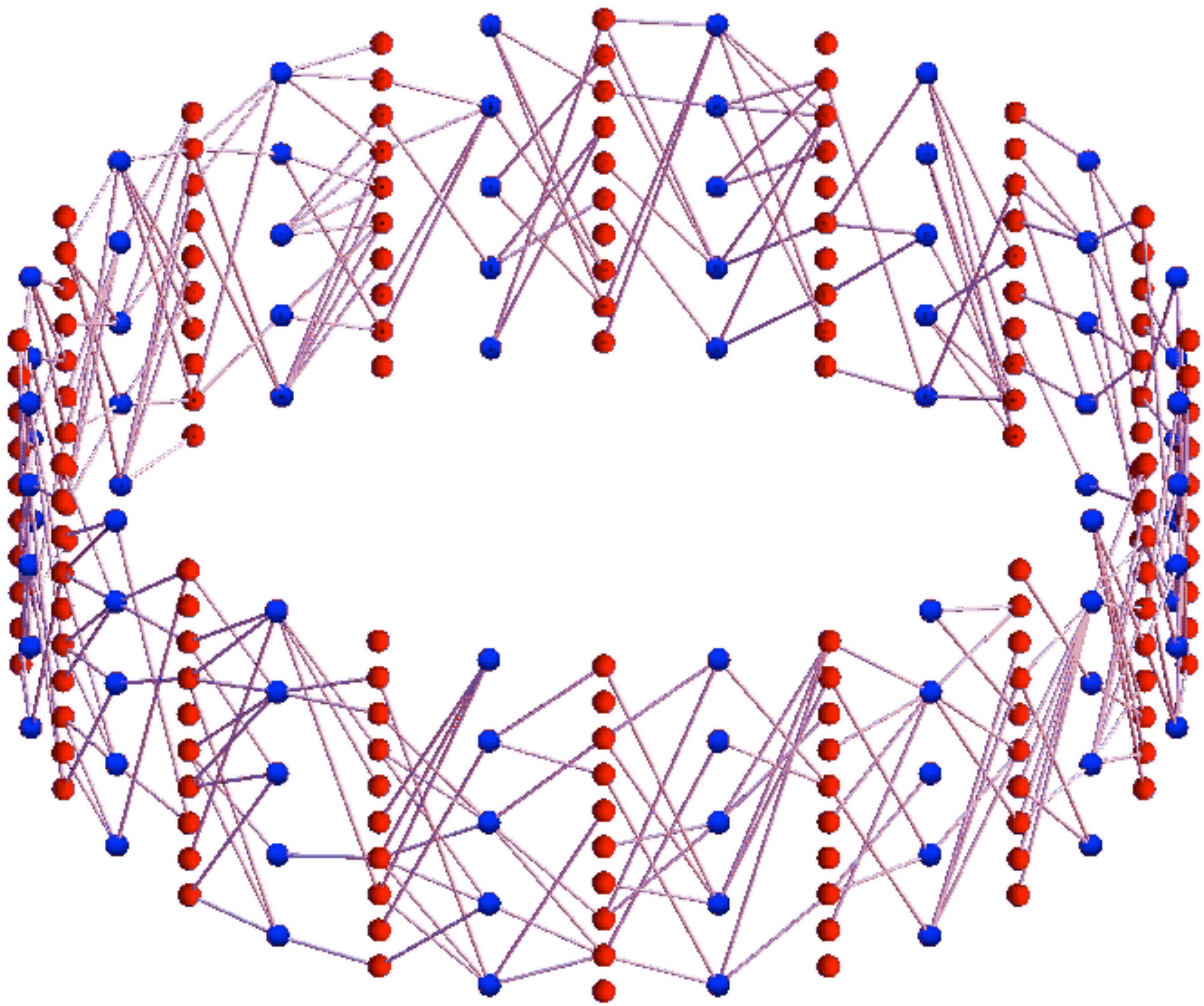


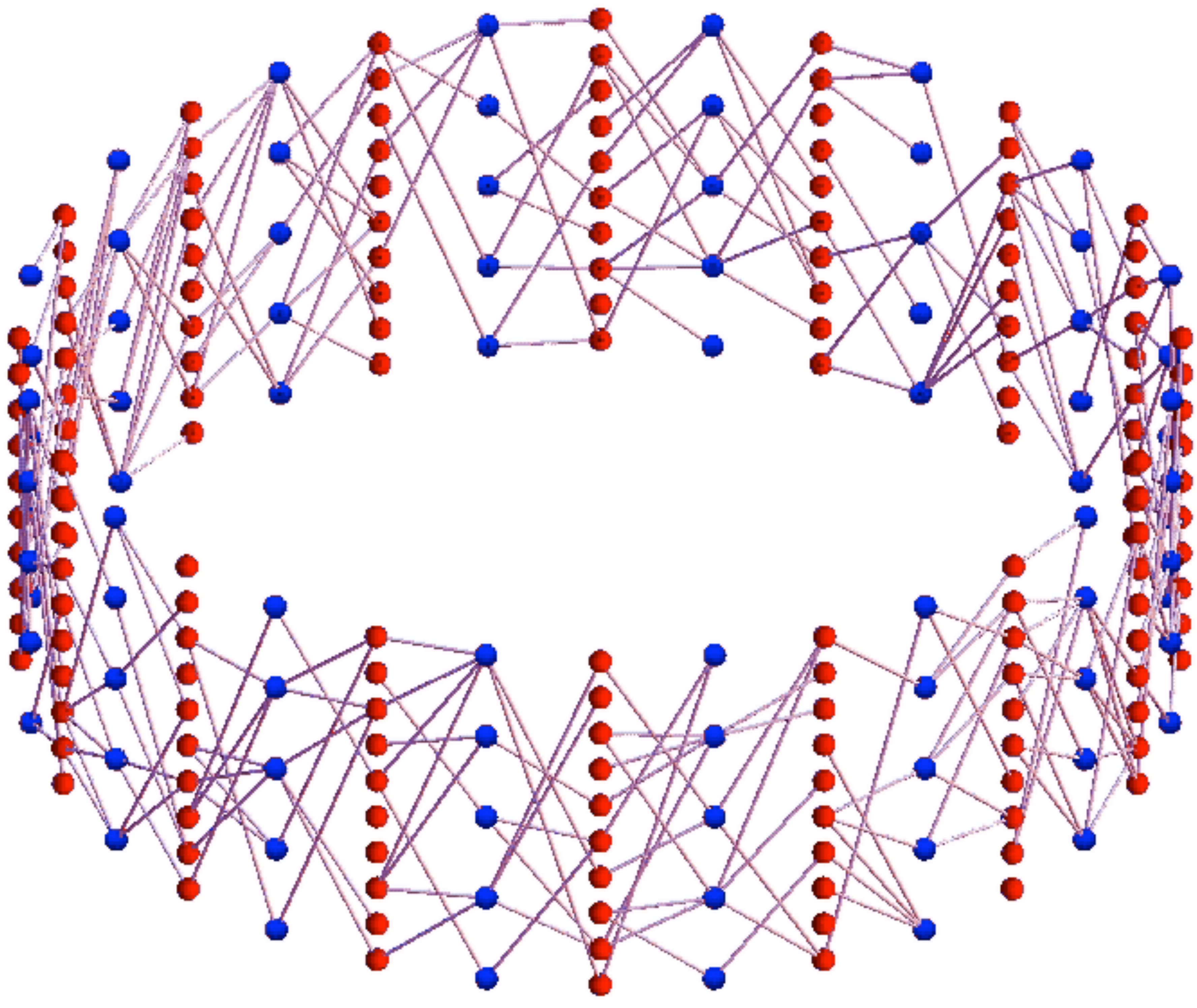


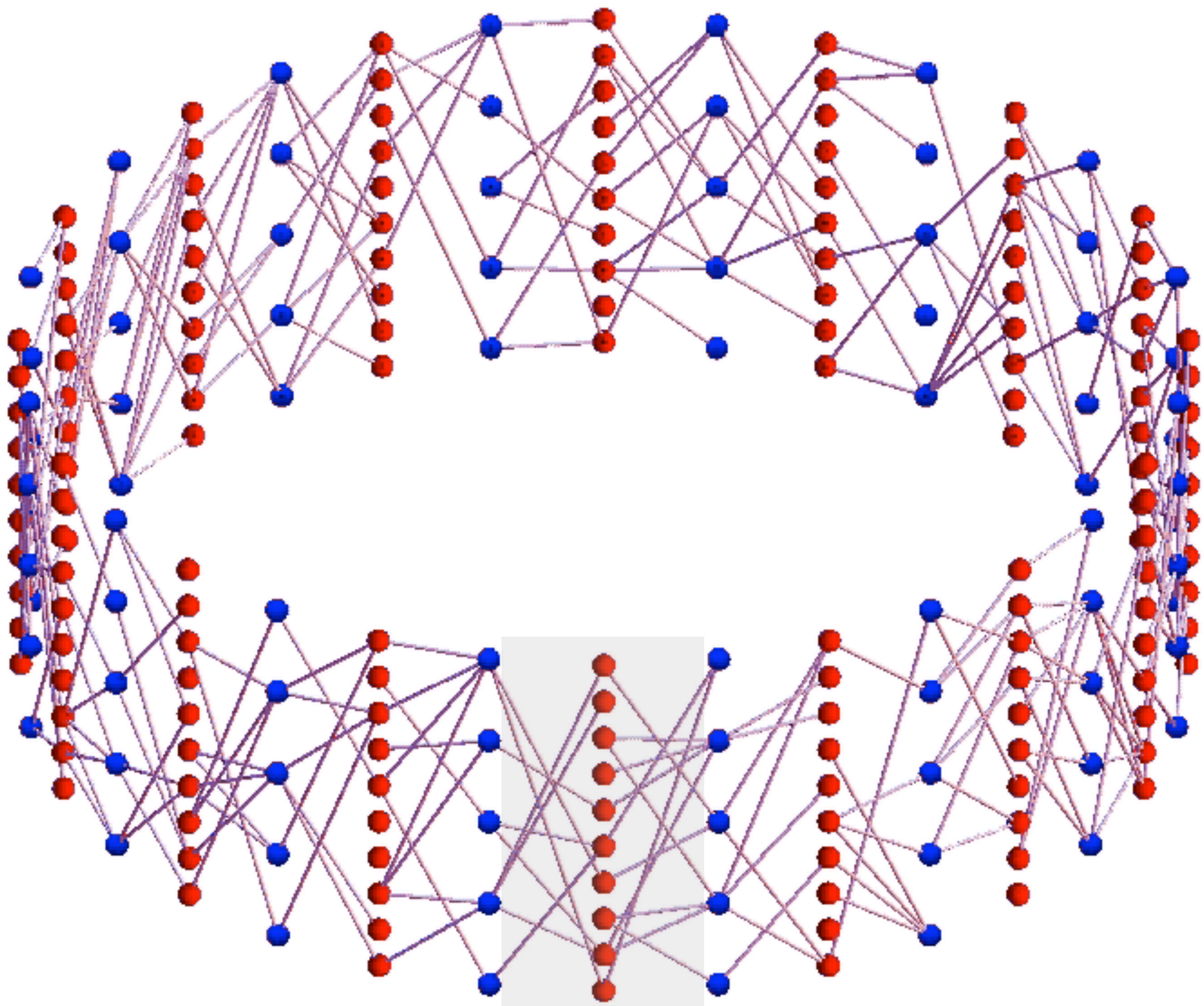


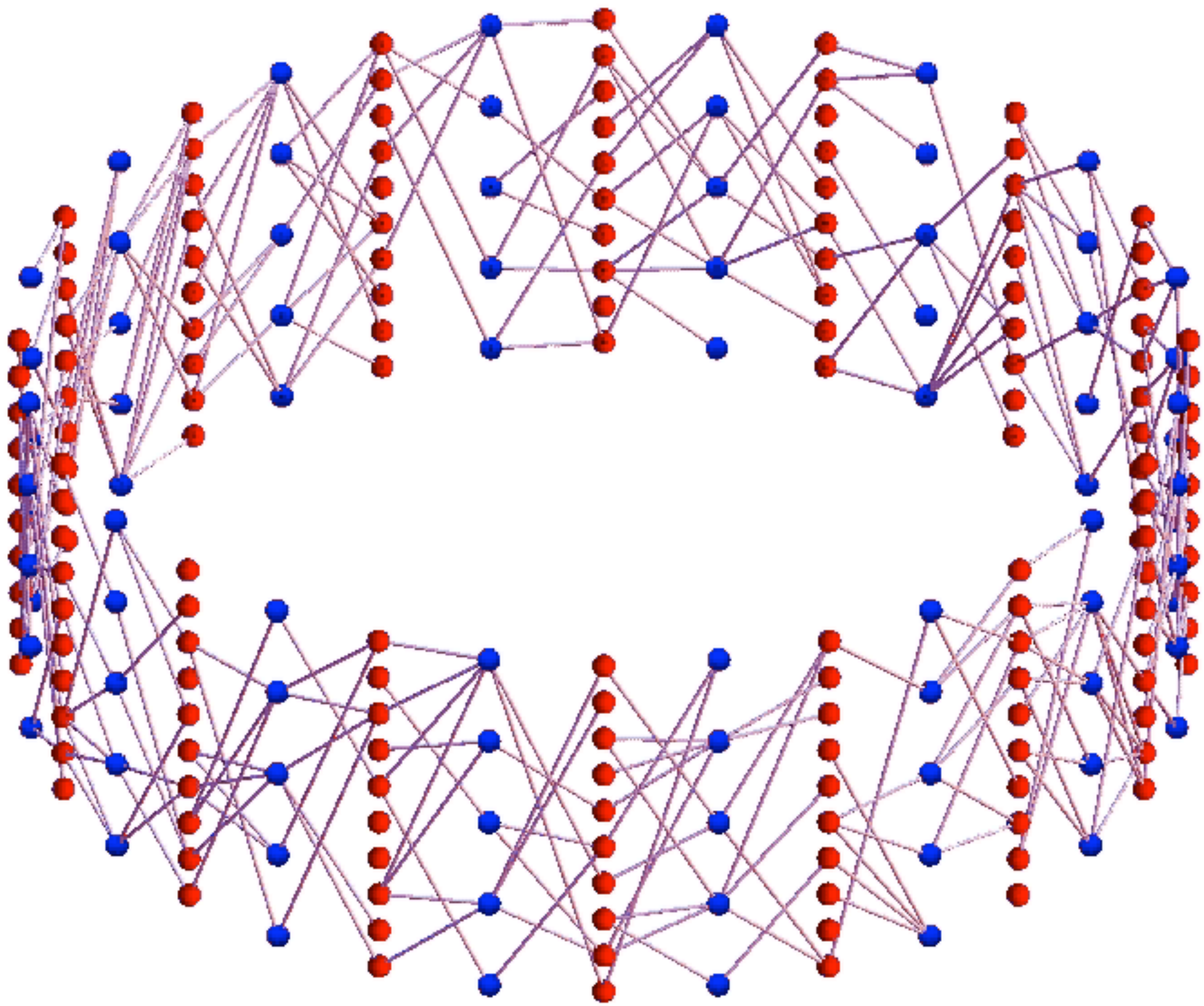


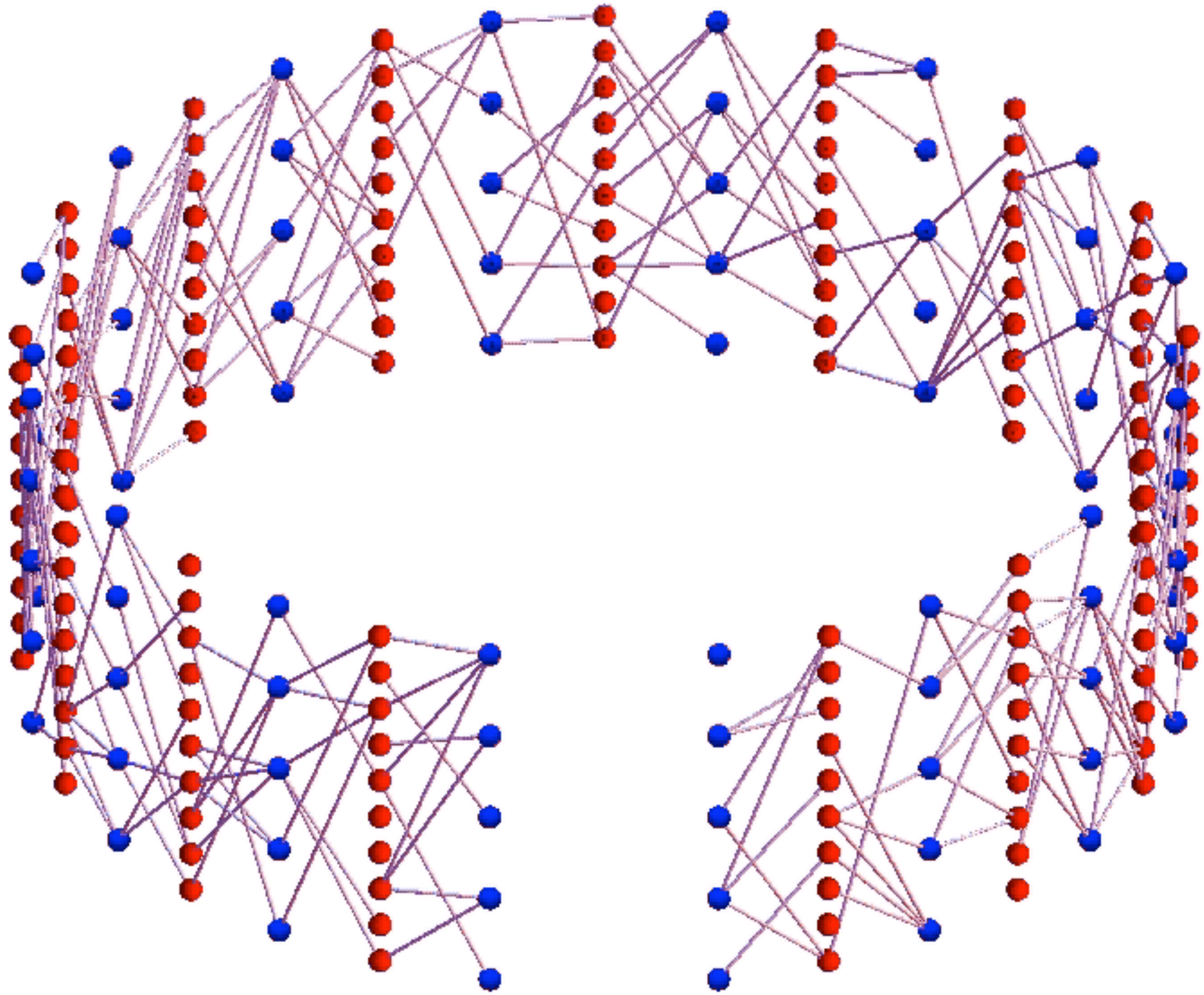


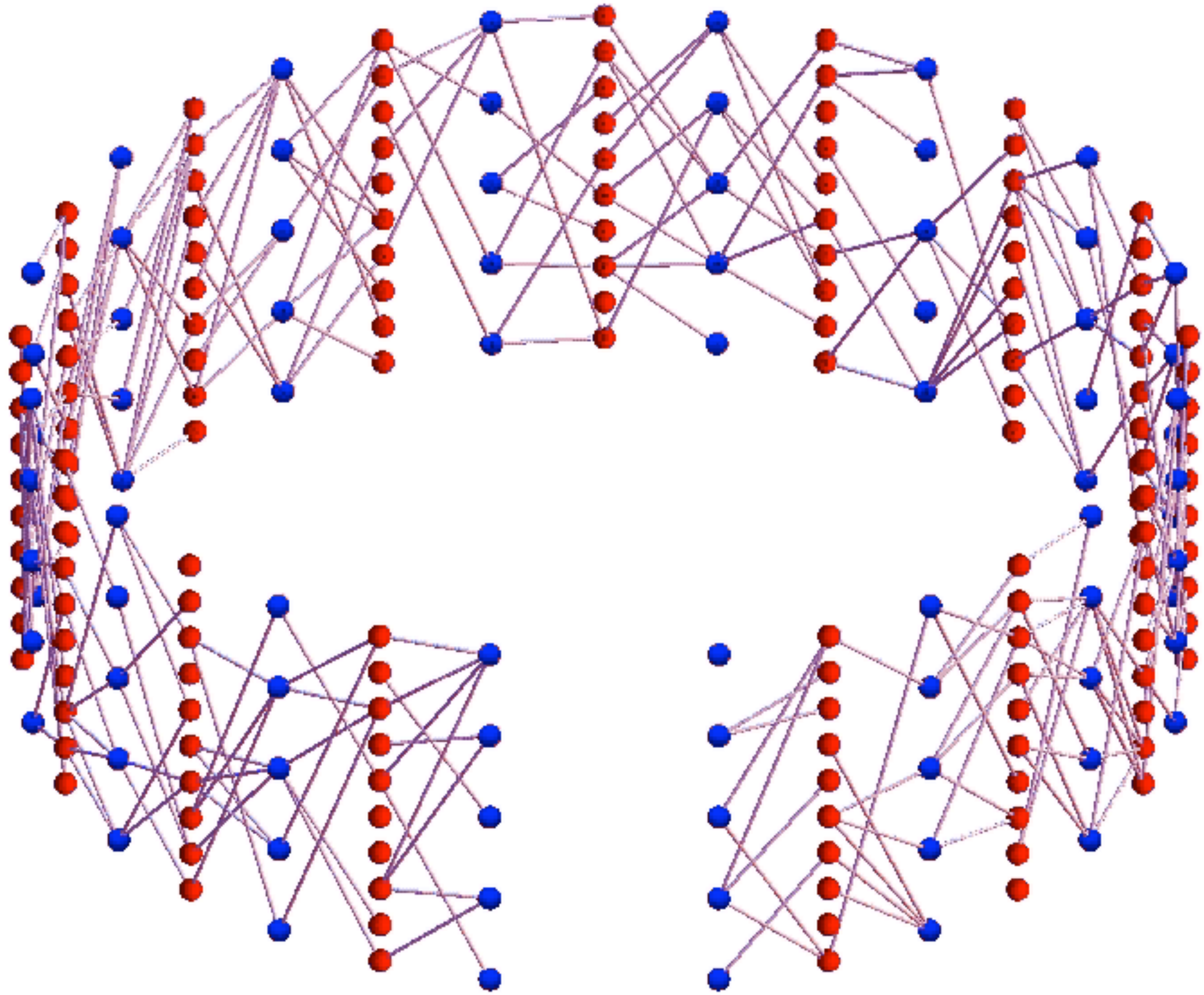


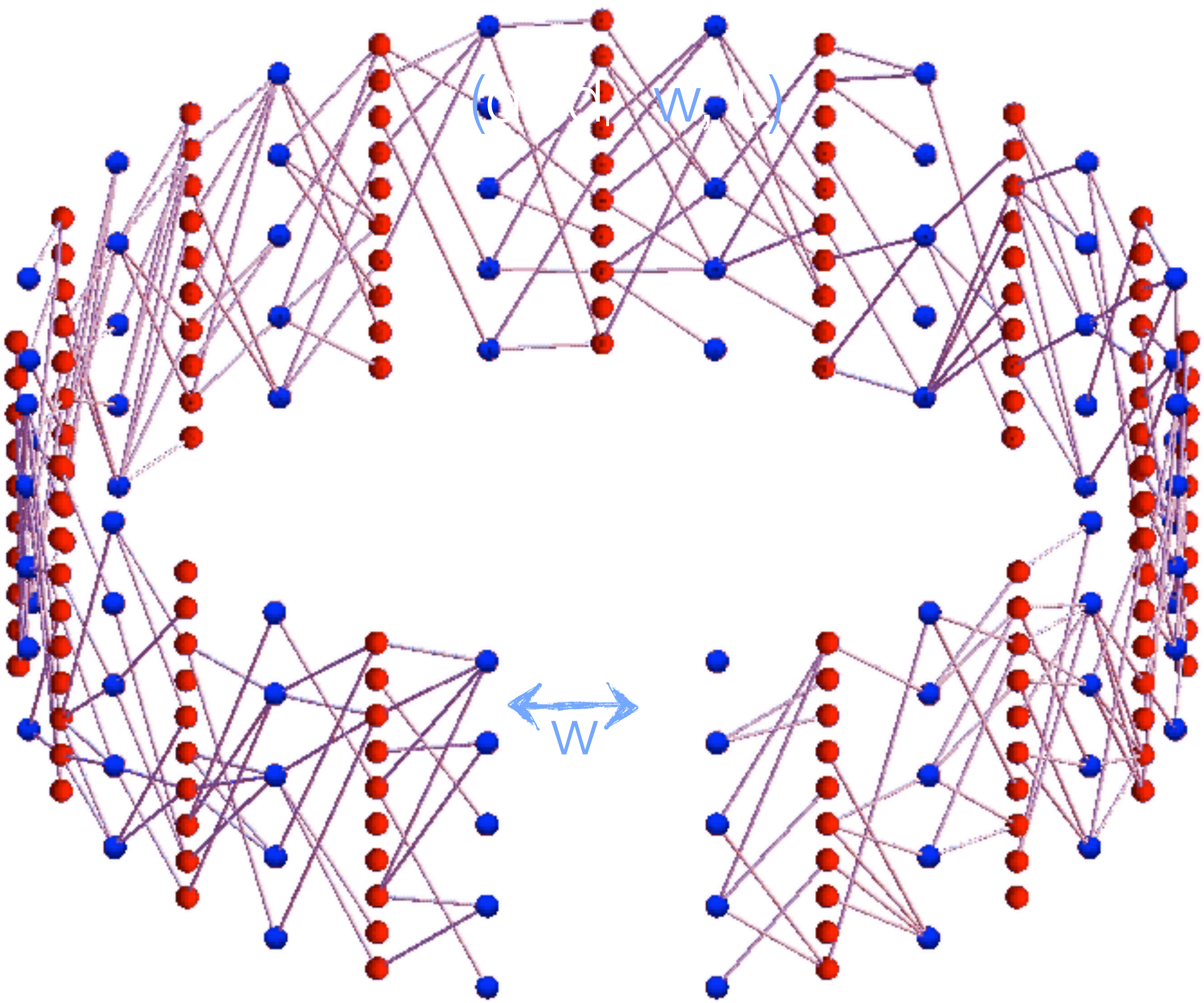


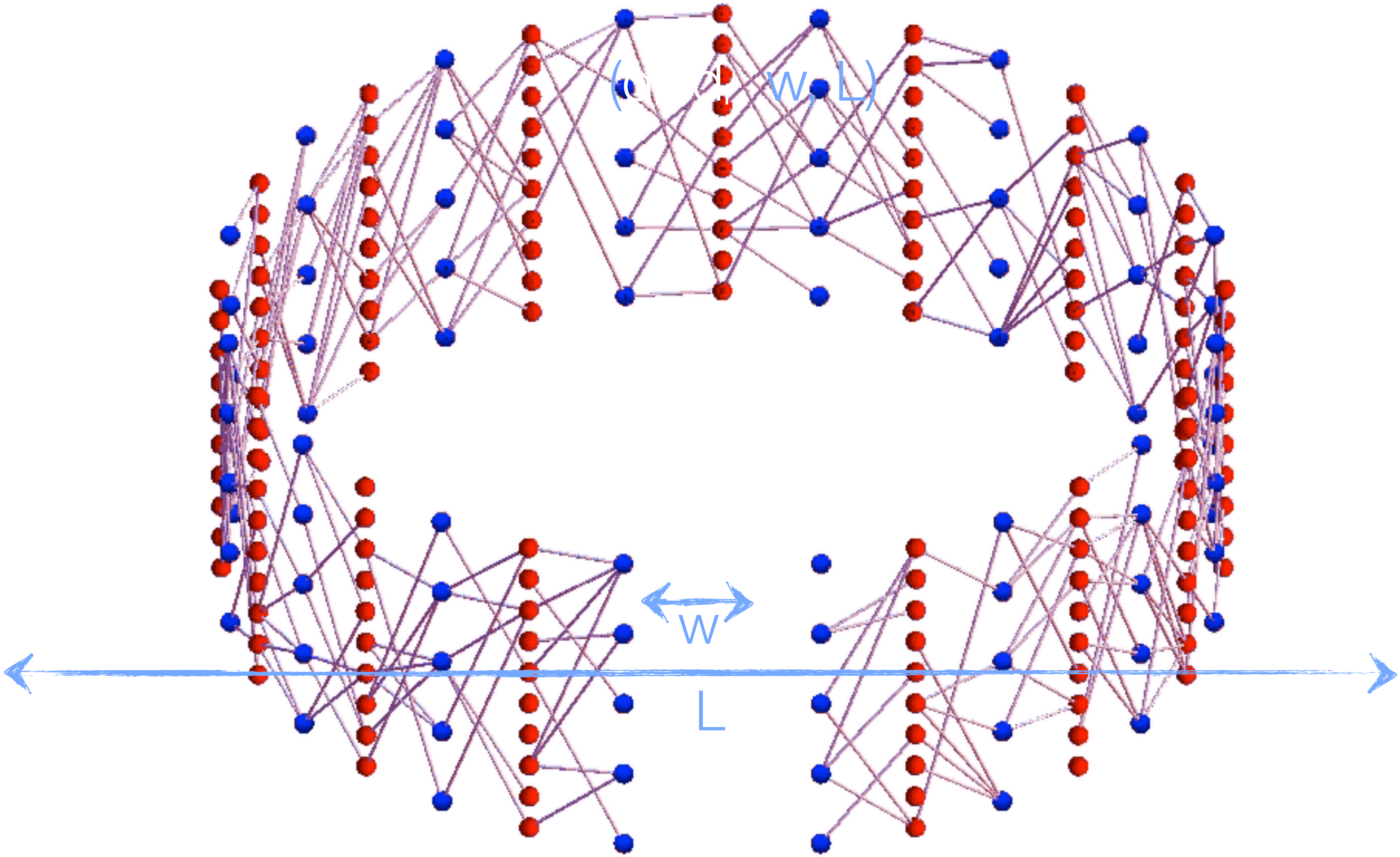


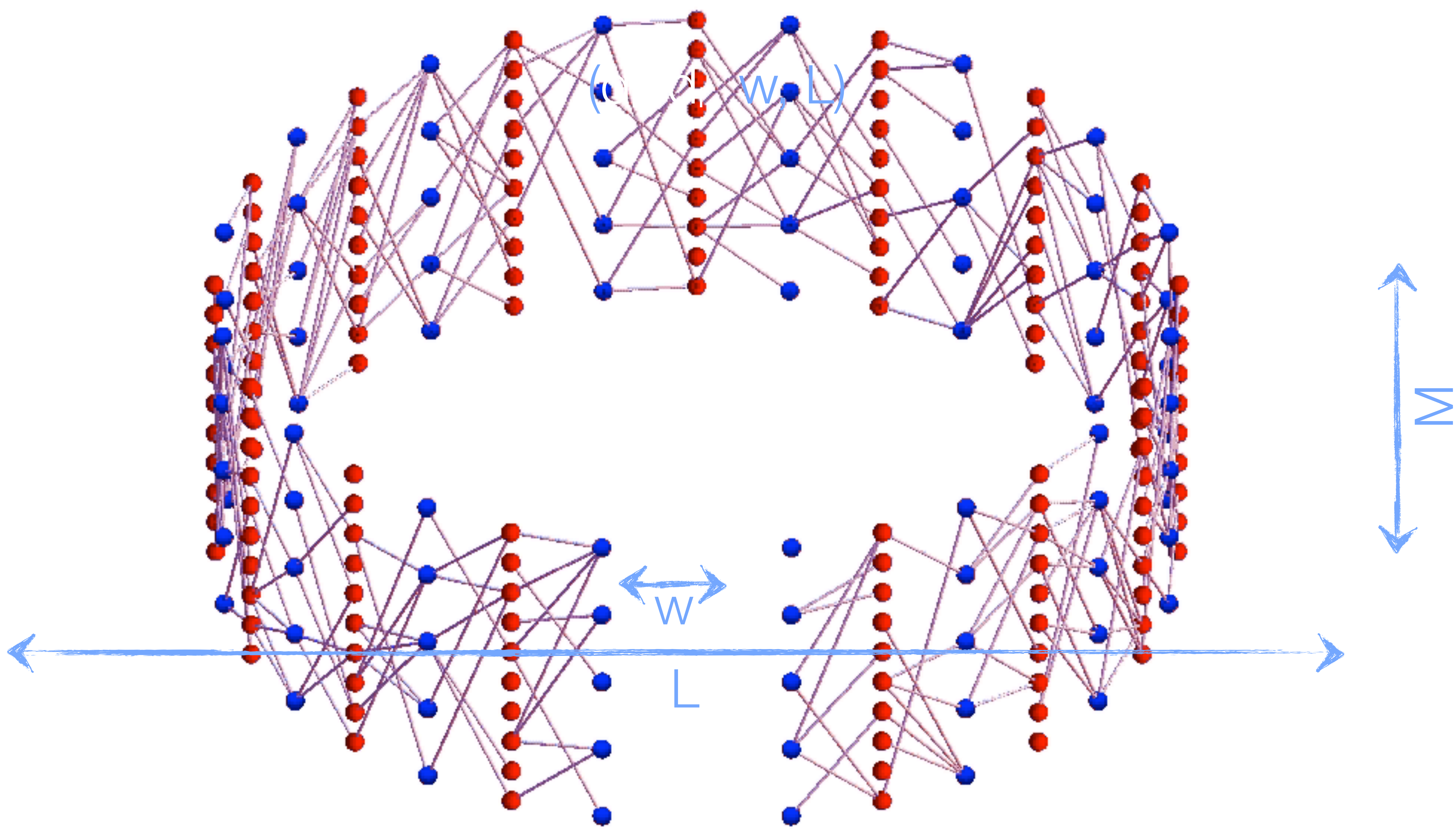




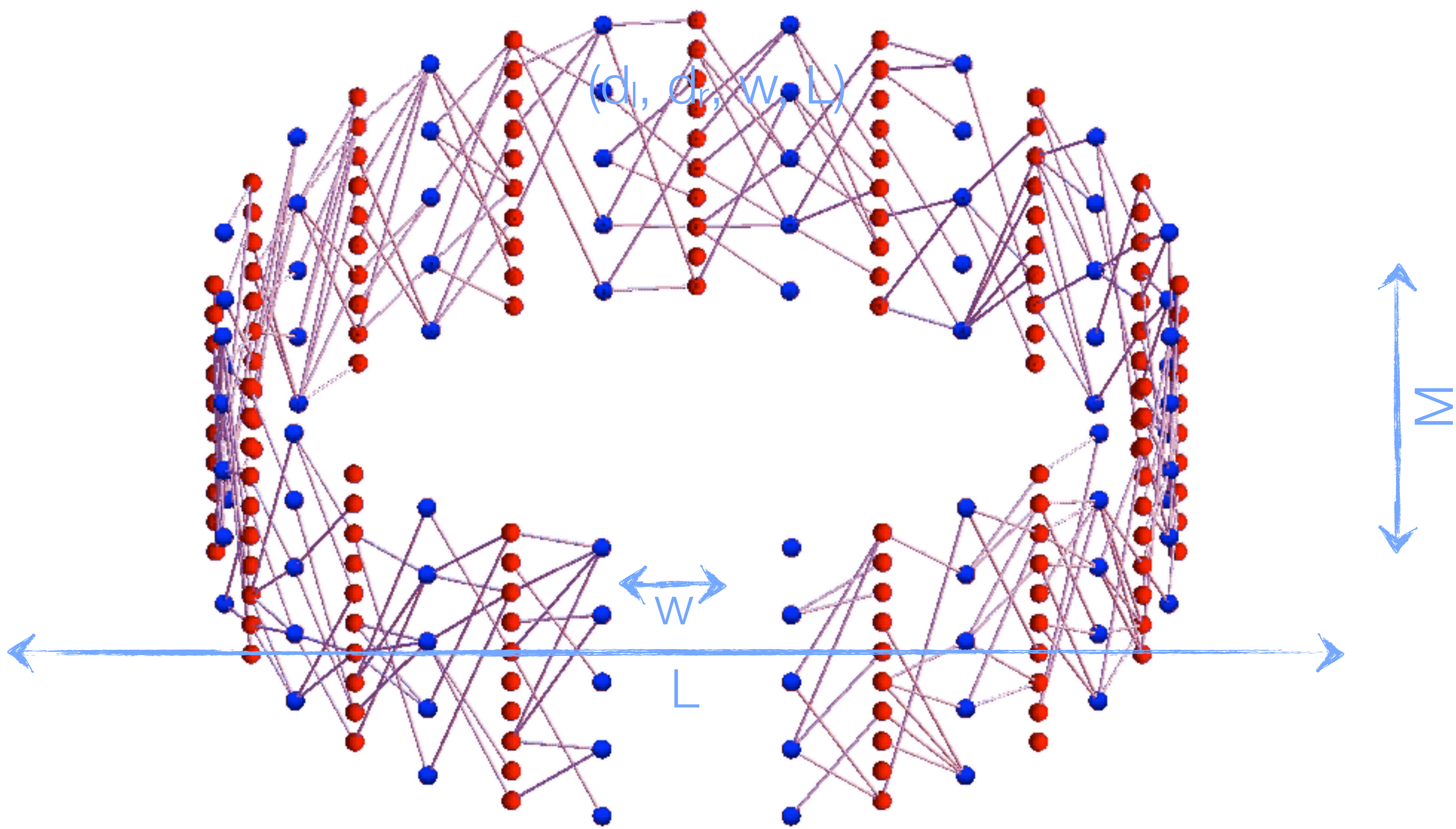



w


(w, L)
w
L

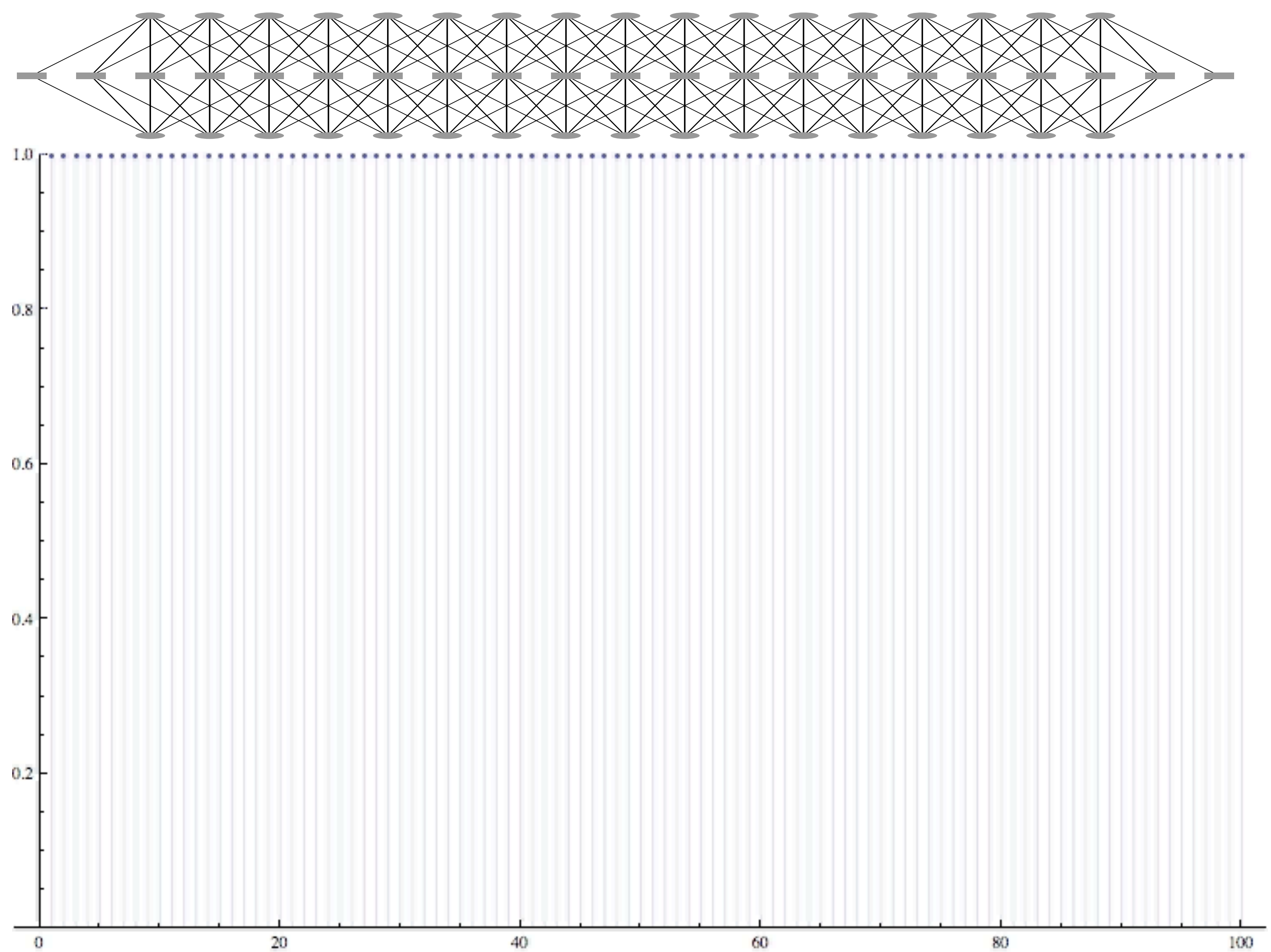


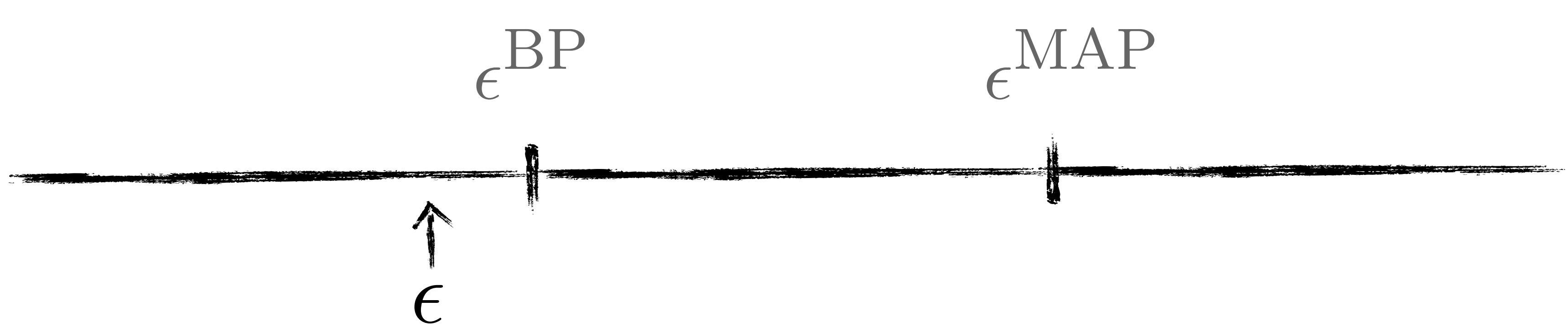
w, L)
w
L
M



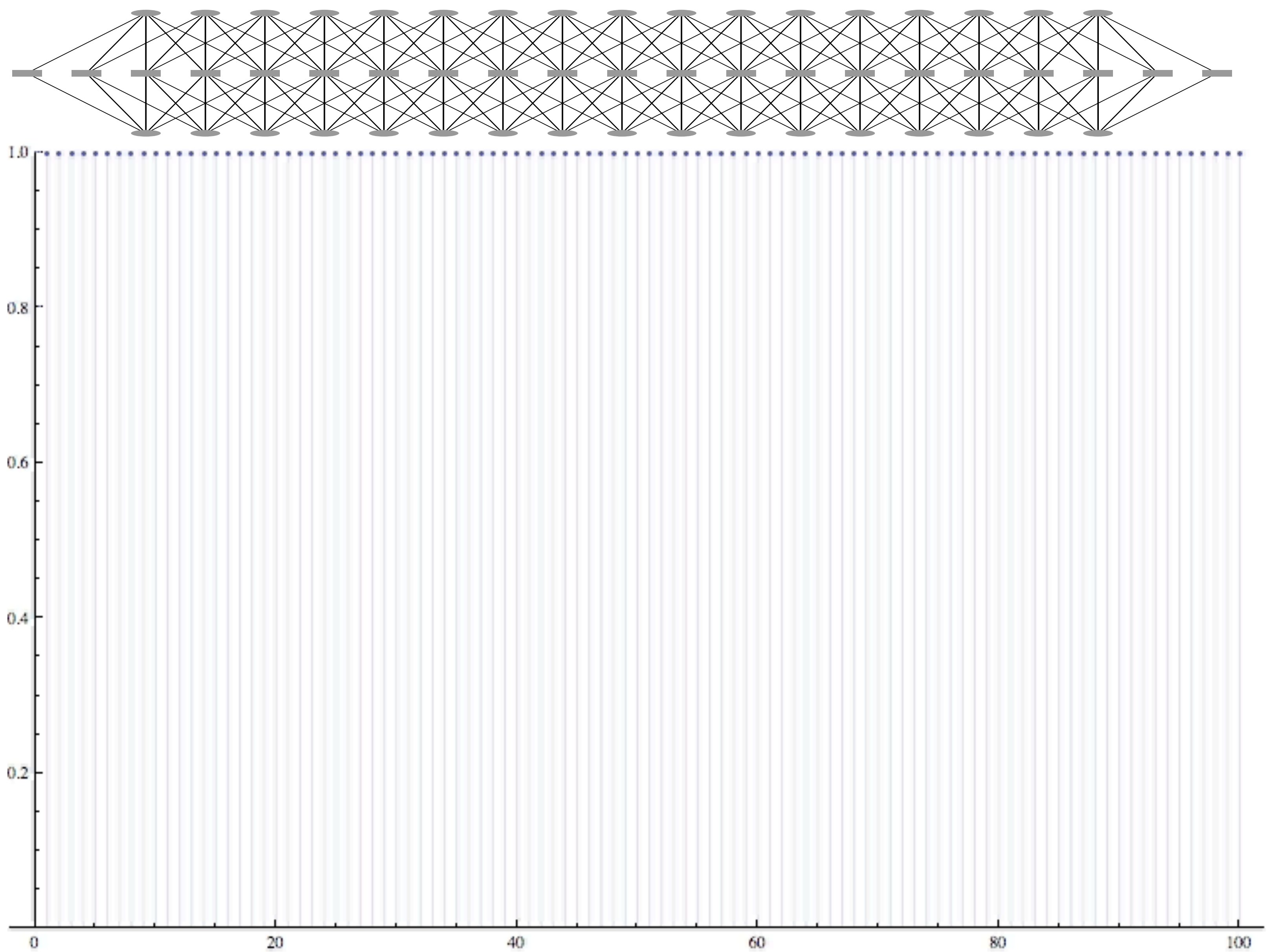
(dl, dr, w, L)
w
L
M

DE for Coupled Ensemble

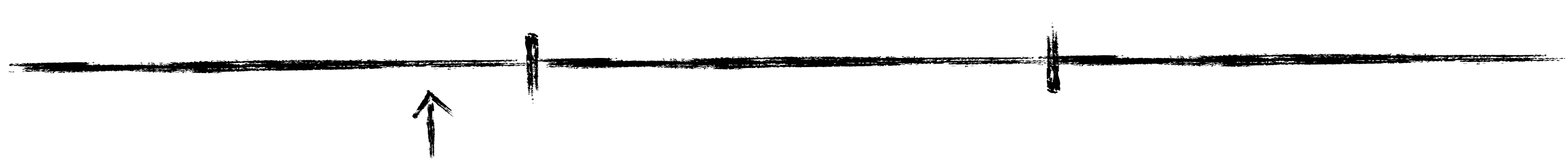




DE for Coupled Ensemble

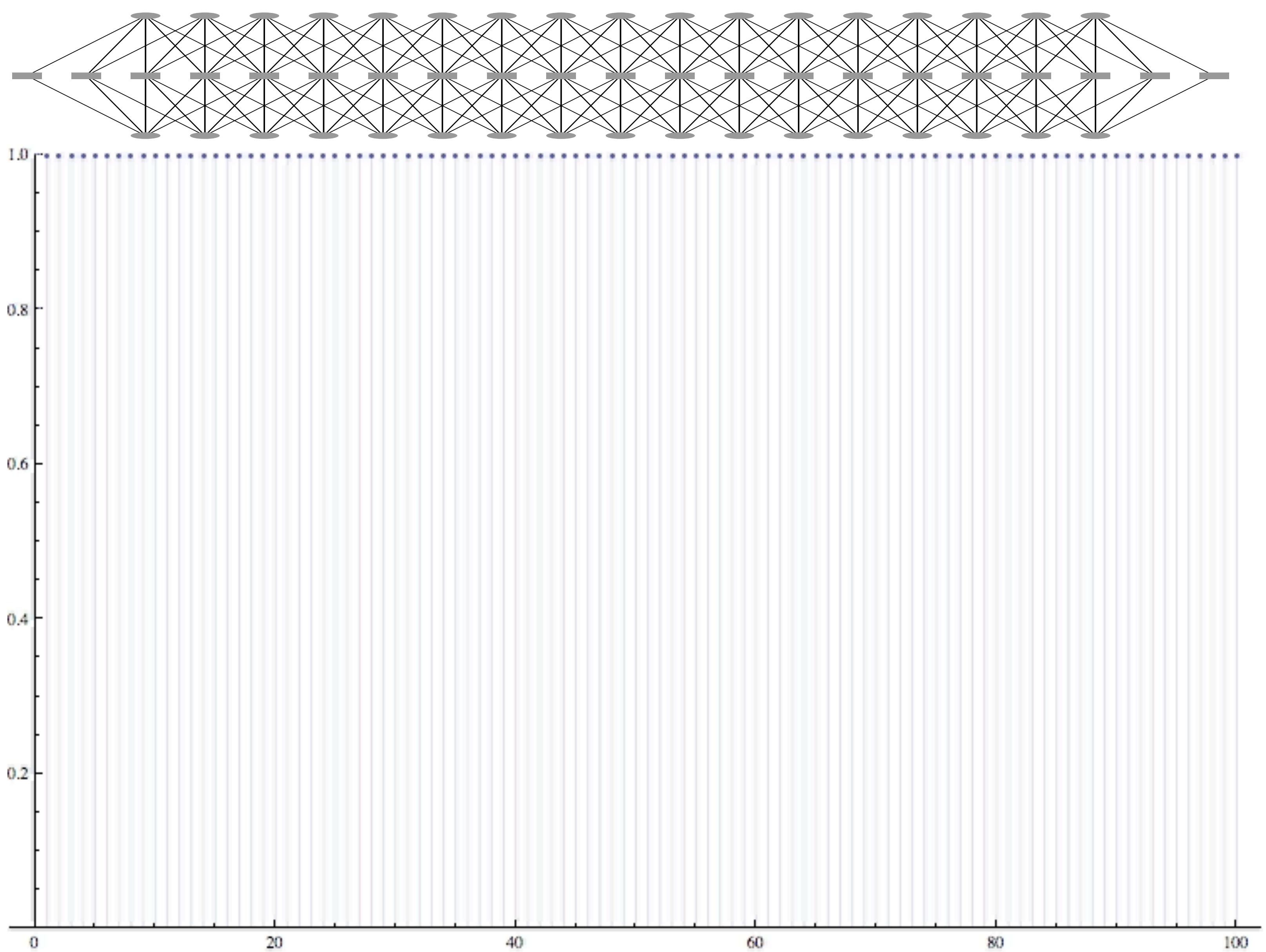


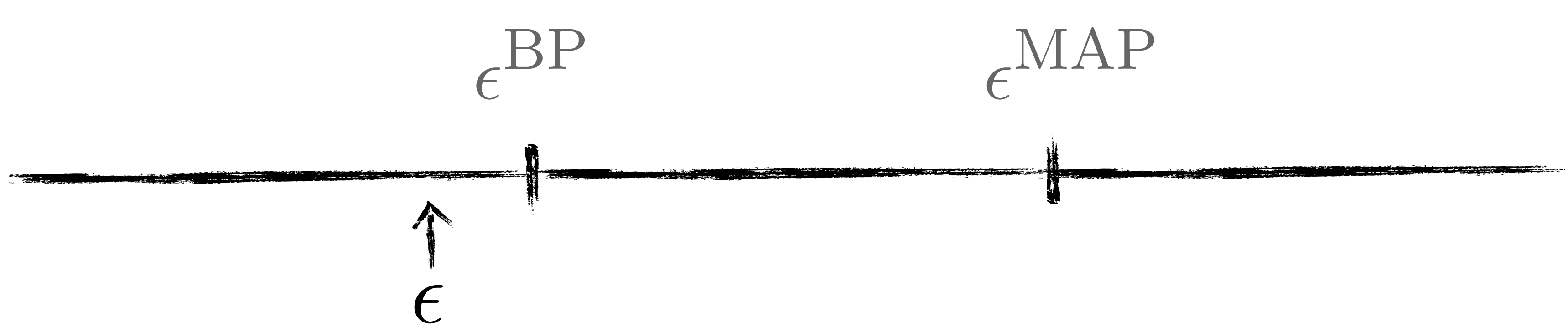
ϵ^{BP} ϵ^{MAP}



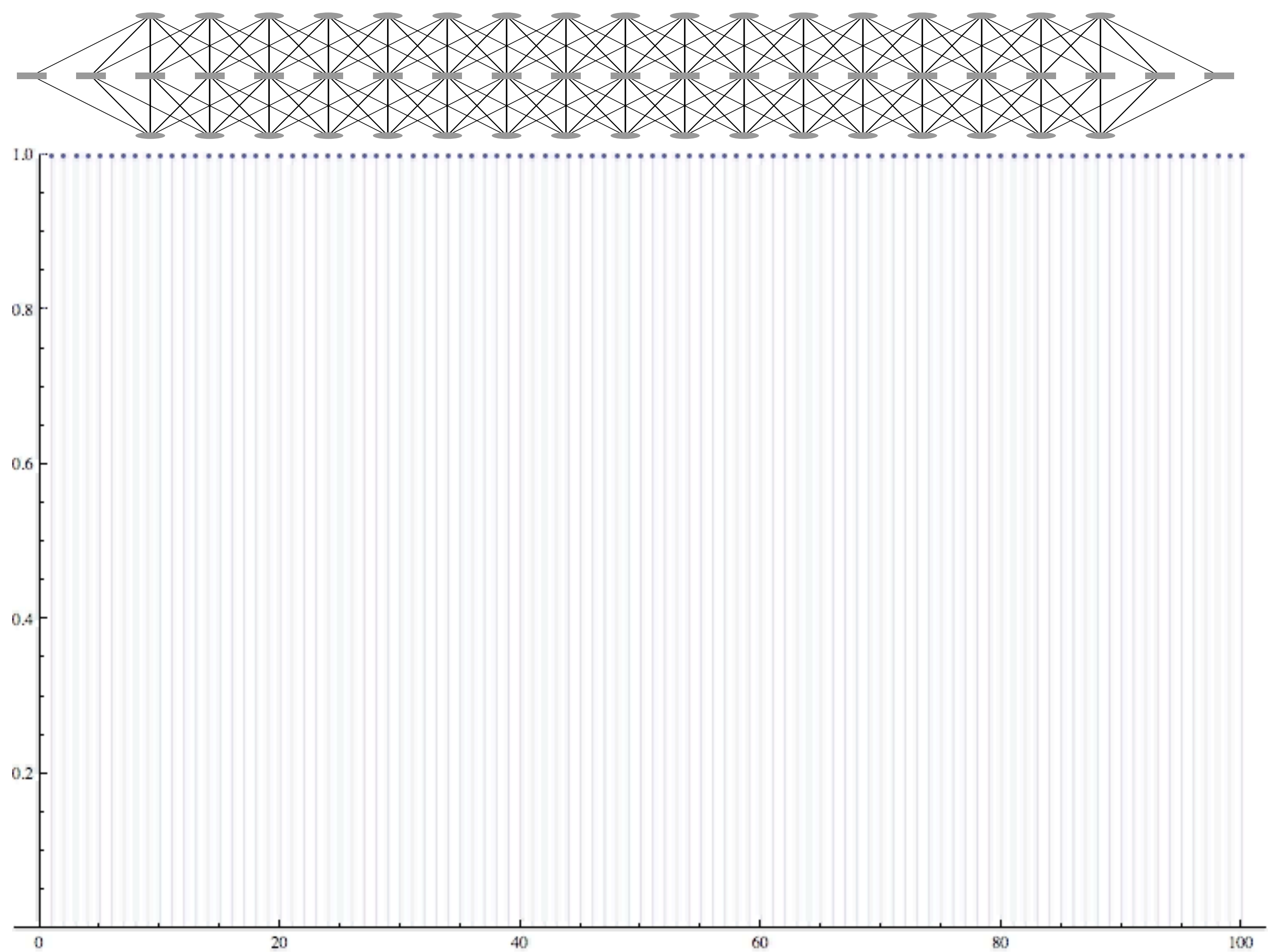
ϵ

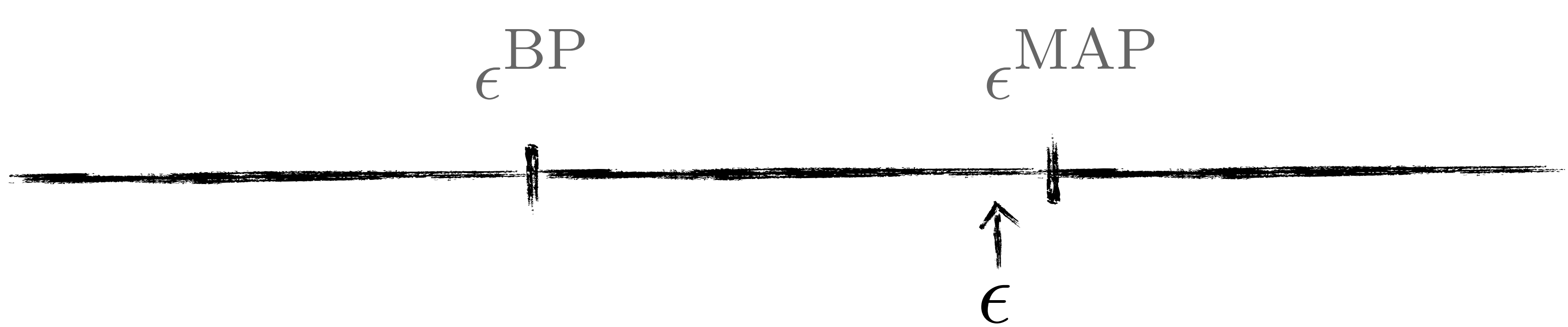
DE for Coupled Ensemble



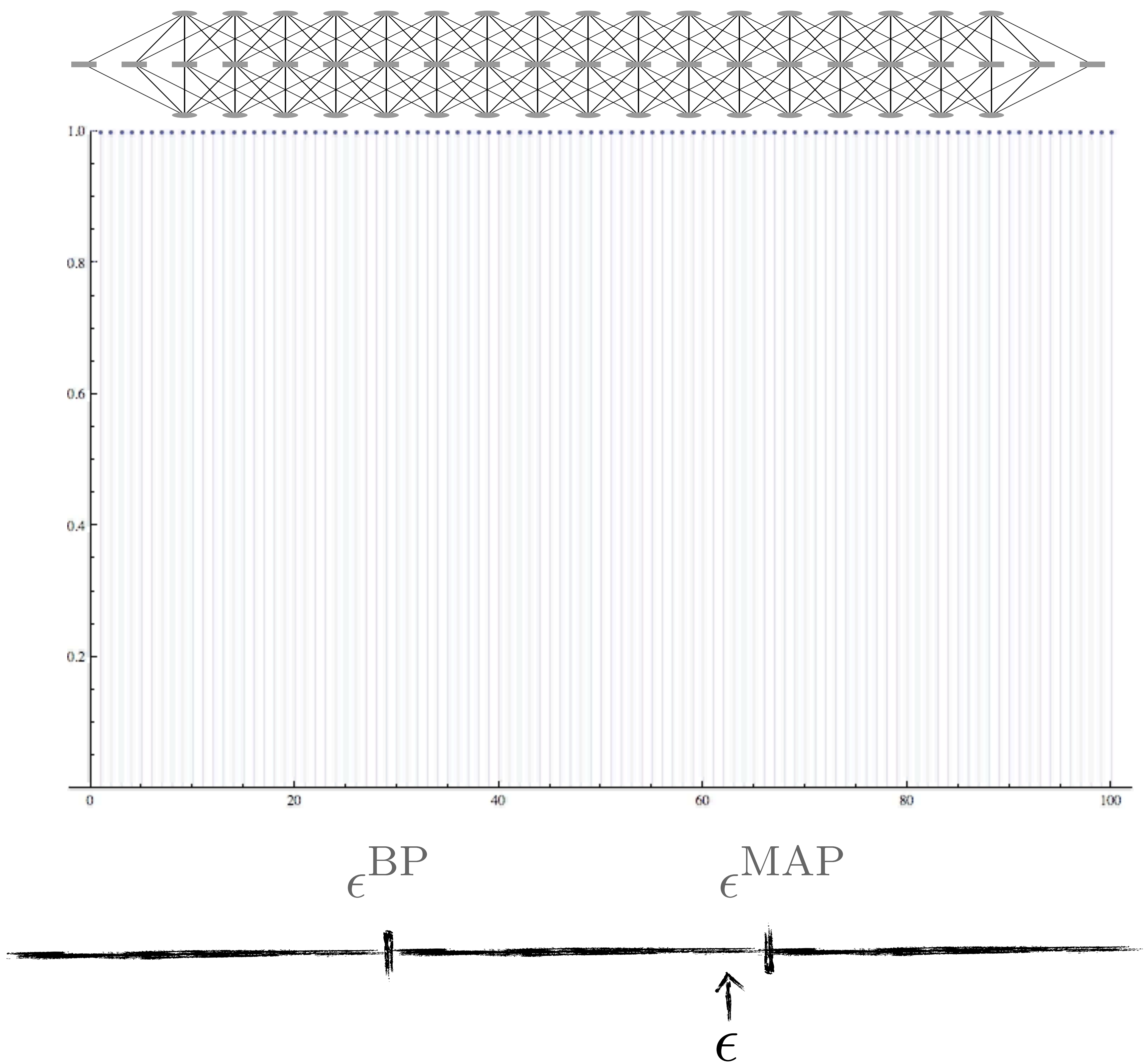


DE for Coupled Ensemble

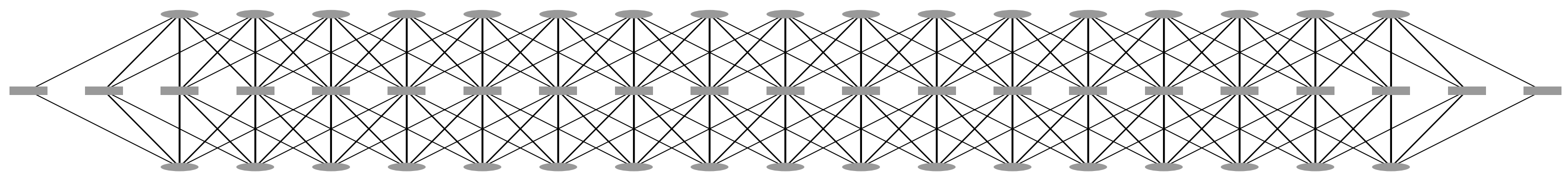




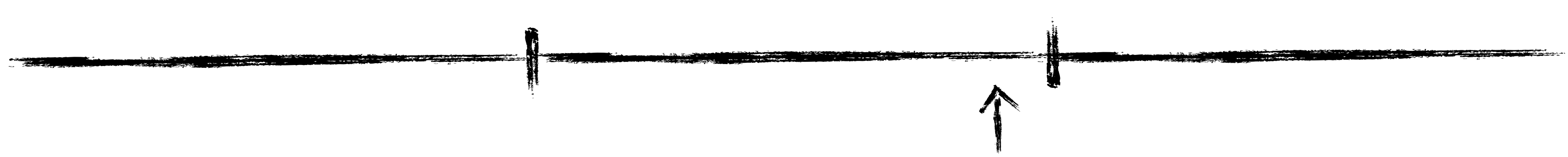
DE for Coupled Ensemble



DE for Coupled Ensemble

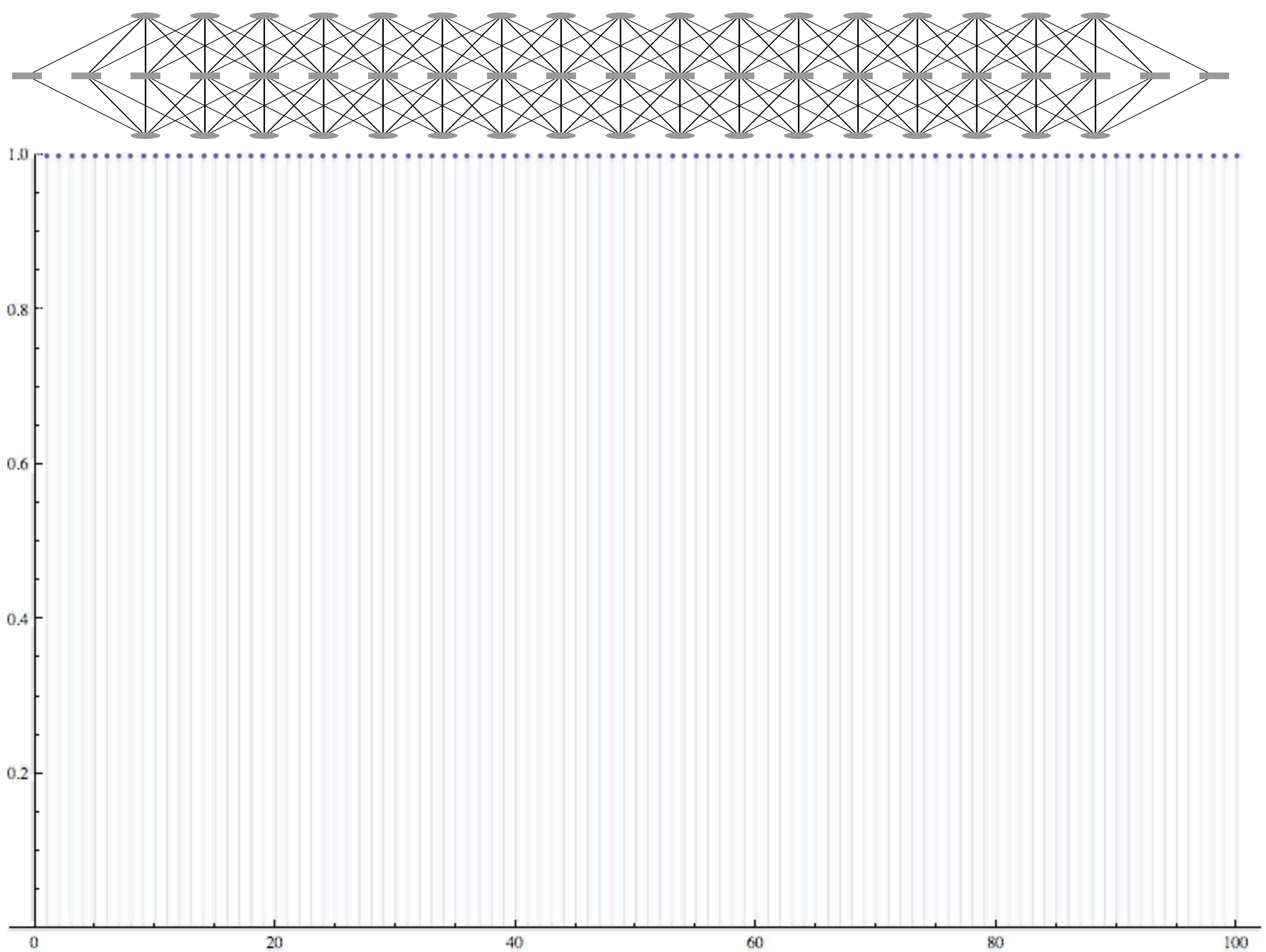


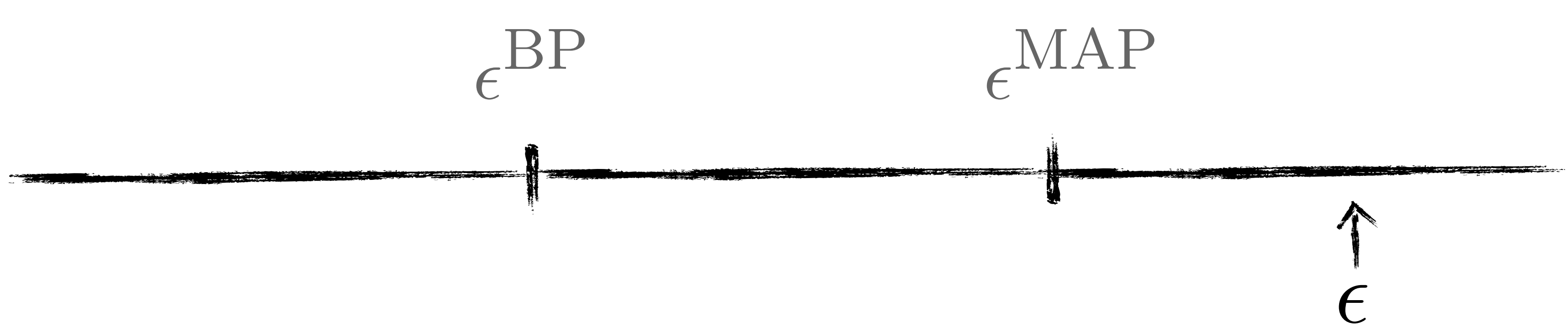
ϵ^{BP} ϵ^{MAP}



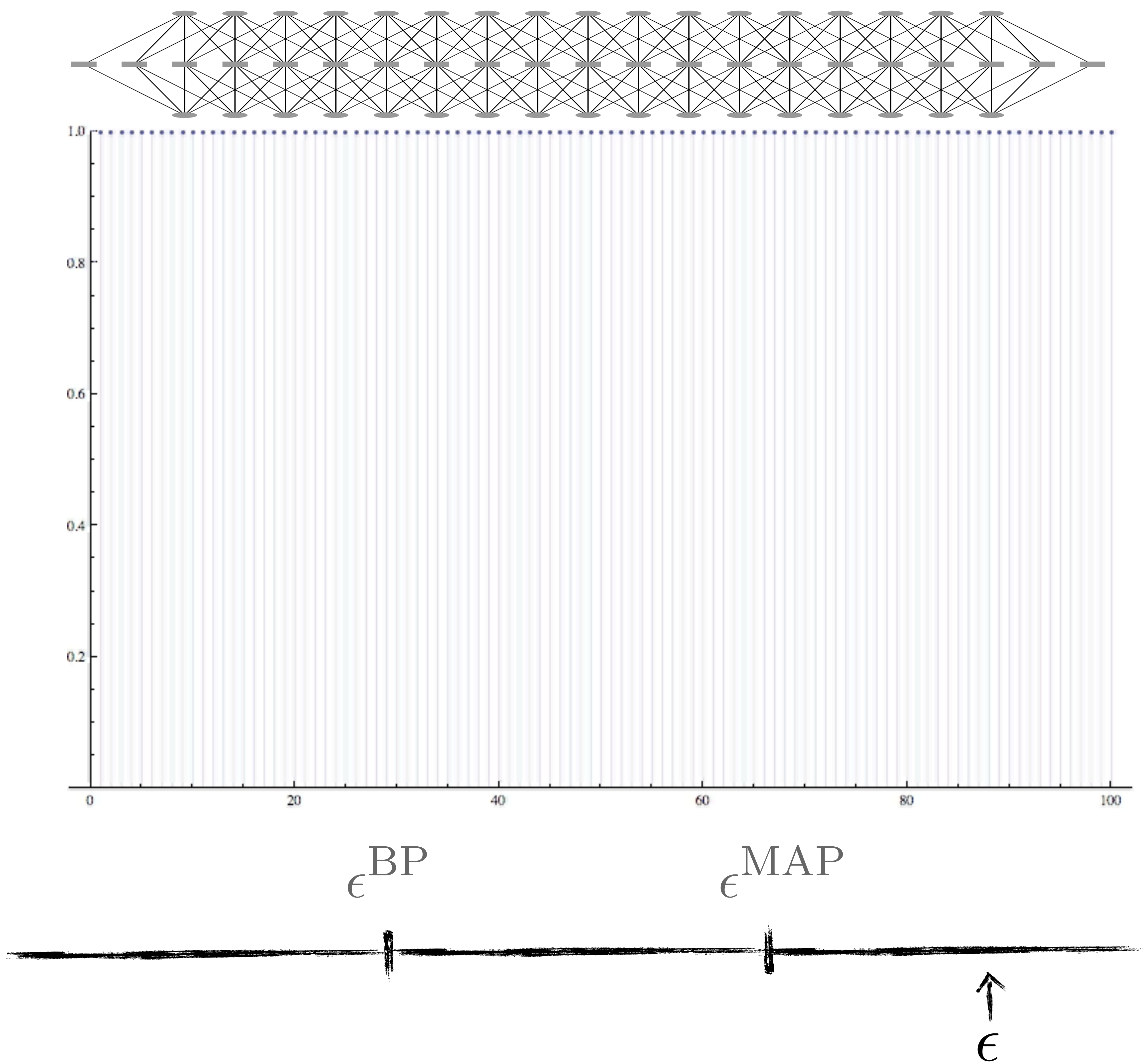
ϵ

DE for Coupled Ensemble





DE for Coupled Ensemble



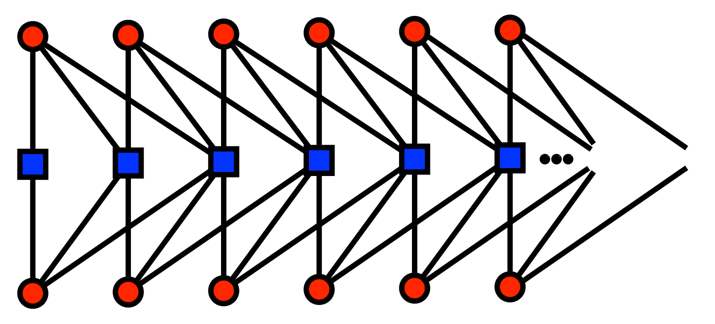
Spatial Coupling — Key Ideas

a little help at boundary gets things started (nucleation)

proper structure ensures that process continues (crystallisation)

universal phenomenon

Spatial Coupling — Key Ideas

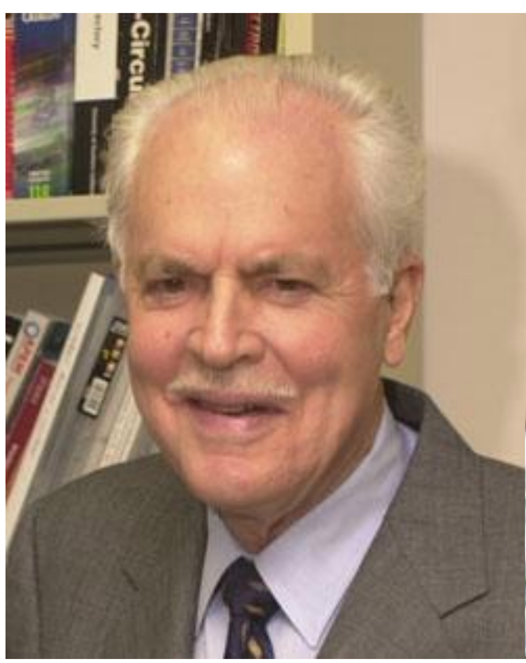


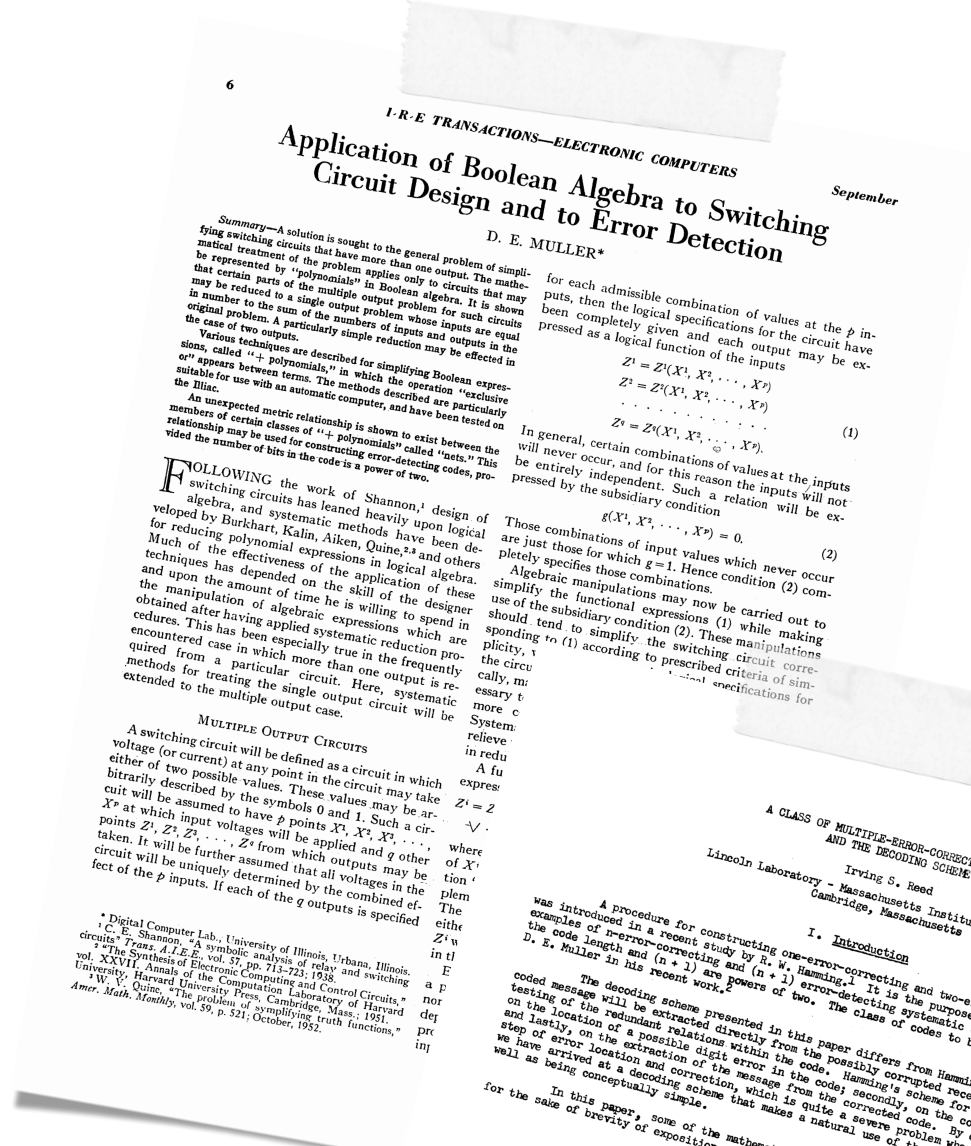
a little help at boundary gets things started (nucleation)

proper structure ensures that process continues (crystallisation)

universal phenomenon

Reed-Muller Codes





6 I·R·E TRANSACTIONS—ELECTRONIC COMPUTERS September

Application of Boolean Algebra to Switching Circuit Design and to Error Detection

D. E. MULLER*

Summary—A solution is sought to the general problem of simplifying switching circuits that have more than one output. The mathematical treatment of the problem applies only to circuits that may be represented by "polynomials" in Boolean algebra. It is shown that certain parts of the multiple output problem for such circuits may be reduced to a single output problem whose inputs are equal in number to the sum of the numbers of inputs and outputs in the original problem. A particularly simple reduction may be effected in the case of two outputs.

Various techniques are described for simplifying Boolean expressions, called "+ polynomials," in which the operation "exclusive or" appears between terms. The methods described are particularly suitable for use with an automatic computer, and have been tested on the Illiac.

An unexpected metric relationship is shown to exist between the members of certain classes of "+ polynomials" called "nets." This relationship may be used for constructing error-detecting codes, provided the number of bits in the code is a power of two.

FOLLOWING the work of Shannon,[1] design of switching circuits has leaned heavily upon logical algebra, and systematic methods have been developed by Burkhart, Kalin, Aiken, Quine,[2,3] and others for reducing polynomial expressions in logical algebra. Much of the effectiveness of the application of these techniques has depended on the skill of the designer and upon the amount of time he is willing to spend in the manipulation of algebraic expressions which are obtained after having applied systematic reduction procedures. This has been especially true in the frequently encountered case in which more than one output is required from a particular circuit. Here, systematic methods for treating the single output circuit will be extended to the multiple output case.

MULTIPLE OUTPUT CIRCUITS

A switching circuit will be defined as a circuit in which voltage (or current) at any point in the circuit may take either of two possible values. These values may be arbitrarily described by the symbols 0 and 1. Such a circuit will be assumed to have p points $X^1, X^2, X^3, \cdots, X^p$ at which input voltages will be applied and q other points $Z^1, Z^2, Z^3, \cdots, Z^q$ from which outputs may be taken. It will be further assumed that all voltages in the circuit will be uniquely determined by the combined effect of the p inputs. If each of the q outputs is specified for each admissible combination of values at the p inputs, then the logical specifications for the circuit have been completely given and each output may be expressed as a logical function of the inputs

$$Z^1 = Z^1(X^1, X^2, \cdots, X^p)$$
$$Z^2 = Z^2(X^1, X^2, \cdots, X^p)$$
$$\cdots$$
$$Z^q = Z^q(X^1, X^2, \cdots, X^p). \qquad (1)$$

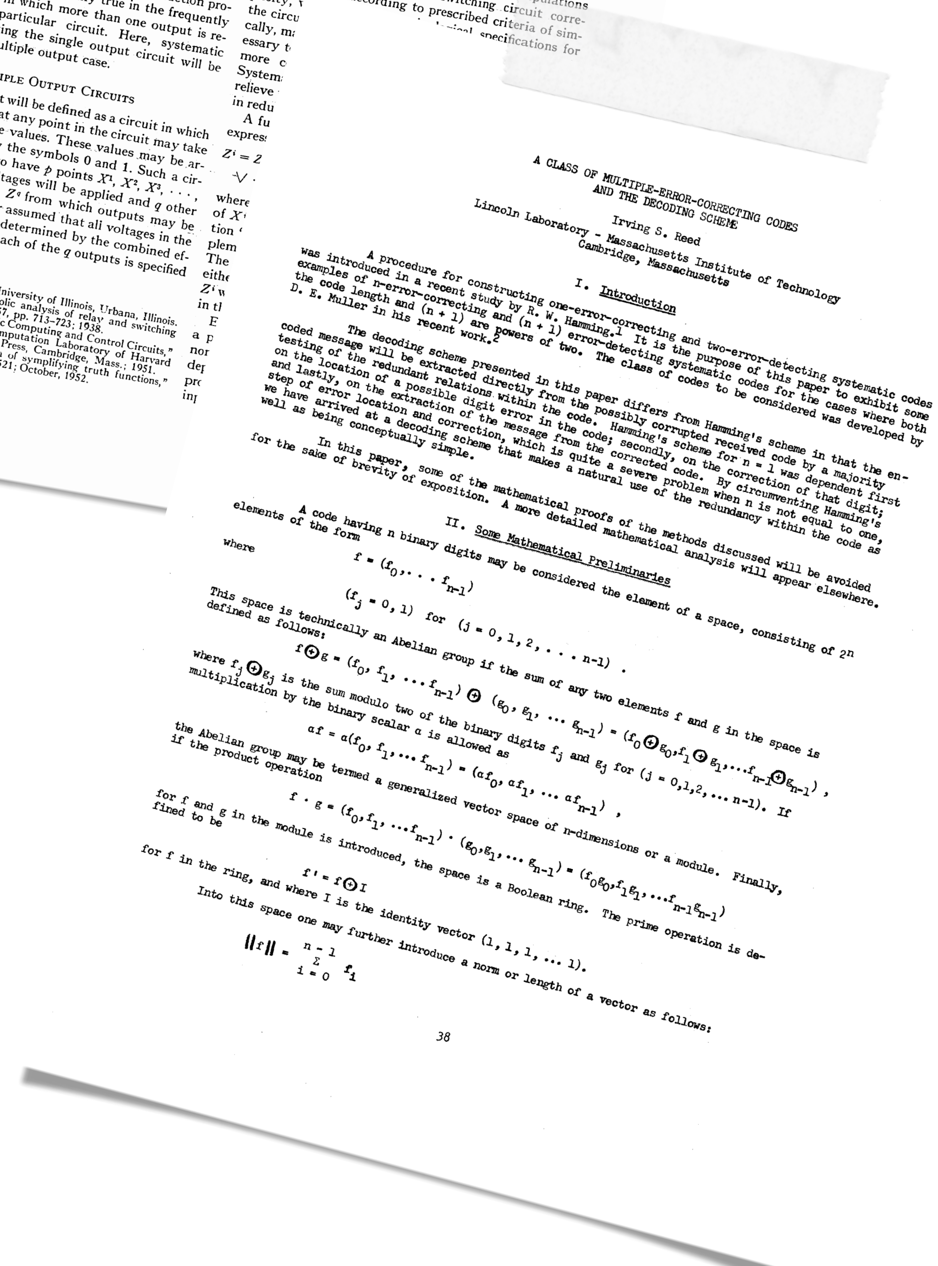
In general, certain combinations of values at the inputs will never occur, and for this reason the inputs will not be entirely independent. Such a relation will be expressed by the subsidiary condition

$$g(X^1, X^2, \cdots, X^p) = 0. \qquad (2)$$

Those combinations of input values which never occur are just those for which $g=1$. Hence condition (2) completely specifies those combinations.

Algebraic manipulations may now be carried out to simplify the functional expressions (1) while making use of the subsidiary condition (2). These manipulations should tend to simplify the switching circuit corresponding to (1) according to prescribed criteria of sim- plicity, ...

* Digital Computer Lab., University of Illinois, Urbana, Illinois.
[1] C. E. Shannon, "A symbolic analysis of relay and switching circuits," Trans. A.I.E.E., vol. 57, pp. 713–723; 1938.
[2] "The Synthesis of Electronic Computing and Control Circuits," vol. XXVII, Annals of the Computation Laboratory of Harvard University, Harvard University Press, Cambridge, Mass.; 1951.
[3] W. V. Quine, "The problem of simplifying truth functions," Amer. Math. Monthly, vol. 59, p. 521; October, 1952.



A CLASS OF MULTIPLE-ERROR-CORRECTING CODES AND THE DECODING SCHEME

Irving S. Reed

Lincoln Laboratory - Massachusetts Institute of Technology Cambridge, Massachusetts

I. Introduction

A procedure for constructing one-error-correcting and two-error-detecting systematic codes was introduced in a recent study by R. W. Hamming.[1] It is the purpose of this paper to exhibit some examples of n-error-correcting and (n + 1) error-detecting systematic codes for the cases where both the code length and (n + 1) are powers of two. The class of codes to be considered was developed by D. E. Muller in his recent work.[2]

The decoding scheme presented in this paper differs from Hamming's scheme in that the encoded message will be extracted directly from the possibly corrupted received code by a majority testing of the redundant relations within the code. Hamming's scheme for n = 1 was dependent first on the location of a possible digit error in the code; secondly, on the correction of that digit; and lastly, on the extraction of the message from the corrected code. By circumventing Hamming's step of error location and correction, which is quite a severe problem when n is not equal to one, we have arrived at a decoding scheme that makes a natural use of the redundancy within the code as well as being conceptually simple.

In this paper, some of the mathematical proofs of the methods discussed will be avoided for the sake of brevity of exposition. A more detailed mathematical analysis will appear elsewhere.

II. Some Mathematical Preliminaries

A code having n binary digits may be considered the element of a space, consisting of 2^n elements of the form

$$f = (f_0, \ldots f_{n-1})$$

where

$$(f_j = 0, 1) \text{ for } (j = 0, 1, 2, \ldots n-1).$$

This space is technically an Abelian group if the sum of any two elements f and g in the space is defined as follows:

$$f \odot g = (f_0, f_1, \ldots f_{n-1}) \odot (g_0, g_1, \ldots g_{n-1}) = (f_0 \odot g_0, f_1 \odot g_1, \ldots f_{n-1} \odot g_{n-1}),$$

where $f_j \odot g_j$ is the sum modulo two of the binary digits f_j and g_j for $(j = 0,1,2,\ldots n-1)$. If multiplication by the binary scalar α is allowed as

$$\alpha f = \alpha(f_0, f_1, \ldots f_{n-1}) = (\alpha f_0, \alpha f_1, \ldots \alpha f_{n-1}),$$

the Abelian group may be termed a generalized vector space of n-dimensions or a module. Finally, if the product operation

$$f \cdot g = (f_0, f_1, \ldots f_{n-1}) \cdot (g_0, g_1, \ldots g_{n-1}) = (f_0 g_0, f_1 g_1, \ldots f_{n-1} g_{n-1})$$

for f and g in the module is introduced, the space is a Boolean ring. The prime operation is defined to be

$$f' = f \odot I$$

for f in the ring, and where I is the identity vector (1, 1, 1, ... 1).

Into this space one may further introduce a norm or length of a vector as follows:

$$\|f\| = \sum_{i=0}^{n-1} f_i$$

38

Eleventh International Workshop on Algebraic and Combinatorial Coding Theory
June 16-22, 2008, Pamporovo, Bulgaria pp. 82-85

On complexity of decoding Reed-Muller codes within their code distance

Ilya Dumer
University of California, Riverside, CA, USA
dumer@ee.ucr.edu
Grigory Kabatiansky
Institute for Information Transmission Problems, Moscow, RUSSIA
kaba@iitp.ru
Cédric Tavernier
Communications and Systems Le Plessis Robinson, FRANCE
cedric.tavernier@c-s.fr

Abstract. Recently Gopalan, Klivans, and Zuckerman proved that any b
Reed-Muller (RM) code RM(s,m) can be list-decoded up to its minimum di
d with a polynomial complexity of order n^3 in blocklength n. The GKZ alg
employs a new upper bound that is substantially tighter for RM codes of fixe
s than the universal Johnson bound, and yields a constant number of co
in a sphere of radius less than d. In this note, we modify the GKZ algori
show that full list decoding up to the code distance d can be performed wit
complexity order of at most $n \ln^{s-1} n$. We also show that our former algorit
the same complexity order $n \ln^{s-1} n$ if combined with the new GKZ bo
list size.

1 Introduction

Binary Reed-Muller (RM) codes RM(s,m) of order s have le
dimension $k = k(s,m)$, and distance $d = d(s,m)$ as follows

$$n = 2^m, \quad k = \sum_{i=0}^{s} \binom{m}{i}, \quad d = 2^{m-s}.$$

The renowned majority decoding algorithm of [1] provide
decoding (BDD) for any code $RM(s,m)$ and corrects
less than $d/2$ with complexity order of kn. Even a low
of $n \min(s, m-s)$ is required for various recursive techni
Both recursive and majority algorithms correct many
the BDD radius $d/2$; however, they fall short of compl
within any given decoding radius $T \geq d/2$. Therefore
decoding [5] algorithms that output the list

$$L_T(\mathbf{y}) = \{\mathbf{c} \in \mathrm{RM}(s,m) : d(\mathbf{y},\mathbf{c})$$ located within

of *all* vectors $\mathbf{c}$ of a code RM(s,m) located within
received vector $\mathbf{y}$.

Soft Decision Majority Decoding of Reed-Muller Codes

Ilya Dumer and Rafail Krichevskiy*

Abstract

We present a new soft decision majority decoding algorithm for Reed-Muller codes $RM(r,m)$.
First, the reliabilities of 2^m transmitted symbols are recalculated into the reliabilities of 2^{m-r}
parity checks that represent each information bit. In turn, information bits are obtained by
the *weighted majority* that gives more weight to more reliable parity checks. It is proven that
for long low-rat $RM(r,m)$, our soft decision algorithm outperforms its conventional hard
decisi $10\log_{10}(\pi/2) \approx 2$ dB at any given output error probability. For fixed
, our algorithm increases almost $2^{r/2}$ times the correcting capability of
tance decoding.
decoding, parity checks, reliabilities, Gaussian channel.

] have length $n = 2^m$, dimension $k = \sum_{i=0}^{r} \binom{m}{i}$, and code distance
ding has been introduced in the seminal paper [13] followed by
5], [7], [11], [12], [14], [17], and references therein). The algorithm
ng with complexity order of nk or less. It is also known [8] that
rror patterns beyond the weight $\lfloor (d-1)/2 \rfloor$. Namely, for fixed
tors of Hamming weight $n(1-\varepsilon_r)/2$, where the residual term ε_r
$m \to \infty$. For RM codes of fixed rate R, we correct most error

hemes were designed for RM codes in the past decade. The
rovide bounded distance decoding with low complexity order
nd soft-decision channels [6]. Simulation results [16] also
rithms can significantly surpass bounded distance decoding.
lightly higher complexity order of $n^2 m$ for codes $RM(2,m)$
higher weight $n(1-\varepsilon)/2$, where ε has order of $(m/n)^{1/4}$
ers multistage maximum-likelihood decoding by using an
ity is exponent in n.
coding for RM codes used over additive white Gaussian
l with white Gaussian noise $\mathcal{N}(0,\sigma^2)$ and the probability

$$g(u) = e^{-u^2/2\sigma^2}/\sqrt{2\pi}\sigma. \qquad (1)$$

*The authors are with the College of Engineering, University of California, Riverside, CA 92521. This research was supported by the NSF grant NCR-9703844.

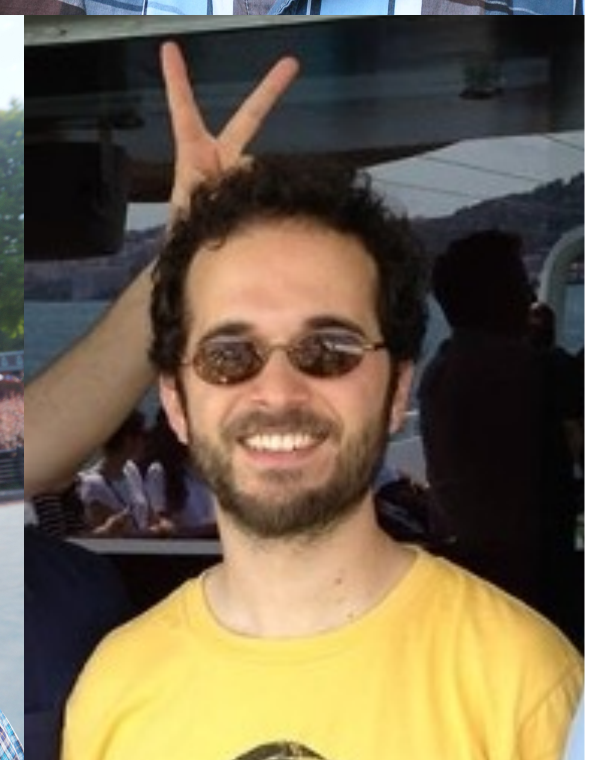
SIMONS INSTITUTE
for the Theory of Computing

H. D. Pfister

S. Kumar

S. Kudekar

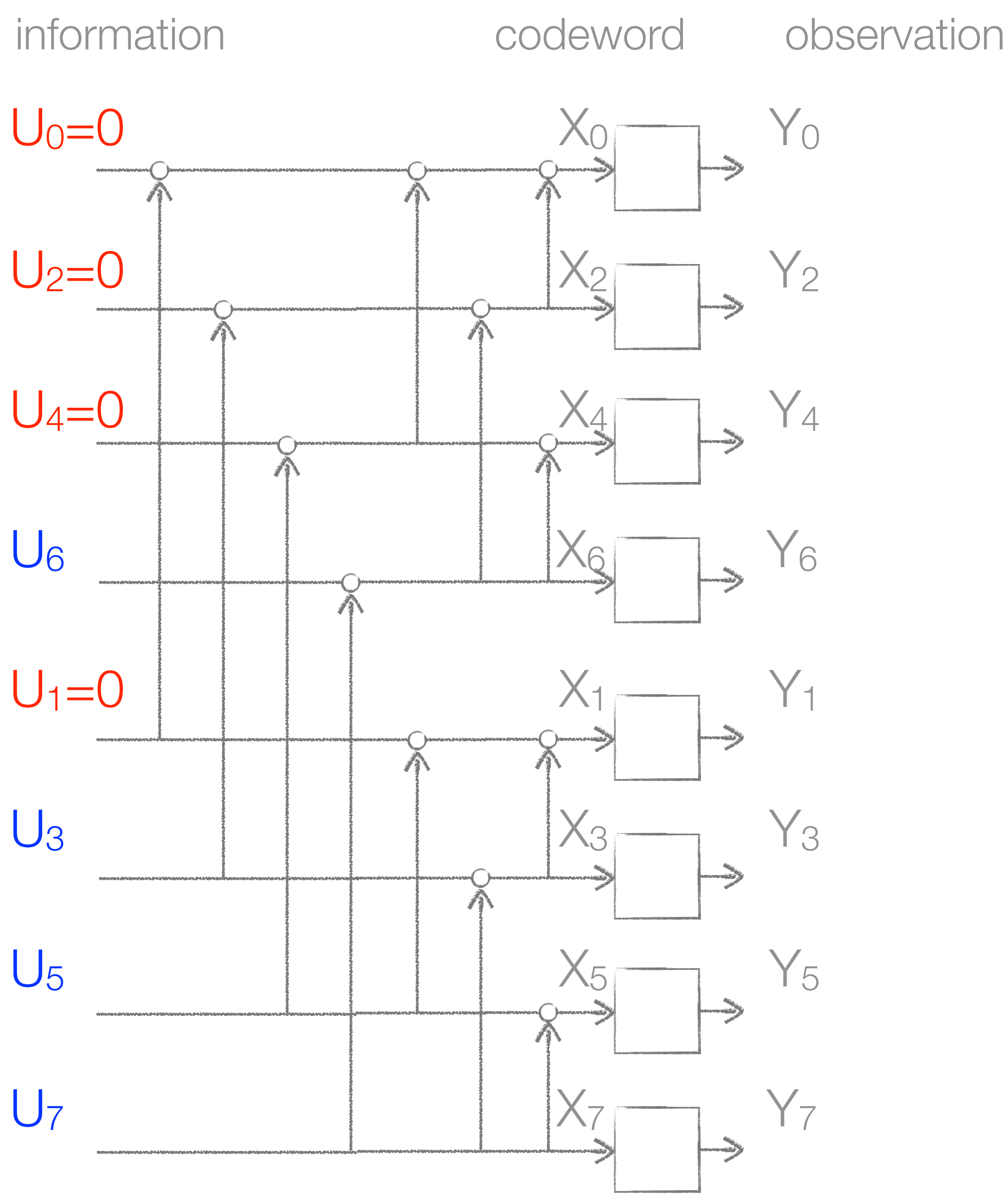
M. Mondelli



E. Sasoglu

RM versus Polar

	Polar	RM
bad	$Z \not\sim 0$	w small
good	$Z \sim 0$	w large



Do RM codes achieve capacity?

iv:1411.4590v1 [cs.IT] 17 Nov 2014

Reed-Muller codes for random erasures and errors

Emmanuel Abbe* Amir Shpilka† Avi Wigderson‡

Abstract

This paper studies the parameters for which Reed-Muller (RM) codes over $GF(2)$ can correct random erasures and random errors with high probability, and in particular when can they achieve capacity for these two classical channels. Necessarily, the paper also studies properties of evaluations of multi-variate $GF(2)$ polynomials on random sets of inputs.

For erasures, we prove that RM codes achieve capacity both for very high rate and very low rate regimes. For errors, we prove that RM codes achieve capacity for very low rate regimes, and for very high rates, we show that they can uniquely decode at about square root of the number of errors at capacity.

The proofs of these four results are based on different techniques, which we find interesting in their own right. In particular, we study the following questions about $E(m,r)$, the matrix whose rows are truth tables of all monomials of degree $\leq r$ in m variables. What is the most (resp. least) number of random columns in $E(m,r)$ that define a submatrix having full column rank (resp. full row rank) with high probability? We obtain tight bounds for very small (resp. very large) degrees r, which we use to show that RM codes achieve capacity for erasures in these regimes.

Our decoding from random errors follows from the following novel reduction. For every linear code C of sufficiently high rate we construct a new code C', also of very high rate, such that for every subset S of coordinates, if C can recover from erasures in S, then C' can recover from errors in S. Specializing this to RM codes and using our results for erasures imply our result on unique decoding of RM codes at high rate.

Finally, two of our capacity achieving results require tight bounds on the weight distribution of RM codes. We obtain such bounds extending the recent [KLP12] bounds from constant degree to linear degree polynomials.

in Applied and Computational Mathematics, and Department of Electrical Engineering, Princeton
inceton, USA, eabbe@princeton.edu
t of Computer Science,Tel-Aviv University, Tel-Aviv, Israel, shpilka@post.tau.ac.il. The research
e results has received funding from the European Community's Seventh Framework Programme
) under grant agreement number 257575, and from the Israel Science Foundation (grant number
dvanced Study, Princeton, USA, avi@ias.edu. This research was partially supported by NSF grant

BEC: Yes, for $R \to 0$ or 1

BSC: Yes, for $R \to 0$

BSC: "Not too bad", for $R \to 1$

Yes, $0 < R < 1$, for BEC.

Ingredients

- *RM codes are 2-transitive*
- *symmetric monotone sets have sharp thresholds*
- *EXIT functions satisfy the Area Theorem*



RM Codes are 2-Transitive

0	1	1	0	1	0	0	1
1	1	0	0	0	0	1	1
1	0	1	0	0	1	0	1
0	0	0	0	1	1	1	1
1	0	0	1	1	0	0	1
0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1
1	1	1	1	1	1	1	1
1	0	0	1	0	1	1	0
0	0	1	1	1	1	0	0
0	1	0	1	1	0	1	0
1	1	1	1	0	0	0	0
0	1	1	0	0	1	1	0
1	1	0	0	1	1	0	0
1	0	1	0	1	0	1	0
0	0	0	0	0	0	0	0

IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. IT-14, NO. 2, MARCH 1968 189

New Generalizations of the Reed-Muller Codes Part I: Primitive Codes

TADAO KASAMI, MEMBER, IEEE, SHU LIN, MEMBER, IEEE, AND W. WESLEY PETERSON, FELLOW, IEEE

***Abstract*—First it is shown that all binary Reed–Muller codes with one digit dropped can be made cyclic by rearranging the digits. Then a natural generalization to the nonbinary case is presented, which also includes the Reed–Muller codes and Reed–Solomon codes as special cases. The generator polynomial is characterized and the minimum weight is established. Finally, some results on weight distribution are given.**

I. Introduction

It is well known that the first-order Reed–Muller codes with one digit dropped can be made cyclic by rearranging the digits.[1] In fact, for the cyclic form, the all 1's vector and the maximal length sequences of length $2^m - 1$ generate the code. We have observed that the entire class of Reed–Muller codes are cyclic. This has led to a new generalization which is better in many cases than previous ones. This newly found mathematical structure for this class of codes has made it possible to find some other new facts about Reed–Muller codes.

First we prove that the binary Reed–Muller codes are all cyclic, in order to show the very simple ideas involved. Then in the next section more general results are given. In the final section, results on the weight distribution of Reed–Muller codes are given.

Let v_I denote a vector of length $2^m - 1$ over $GF(2)$ consisting of all 1's, and let $v_1, v_2, \cdots, v_m$ denote m linearly independent "maximal length sequences" generated by the same linear shift register. It is well known that the code generated by these $m + 1$ vectors is cyclic and is equivalent to the first-order Reed–Muller code with one digit dropped.[1] Multiplication of vectors is defined as follows. If

$$u = (u_1, u_2, \cdots, u_n)$$

$$v = (v_1, v_2, \cdots, v_n),$$

then

$$uv = (u_1v_1, u_2v_2, \cdots, u_nv_n).$$

The νth order Reed–Muller code (with one digit dropped) is generated by $v_I, v_1, v_2, \cdots, v_m$ and products of any ν or fewer of these vectors. An equivalent statement is that a vector v is a code vector in a νth order Reed–Muller code (with one digit dropped) if and only if it can be expressed as a polynomial of degree ν or less in $v_I, v_1, v_2, \cdots, v_m$.

Let T denote the operation of shifting cyclically one place to the right. Then a code is a cyclic code if and only if for each code vector v, Tv is also a code vector.

The key idea in the proof is simply the observation that T commutes not only with the addition of vectors but also with multiplication. That is,

$$T(v_1 + v_2) = Tv_1 + Tv_2, \tag{1}$$

and also

$$T(v_1v_2) = (Tv_1)(Tv_2). \tag{2}$$

Consider any code vector in the νth order Reed–Muller code with one digit dropped. It can be expressed as a polynomial of degree ν or less in $v_I, v_1, v_2, \cdots, v_m$

$$v = \sum C_j v_I^{n_{Ij}} v_1^{n_{1j}} v_2^{n_{2j}} \cdots v_m^{n_{mj}}. \tag{3}$$

Then, because of the commutative property

$$Tv = \sum C_j v_I^{n_{Ij}} (Tv_1)^{n_{1j}} (Tv_2)^{n_{2j}} \cdots (Tv_m)^{n_{mj}}. \tag{4}$$

Since the first-order Reed–Muller code with one digit dropped is cyclic, $Tv_1, Tv_2, \cdots, Tv_m$ are code vectors and hence are linear combinations of $v_I, v_1, v_2, \cdots, v_m$. It follows that Tv is a polynomial of the same degree as ν in $v_I, v_1, v_2, \cdots, v_m$ and hence is a code vector, and the νth order Reed–Muller code with one digit dropped is cyclic.

For any cyclic code over $GF(q)$ of length n relatively prime to q, there is a set of three closely related codes—the original code, another cyclic code, and a code found by adding an overall parity check to one of the cyclic codes. Let us assume that the original code includes the all 1's word as a code vector. The extended code, of length $n + 1$, is found by adding an overall check digit. The other cyclic code is found by taking the subset of code vectors in the original code whose symbols add to zero, i.e., the code words which have 1 as a root. Given any one of the three codes, the others can be found easily. In the binary case, at least, given the weight distribution of one code, the weight distribution of the others can be found in a trivial way. The generalized Reed–Muller codes as defined here are cyclic codes in which the all 1's vector is a code vector, and their dual codes have 1 as a root of every code word. These codes have length $q^m - 1$.

Manuscript received March 6, 1967. This paper was presented at the 1967 International Symposium on Information Theory, San Remo, Italy. This work was supported in part by the Air Force Cambridge Research Laboratories, Office of Aerospace Research, Bedford, Mass., under Contract AF 19(628)-4379.
T. Kasami is with the Department of Control Engineering, Faculty of Engineering Science, Osaka University, Toyonaka, Japan.
S. Lin and W. W. Peterson are with the Department of Electrical Engineering, University of Hawaii, Honolulu, Hawaii.

Code is 1-Transitive

$$
\begin{matrix}
0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}
\qquad\qquad
\begin{matrix}
0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}
$$

$$\text{1-transitive} : \forall i, j \in [N], \exists \pi : [N] \to [N], \pi(i) = j \wedge \pi(\mathcal{C}) = \mathcal{C}$$

Code is 1-Transitive

0	1	1	0	1	0	0	1
1	1	0	0	0	0	1	1
1	0	1	0	0	1	0	1
0	0	0	0	1	1	1	1
1	0	0	1	1	0	0	1
0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1
1	1	1	1	1	1	1	1
1	0	0	1	0	1	1	0
0	0	1	1	1	1	0	0
0	1	0	1	1	0	1	0
1	1	1	1	0	0	0	0
0	1	1	0	0	1	1	0
1	1	0	0	1	1	0	0
1	0	1	0	1	0	1	0
0	0	0	0	0	0	0	0

0	1	1	0	1	0	0	1
1	1	0	0	0	0	1	1
1	0	1	0	0	1	0	1
0	0	0	0	1	1	1	1
1	0	0	1	1	0	0	1
0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1
1	1	1	1	1	1	1	1
1	0	0	1	0	1	1	0
0	0	1	1	1	1	0	0
0	1	0	1	1	0	1	0
1	1	1	1	0	0	0	0
0	1	1	0	0	1	1	0
1	1	0	0	1	1	0	0
1	0	1	0	1	0	1	0
0	0	0	0	0	0	0	0

$$\text{1-transitive} : \forall i, j \in [N], \exists \pi : [N] \to [N], \pi(i) = j \wedge \pi(\mathcal{C}) = \mathcal{C}$$

Code is 1-Transitive

0	1	1	0	1	0	0	1
1	1	0	0	0	0	1	1
1	0	1	0	0	1	0	1
0	0	0	0	1	1	1	1
1	0	0	1	1	0	0	1
0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1
1	1	1	1	1	1	1	1
1	0	0	1	0	1	1	0
0	0	1	1	1	1	0	0
0	1	0	1	1	0	1	0
1	1	1	1	0	0	0	0
0	1	1	0	0	1	1	0
1	1	0	0	1	1	0	0
1	0	1	0	1	0	1	0
0	0	0	0	0	0	0	0

0	1	1	0	1	0	0	1
1	1	0	0	0	0	1	1
1	0	1	0	0	1	0	1
0	0	0	0	1	1	1	1
1	0	0	1	1	0	0	1
0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1
1	1	1	1	1	1	1	1
1	0	0	1	0	1	1	0
0	0	1	1	1	1	0	0
0	1	0	1	1	0	1	0
1	1	1	1	0	0	0	0
0	1	1	0	0	1	1	0
1	1	0	0	1	1	0	0
1	0	1	0	1	0	1	0
0	0	0	0	0	0	0	0

$$\text{1-transitive} : \forall i, j \in [N], \exists \pi : [N] \to [N], \pi(i) = j \wedge \pi(\mathcal{C}) = \mathcal{C}$$

Code is 1-Transitive

0	1	1	0	1	0	0	1
1	1	0	0	0	0	1	1
1	0	1	0	0	1	0	1
0	0	0	0	1	1	1	1
1	0	0	1	1	0	0	1
0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1
1	1	1	1	1	1	1	1
1	0	0	1	0	1	1	0
0	0	1	1	1	1	0	0
0	1	0	1	1	0	1	0
1	1	1	1	0	0	0	0
0	1	1	0	0	1	1	0
1	1	0	0	1	1	0	0
1	0	1	0	1	0	1	0
0	0	0	0	0	0	0	0

0	1	1	0	1	0	0	1
1	1	0	0	0	0	1	1
1	0	1	0	0	1	0	1
0	0	0	0	1	1	1	1
1	0	0	1	1	0	0	1
0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1
1	1	1	1	1	1	1	1
1	0	0	1	0	1	1	0
0	0	1	1	1	1	0	0
0	1	0	1	1	0	1	0
1	1	1	1	0	0	0	0
0	1	1	0	0	1	1	0
1	1	0	0	1	1	0	0
1	0	1	0	1	0	1	0
0	0	0	0	0	0	0	0

$$\text{1-transitive} : \forall i, j \in [N], \exists \pi : [N] \to [N], \pi(i) = j \wedge \pi(\mathcal{C}) = \mathcal{C}$$

Code is 2-Transitive

0	1	1	0	1	0	0	1
1	1	0	0	0	0	1	1
1	0	1	0	0	1	0	1
0	0	0	0	1	1	1	1
1	0	0	1	1	0	0	1
0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1
1	1	1	1	1	1	1	1
1	0	0	1	0	1	1	0
0	0	1	1	1	1	0	0
0	1	0	1	1	0	1	0
1	1	1	1	0	0	0	0
0	1	1	0	0	1	1	0
1	1	0	0	1	1	0	0
1	0	1	0	1	0	1	0
0	0	0	0	0	0	0	0

0	1	1	0	1	0	0	1
1	1	0	0	0	0	1	1
1	0	1	0	0	1	0	1
0	0	0	0	1	1	1	1
1	0	0	1	1	0	0	1
0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1
1	1	1	1	1	1	1	1
1	0	0	1	0	1	1	0
0	0	1	1	1	1	0	0
0	1	0	1	1	0	1	0
1	1	1	1	0	0	0	0
0	1	1	0	0	1	1	0
1	1	0	0	1	1	0	0
1	0	1	0	1	0	1	0
0	0	0	0	0	0	0	0

Code is 2-Transitive

0	1	1	0	1	0	0	1
1	1	0	0	0	0	1	1
1	0	1	0	0	1	0	1
0	0	0	0	1	1	1	1
1	0	0	1	1	0	0	1
0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1
1	1	1	1	1	1	1	1
1	0	0	1	0	1	1	0
0	0	1	1	1	1	0	0
0	1	0	1	1	0	1	0
1	1	1	1	0	0	0	0
0	1	1	0	0	1	1	0
1	1	0	0	1	1	0	0
1	0	1	0	1	0	1	0
0	0	0	0	0	0	0	0

0	1	1	0	1	0	0	1
1	1	0	0	0	0	1	1
1	0	1	0	0	1	0	1
0	0	0	0	1	1	1	1
1	0	0	1	1	0	0	1
0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1
1	1	1	1	1	1	1	1
1	0	0	1	0	1	1	0
0	0	1	1	1	1	0	0
0	1	0	1	1	0	1	0
1	1	1	1	0	0	0	0
0	1	1	0	0	1	1	0
1	1	0	0	1	1	0	0
1	0	1	0	1	0	1	0
0	0	0	0	0	0	0	0

Code is 2-Transitive

0	1	1	0	1	0	0	1
1	1	0	0	0	0	1	1
1	0	1	0	0	1	0	1
0	0	0	0	1	1	1	1
1	0	0	1	1	0	0	1
0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1
1	1	1	1	1	1	1	1
1	0	0	1	0	1	1	0
0	0	1	1	1	1	0	0
0	1	0	1	1	0	1	0
1	1	1	1	0	0	0	0
0	1	1	0	0	1	1	0
1	1	0	0	1	1	0	0
1	0	1	0	1	0	1	0
0	0	0	0	0	0	0	0

0	1	1	0	1	0	0	1
1	1	0	0	0	0	1	1
1	0	1	0	0	1	0	1
0	0	0	0	1	1	1	1
1	0	0	1	1	0	0	1
0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1
1	1	1	1	1	1	1	1
1	0	0	1	0	1	1	0
0	0	1	1	1	1	0	0
0	1	0	1	1	0	1	0
1	1	1	1	0	0	0	0
0	1	1	0	0	1	1	0
1	1	0	0	1	1	0	0
1	0	1	0	1	0	1	0
0	0	0	0	0	0	0	0

Code is 2-Transitive

0	1	1	0	1	0	0	1
1	1	0	0	0	0	1	1
1	0	1	0	0	1	0	1
0	0	0	0	1	1	1	1
1	0	0	1	1	0	0	1
0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1
1	1	1	1	1	1	1	1
1	0	0	1	0	1	1	0
0	0	1	1	1	1	0	0
0	1	0	1	1	0	1	0
1	1	1	1	0	0	0	0
0	1	1	0	0	1	1	0
1	1	0	0	1	1	0	0
1	0	1	0	1	0	1	0
0	0	0	0	0	0	0	0

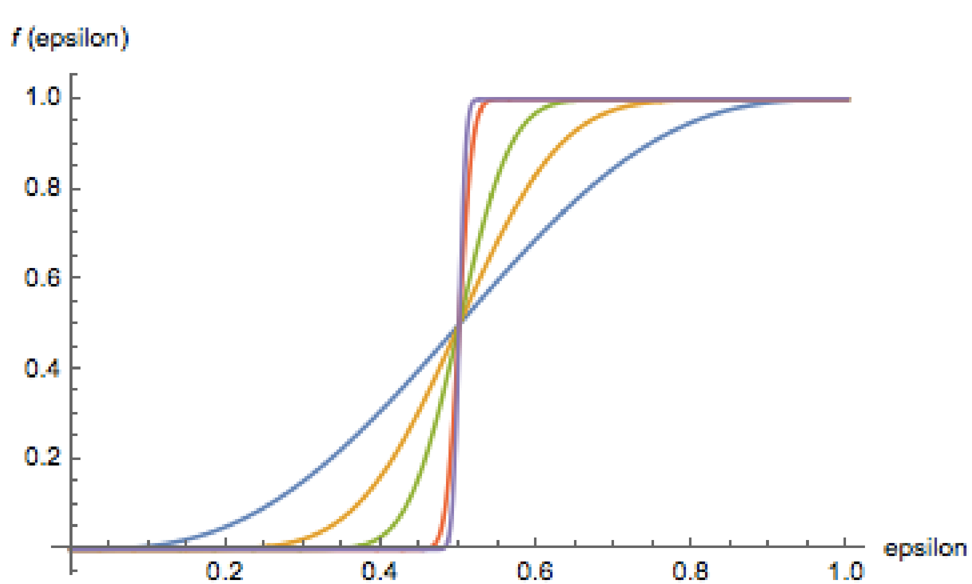
0	1	1	0	1	0	0	1
1	1	0	0	0	0	1	1
1	0	1	0	0	1	0	1
0	0	0	0	1	1	1	1
1	0	0	1	1	0	0	1
0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1
1	1	1	1	1	1	1	1
1	0	0	1	0	1	1	0
0	0	1	1	1	1	0	0
0	1	0	1	1	0	1	0
1	1	1	1	0	0	0	0
0	1	1	0	0	1	1	0
1	1	0	0	1	1	0	0
1	0	1	0	1	0	1	0
0	0	0	0	0	0	0	0

Code is 2-Transitive

0	1	1	0	1	0	0	1
1	1	0	0	0	0	1	1
1	0	1	0	0	1	0	1
0	0	0	0	1	1	1	1
1	0	0	1	1	0	0	1
0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1
1	1	1	1	1	1	1	1
1	0	0	1	0	1	1	0
0	0	1	1	1	1	0	0
0	1	0	1	1	0	1	0
1	1	1	1	0	0	0	0
0	1	1	0	0	1	1	0
1	1	0	0	1	1	0	0
1	0	1	0	1	0	1	0
0	0	0	0	0	0	0	0

0	1	1	0	1	0	0	1
1	1	0	0	0	0	1	1
1	0	1	0	0	1	0	1
0	0	0	0	1	1	1	1
1	0	0	1	1	0	0	1
0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1
1	1	1	1	1	1	1	1
1	0	0	1	0	1	1	0
0	0	1	1	1	1	0	0
0	1	0	1	1	0	1	0
1	1	1	1	0	0	0	0
0	1	1	0	0	1	1	0
1	1	0	0	1	1	0	0
1	0	1	0	1	0	1	0
0	0	0	0	0	0	0	0

Symmetric Monotone Sets Have Sharp Thresholds



Convex Geometric Analysis
MSRI Publications
Volume **34**, 1998

Threshold Intervals under Group Symmetries

JEAN BOURGAIN AND GIL KALAI

ABSTRACT. This article contains a brief description of new results on threshold phenomena for monotone properties of random systems. These results sharpen recent estimates of Talagrand, Russo and Margulis. In particular, for isomorphism invariant properties of random graphs, we get a threshold whose length is only of order $1/(\log n)^{2-\epsilon}$, instead of previous estimates of the order $1/\log n$. The new ingredients are delicate inequalities in the spirit of harmonic analysis on the Cantor group.

A subset A of $\{0,1\}^n$ is called monotone if the conditions $x \in A$, x' and $x_i \le x'_i$ for $i = 1, \dots, n$ imply $x' \in A$. For $0 \le p \le 1$, define μ_p measure on $\{0,1\}^n$ with weights $1-p$ at 0 and p at 1. Thus

$\mu_p(\{x\}) = (1-p)^{n-j} p^j$ where $j = \#\{i = 1, \dots, n \mid$

If A is monotone, then $\mu_p(A)$ is clearly an increasing functi
A as a "property", one observes in many cases a threshol
sense that $\mu_p(A)$ jumps from near 0 to near 1 in a shor
Well known examples of these phase transitions appear
of random graphs. A general understanding of such
pursued by various authors (see for instance Marguli
out that this phenomenon occurs as soon as A de
coordinate (Russo's zero-one law). A precise st
[T] in the form of the following inequality.
Define for $i = 1, \dots, n$

$A_i = \{x \in \{0,1\}^n \mid$

where $U_i(x)$ is obtained by replaceme
leaving the other coordinates uncha
the i-th coordinate (with respect

PROCEEDINGS OF THE
AMERICAN MATHEMATICAL SOCIETY
Volume 124, Number 10, October 1996

EVERY MONOTONE GRAPH PROPERTY HAS A SHARP THRESHOLD

EHUD FRIEDGUT AND GIL KALAI

(Communicated by Jeffry N. Kahn)

ABSTRACT. In their seminal work which initiated random graph theory Erdös and Rényi discovered that many graph properties have sharp thresholds as the number of vertices tends to infinity. We prove a conjecture of Linial that every monotone graph property has a sharp threshold. This follows from the following theorem.

Let $V_n(p) = \{0,1\}^n$ denote the Hamming space endowed with the probability measure μ_p defined by $\mu_p(\epsilon_1, \epsilon_2, \dots, \epsilon_n) = p^k \cdot (1-p)^{n-k}$, where $k = \epsilon_1 + \epsilon_2 + \dots + \epsilon_n$. Let A be a monotone subset of V_n. We say that A is symmetric if there is a transitive permutation group Γ on $\{1, 2, \dots, n\}$ such that A is invariant under Γ.

Theorem. *For every symmetric monotone A, if $\mu_p(A) > \epsilon$ then $\mu_q(A) > 1-\epsilon$ for $q = p + c_1 \log(1/2\epsilon)/\log n$. ($c_1$ is an absolute constant.)*

1. GRAPH PROPERTIES

A *graph property* is a property of graphs which depends only on their isomorphism class. Let P be a monotone graph property; that is, if a graph G satisfies P then every graph H on the same set of vertices, which contains G as a subgraph satisfies P as well. Examples of such properties are: G is connected, G is Hamiltonian, G contains a clique (=complete subgraph) of size t, G is not planar, the clique number of G is larger than that of its complement, the diameter of G is at most s, etc.

For a property P of graphs with a fixed set of n vertices we will denote by $\mu_p(P)$ the probability that a random graph on n vertices with edge probability p satisfies P. The theory of random graphs was founded by Erdös and Rényi [8, 4], and one of their significant discoveries was the existence of sharp thresholds for various graph properties; that is, the transition from a property being very unlikely to it being very likely is very swift. Many results on various aspects of this phenomenon have appeared since then. In what follows c_1, c_2, etc. are universal constants.

Theorem 1.1. *Let P be any monotone property of graphs on n vertices. If $\mu_p(P) > \epsilon$ then $\mu_q(P) > 1-\epsilon$ for $q = p + c_1 \log(1/2\epsilon)/\log n$.*

Received by the editors March 27, 1995.
1991 *Mathematics Subject Classification.* Primary 05C80, 28A35, 60K35.
Research supported in part by grants from the Israeli Academy of Sciences, the U.S.-Israel Binational Science Foundation, the Sloan foundation and by a grant from the state of Niedersachsen.

Monotone Sets

$$\Omega \subseteq \{0,1\}^N$$

Monotone Sets

$$\Omega \subseteq \{0,1\}^N$$

$$\Omega \quad \text{monotone} \qquad \Leftrightarrow \qquad \omega \succeq \omega' \Rightarrow \mathbb{1}_\Omega(\omega) \geq \mathbb{1}_\Omega(\omega')$$
$$\text{for all } \omega, \omega' \in \Omega$$

by adding more 1s, one remains in set

Monotone Sets

$$\Omega \subseteq \{0,1\}^N$$

$$\Omega \quad \text{monotone} \qquad \Leftrightarrow \qquad \omega \succeq \omega' \Rightarrow \mathbb{1}_\Omega(\omega) \geq \mathbb{1}_\Omega(\omega')$$
$$\text{for all } \omega, \omega' \in \Omega$$

by adding more 1s, one remains in set

$$\Omega = \left\{ \begin{array}{c} (1,1,0) \\ (0,1,1) \\ (1,1,1) \end{array} \right\} \quad \text{monotone}$$

Monotone Sets

$$\Omega \subseteq \{0,1\}^N$$

$$\Omega \quad \text{monotone} \qquad \Leftrightarrow \qquad \omega \succeq \omega' \Rightarrow \mathbb{1}_\Omega(\omega) \geq \mathbb{1}_\Omega(\omega')$$
$$\text{for all } \omega, \omega' \in \Omega$$

by adding more 1s, one remains in set

$$\Omega = \left\{ \begin{array}{c} (1,1,0) \\ (0,1,1) \\ \xcancel{(1,1,1)} \end{array} \right\} \quad \xcancel{\text{monotone}}$$

Symmetric Sets

$$\Omega \subseteq \{0,1\}^N$$

Symmetric Sets

$$\Omega \subseteq \{0,1\}^N$$

Ω symmetric $\Leftrightarrow$ the set Ω is 1-transitive
(e.g., preserved by cyclic shift)

Symmetric Sets

$$\Omega \subseteq \{0,1\}^N$$

Ω symmetric $\Leftrightarrow$ the set Ω is 1-transitive

(e.g., preserved by cyclic shift)

$$\Omega = \left\{ \begin{array}{c} (1,1,0) \\ (0,1,1) \\ (1,0,1) \\ (1,1,1) \end{array} \right\} \quad \text{symmetric}$$

Symmetric Sets

$$\Omega \subseteq \{0,1\}^N$$

Ω symmetric $\Leftrightarrow$ the set Ω is 1-transitive (e.g., preserved by cyclic shift)

$$\Omega = \left\{ \begin{array}{c} (1,1,0) \\ (0,1,1) \\ \cancel{(1,0,1)} \\ (1,1,1) \end{array} \right\} \quad \cancel{\text{symmetric}}$$

Probability of Set

$$\Omega \subseteq \{0,1\}^N$$

$\mu_\epsilon(\cdot)$ Bernoulli product measure with parameter ϵ

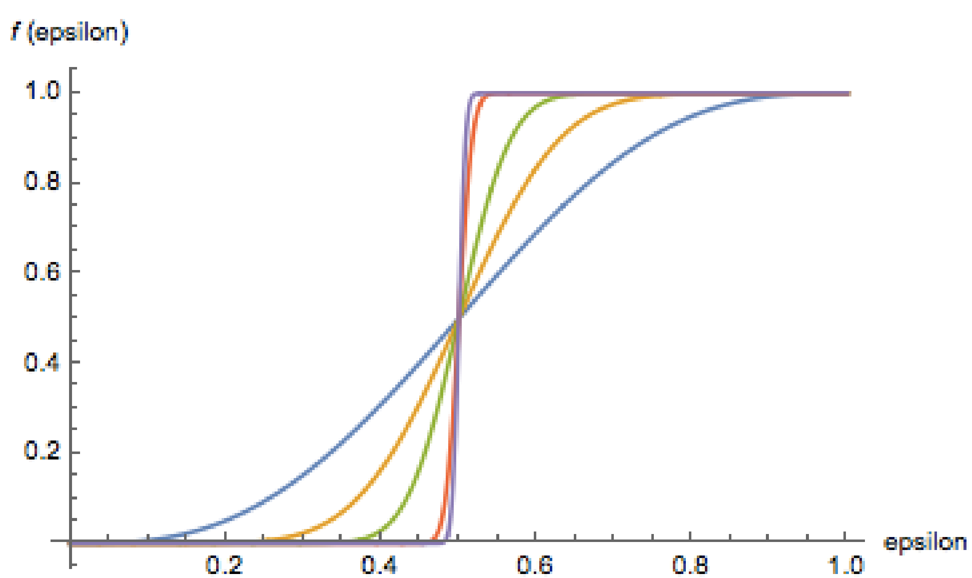
$\omega \in \{0,1\}^N$ $\mu_\epsilon(\omega) = \epsilon^{\mathrm{wt}(\omega)}(1-\epsilon)^{N-\mathrm{wt}(\omega)}$

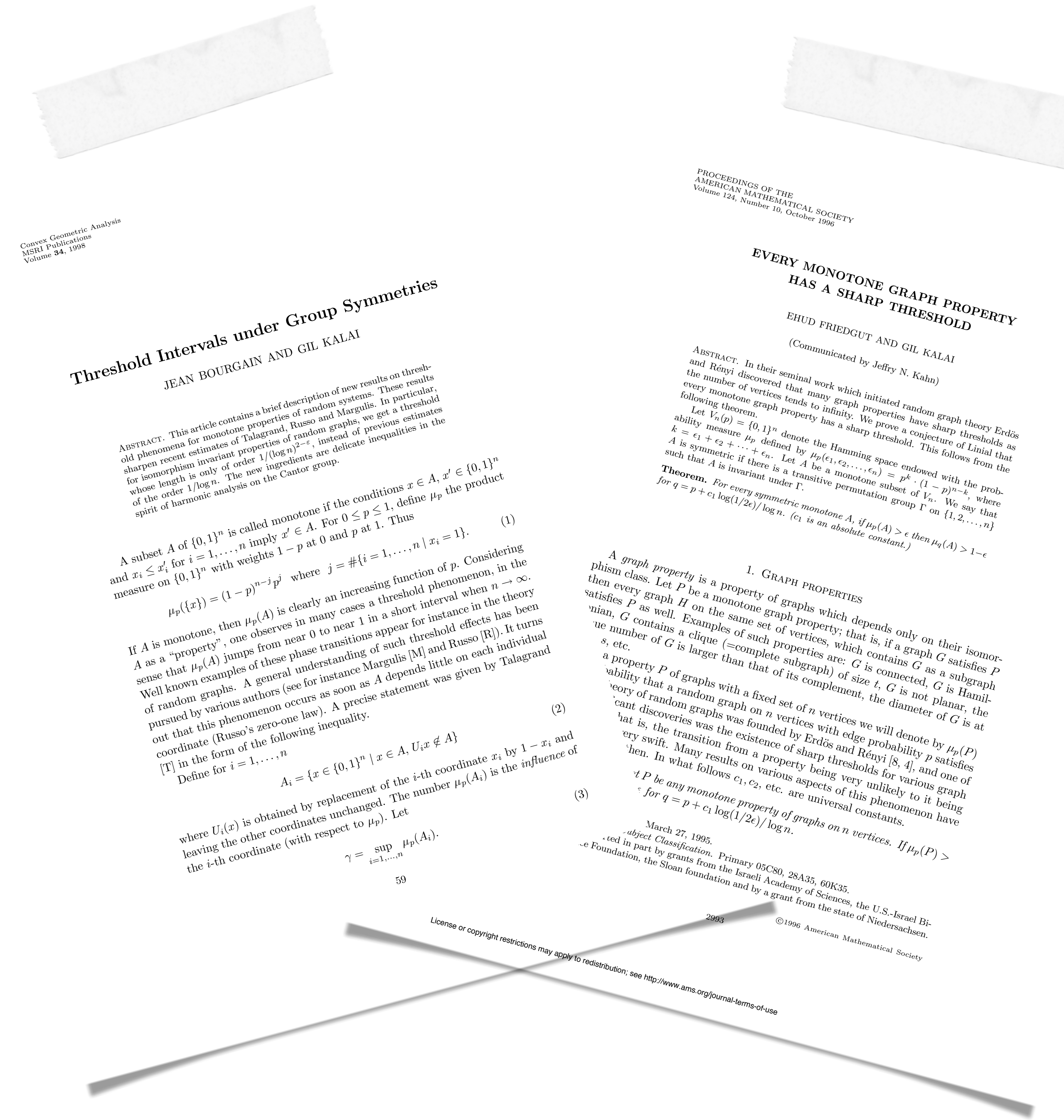
$$\mu_\epsilon(\Omega) = \sum_{\omega \in \Omega} \mu_\epsilon(\omega)$$

$\mu_\epsilon(\Omega)$ is the probability that an iid $B(\epsilon)$ vector is in Ω

later: Ω is the set of bad erasure patterns

Sharp Thresholds





Convex Geometric Analysis
MSRI Publications
Volume 34, 1998

Threshold Intervals under Group Symmetries

JEAN BOURGAIN AND GIL KALAI

Abstract. This article contains a brief description of new results on threshold phenomena for monotone properties of random systems. These results sharpen recent estimates of Talagrand, Russo and Margulis. In particular, for isomorphism invariant properties of random graphs, we get a threshold whose length is only of order $1/(\log n)^{2-\epsilon}$, instead of previous estimates of the order $1/\log n$. The new ingredients are delicate inequalities in the spirit of harmonic analysis on the Cantor group.

A subset A of $\{0,1\}^n$ is called monotone if the conditions $x \in A$, $x' \in \{0,1\}^n$ and $x_i \le x'_i$ for $i = 1, \dots, n$ imply $x' \in A$. For $0 \le p \le 1$, define μ_p the product measure on $\{0,1\}^n$ with weights $1-p$ at 0 and p at 1. Thus

$$\mu_p(\{x\}) = (1-p)^{n-j} p^j \quad \text{where} \quad j = \#\{i = 1, \dots, n \mid x_i = 1\}. \tag{1}$$

If A is monotone, then $\mu_p(A)$ is clearly an increasing function of p. Considering A as a "property", one observes in many cases a threshold phenomenon, in the sense that $\mu_p(A)$ jumps from near 0 to near 1 in a short interval when $n \to \infty$. Well known examples of these phase transitions appear for instance in the theory of random graphs. A general understanding of such threshold effects has been pursued by various authors (see for instance Margulis [M] and Russo [R]). It turns out that this phenomenon occurs as soon as A depends little on each individual coordinate (Russo's zero-one law). A precise statement was given by Talagrand [T] in the form of the following inequality.

Define for $i = 1, \dots, n$

$$A_i = \{x \in \{0,1\}^n \mid x \in A,\ U_i x \notin A\} \tag{2}$$

where $U_i(x)$ is obtained by replacement of the i-th coordinate x_i by $1 - x_i$ and leaving the other coordinates unchanged. The number $\mu_p(A_i)$ is the *influence* of the i-th coordinate (with respect to μ_p). Let

$$\gamma = \sup_{i=1,\dots,n} \mu_p(A_i). \tag{3}$$

59

PROCEEDINGS OF THE
AMERICAN MATHEMATICAL SOCIETY
Volume 124, Number 10, October 1996

EVERY MONOTONE GRAPH PROPERTY HAS A SHARP THRESHOLD

EHUD FRIEDGUT AND GIL KALAI

(Communicated by Jeffry N. Kahn)

Abstract. In their seminal work which initiated random graph theory Erdős and Rényi discovered that many graph properties have sharp thresholds as the number of vertices tends to infinity. We prove a conjecture of Linial that every monotone graph property has a sharp threshold. This follows from the following theorem.

Let $V_n(p) = \{0,1\}^n$ denote the Hamming space endowed with the probability measure μ_p defined by $\mu_p(\epsilon_1, \epsilon_2, \dots, \epsilon_n) = p^k \cdot (1-p)^{n-k}$, where $k = \epsilon_1 + \epsilon_2 + \dots + \epsilon_n$. Let A be a monotone subset of V_n. We say that A is symmetric if there is a transitive permutation group Γ on $\{1, 2, \dots, n\}$ such that A is invariant under Γ.

Theorem. *For every symmetric monotone A, if $\mu_p(A) > \epsilon$ then $\mu_q(A) > 1-\epsilon$ for $q = p + c_1 \log(1/2\epsilon)/\log n$. ($c_1$ is an absolute constant.)*

1. Graph properties

A *graph property* is a property of graphs which depends only on their isomorphism class. Let P be a monotone graph property; that is, if a graph G satisfies P then every graph H on the same set of vertices, which contains G as a subgraph satisfies P as well. Examples of such properties are: G is connected, G is Hamiltonian, G contains a clique (=complete subgraph) of size t, G is not planar, the clique number of G is larger than that of its complement, the diameter of G is at most s, etc.

For a property P of graphs with a fixed set of n vertices we will denote by $\mu_p(P)$ the probability that a random graph on n vertices with edge probability p satisfies P. The theory of random graphs was founded by Erdős and Rényi [8, 4], and one of the significant discoveries was the existence of sharp thresholds for various graph properties, that is, the transition from a property being very unlikely to it being very likely is very swift. Many results on various aspects of this phenomenon have appeared since then. In what follows c_1, c_2, etc. are universal constants.

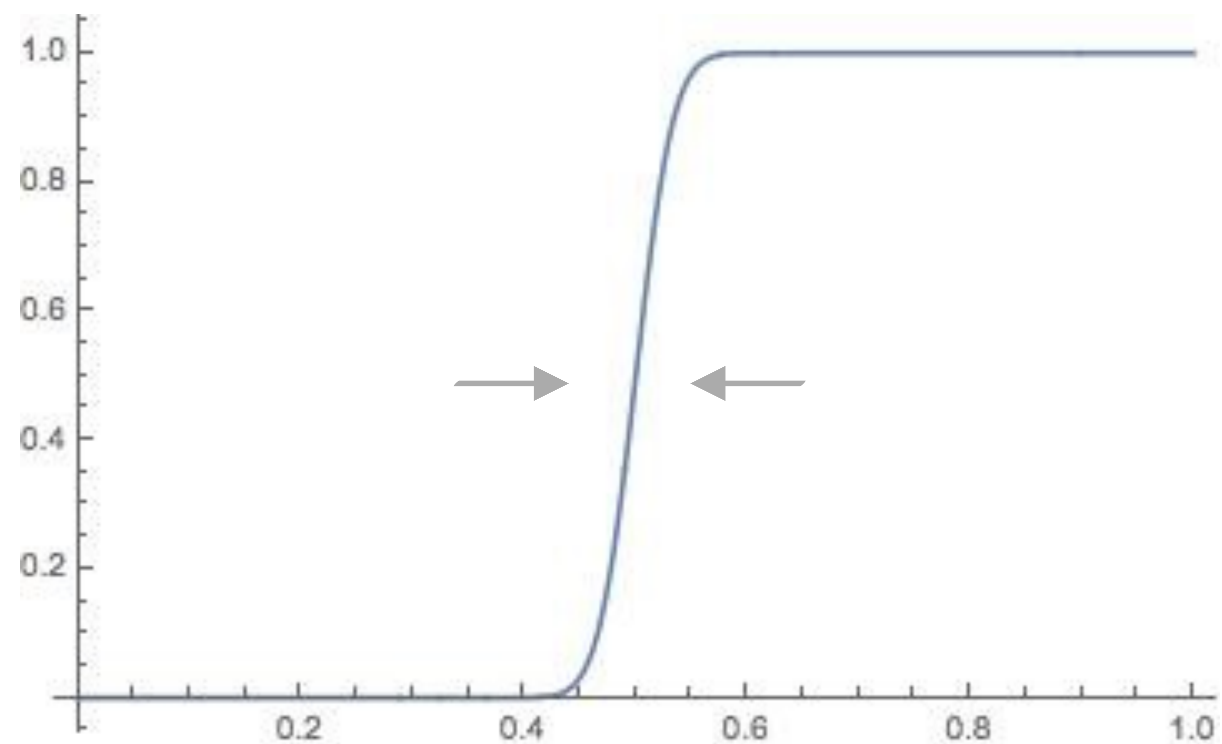
...t P be any monotone property of graphs on n vertices. If $\mu_p(P) >$ ... for $q = p + c_1 \log(1/2\epsilon)/\log n$.

... March 27, 1995.
...ubject Classification. Primary 05C80, 28A35, 60K35.
...ted in part by grants from the Israeli Academy of Sciences, the U.S.-Israel Bi-...e Foundation, the Sloan foundation and by a grant from the state of Niedersachsen.

2993

Friedgut, Kalai (1996)

$\Omega \subseteq \{0,1\}^N$, monotone, symmetric



$\mu_\epsilon(\Omega)$ goes from $\delta > 0$ to $1-\delta$ within a window of size $\dfrac{\log(\frac{1}{2\delta})}{\log(N)}$

Sharp Thresholds — Block-MAP Decoding

Combinatorics, Probability and Computing (2000) 9, 465–479. Printed in the United Kingdom

Discrete Isoperimetric Inequalities and the Probability of a Decoding Error

JEAN-PIERRE TILLICH[1] and GILLES ZÉMOR[2]

[1] LRI, bâtiment 490, Université Paris-Sud, 91405 Orsay, France (e-mail: tillich@lri.fr)

[2] École Nationale Supérieure des Télécommunications, 75 634 Paris 13, France (e-mail: zemor@infres.enst.fr)

Received 20 April 1999; revised 19 January 2000

We derive improved isoperimetric inequalities for discrete product measures on the n-dimensional cube. As a consequence, a general theorem on the threshold behaviour of monotone properties is obtained. This is then applied to coding theory when we study the probability of error after decoding.

1. Introduction

Consider the n-cube, or binary Hamming space $\boldsymbol{H}^n = \{0,1\}^n$ of dimension n, and denote by $|x|$ the weight $\sum_{i=1}^n x_i$ of a binary vector $x = (x_1, x_2, \ldots, x_n) \in \boldsymbol{H}^n$. For $0 < p < 1$ let μ_p denote the product measure on $\boldsymbol{H}^n$ defined for any subset $\Omega \subset \boldsymbol{H}^n$ by

$$\mu_p(\Omega) = \sum_{x \in \Omega} p^{|x|}(1-p)^{n-|x|}.$$

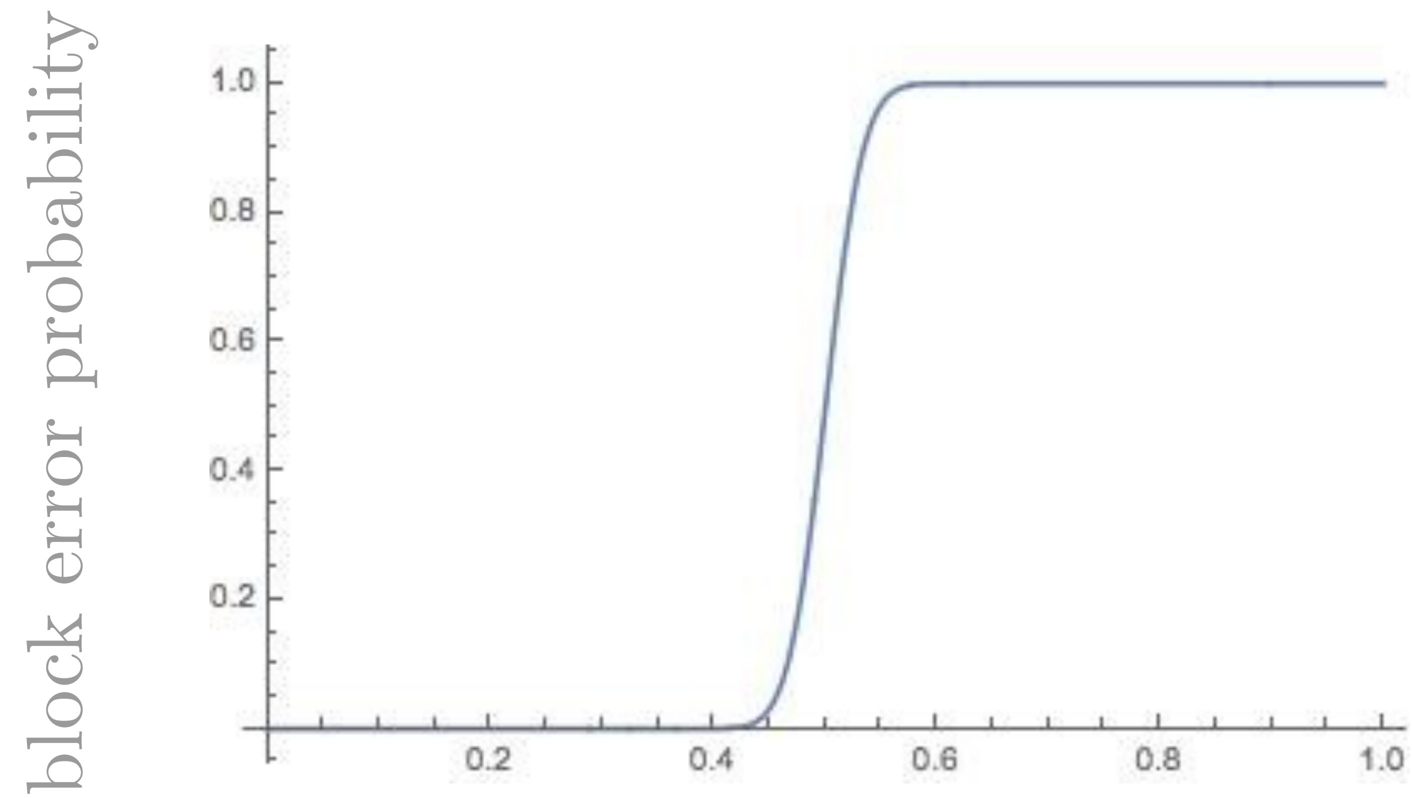
Let us write $x \preceq y$ if for any $i = 1, 2, \ldots, n$ we have $x_i \leqslant y_i$. We shall say that Ω is increasing if, for any $x \in \Omega$, $x \preceq y$ implies that y is also in Ω.

The theory of random graphs has been concerned with many increasing sets Ω and with the behaviour of the function $f(p) = \mu_p(\Omega)$. Quite often a threshold phenomenon is observed: $f(p)$ jumps from near 0 to near 1 in a short interval that shrinks as n grows.

In many cases this threshold behaviour can be proved by a direct study of $f(p)$. This has not always been successful, however, and the following indirect strategy has been investigated by a number of authors, including [4, 8, 12, 13, 14, 15, 16]: find conditions on Ω which are easy to check and which imply that $\mu_p(\Omega)$ satisfies a differential inequality

linear code, BEC or BSC, block-MAP decoding

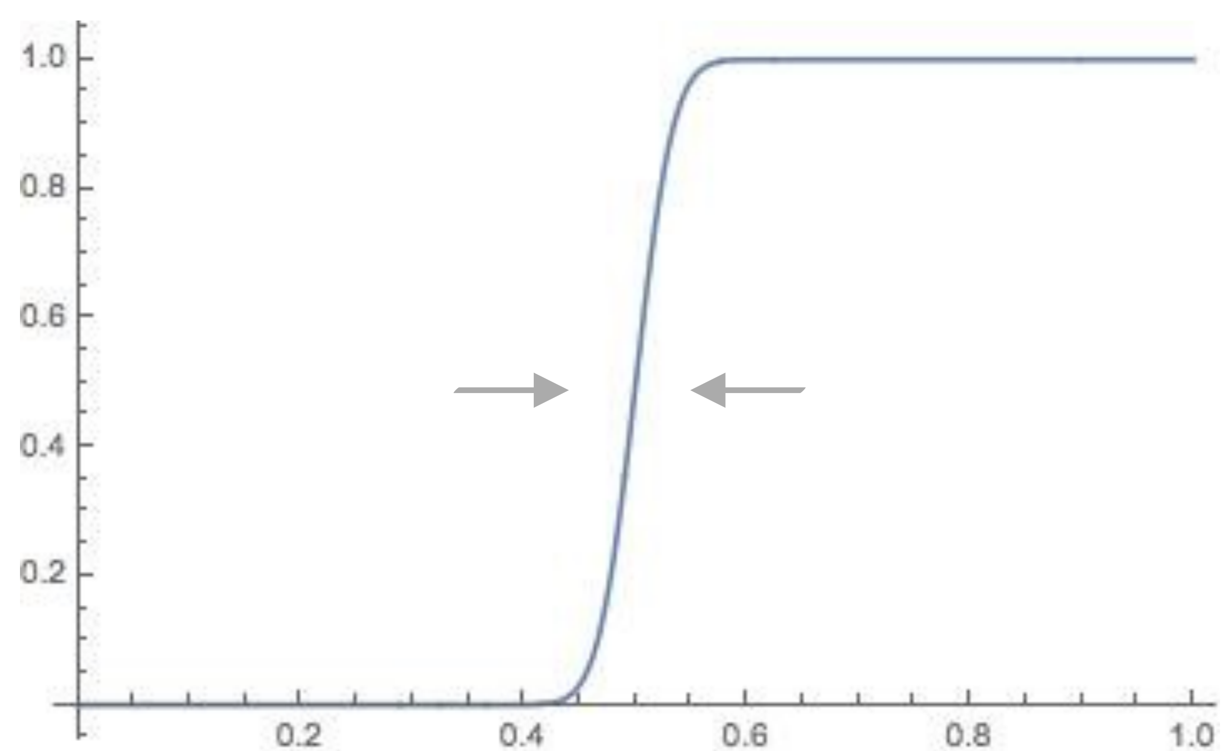
$$O(1/\sqrt{d_H})$$



linear code, BEC or BSC, block-MAP decoding

$$O(1/\sqrt{d_H})$$

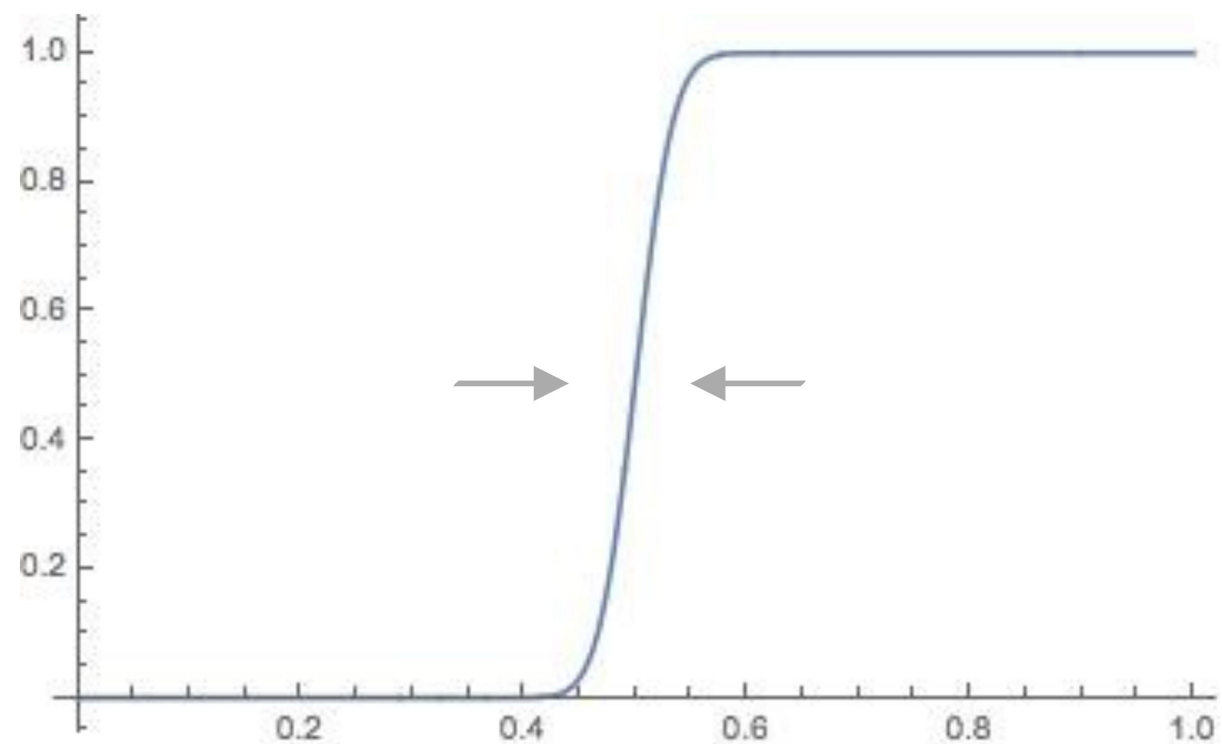
block error probability



linear code, BEC or BSC, block-MAP decoding

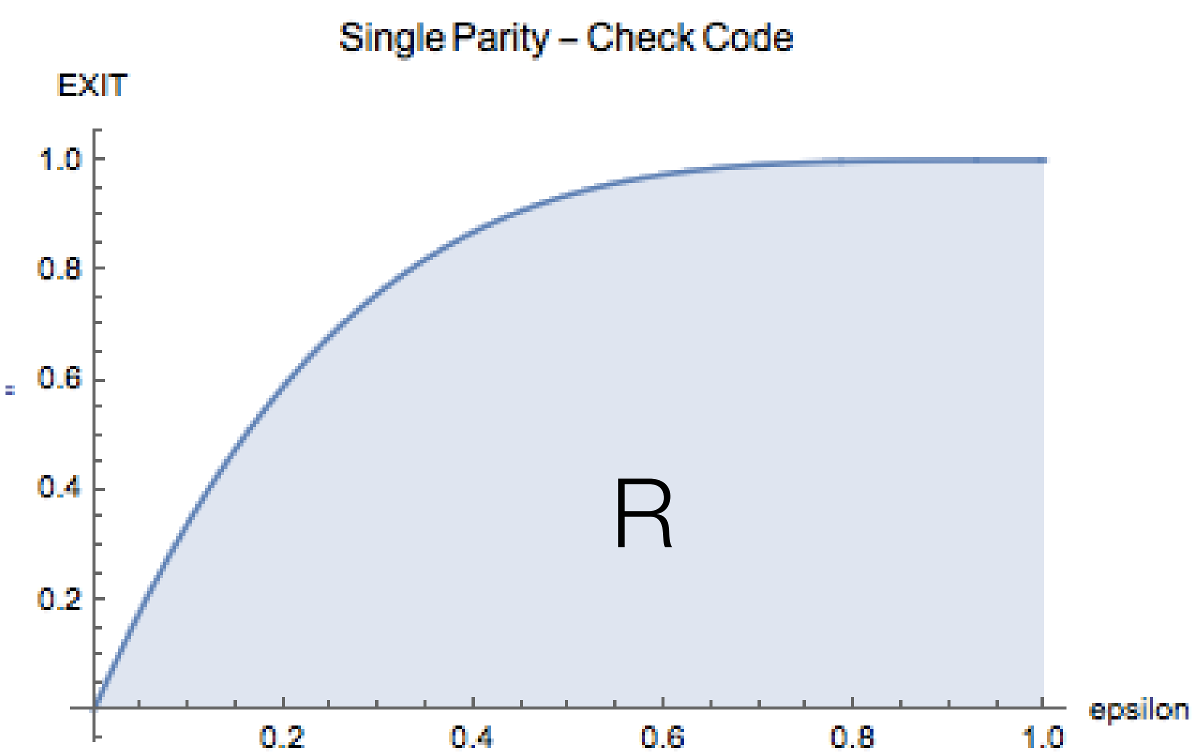
$$O(1/\sqrt{d_H})$$

block error probability



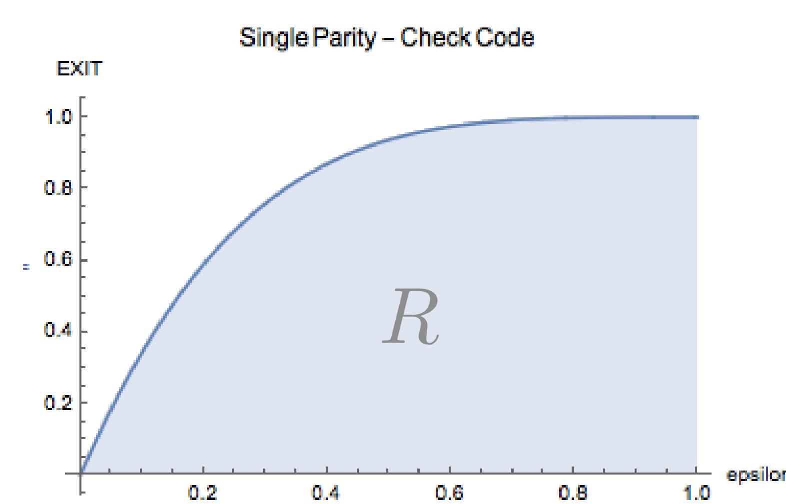
?

Exit Functions satisfy the Area Theorem



EXIT Functions satisfy the Area Theorem

$$\frac{\mathrm{d}H(X \mid Y(\epsilon))}{\mathrm{d}\epsilon} = \sum_{i=1}^{N} \underbrace{P\{\hat{x}_i^{\mathrm{MAP}}(Y_{\sim i}) = ?\}}_{h_i(\epsilon)}$$

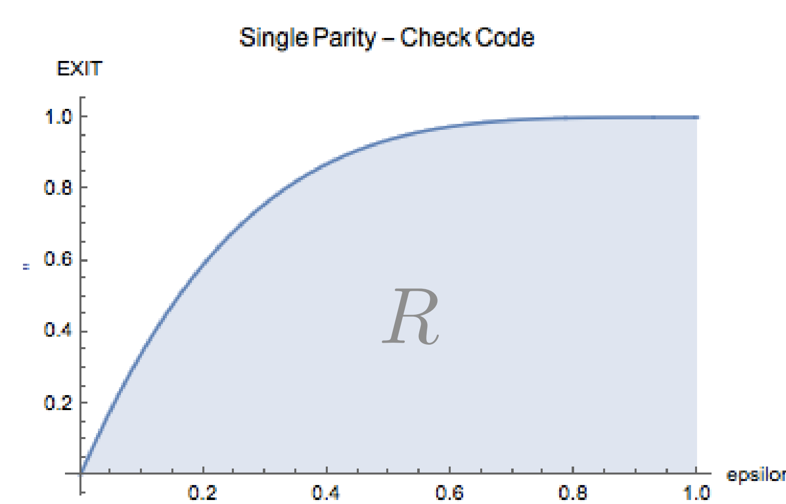


EXIT Functions satisfy the Area Theorem

$$\frac{\mathrm{d}H(X \mid Y(\epsilon))}{\mathrm{d}\epsilon} = \sum_{i=1}^{N} \frac{\partial H(X \mid Y(\epsilon_1, \cdots, \epsilon_N))}{\partial \epsilon_i}|_{\epsilon_j = \epsilon}$$

$$= \sum_{i=1}^{N} \frac{\partial [H(X_i \mid Y) + H(X_{\sim i} \mid Y, X_i)]}{\partial \epsilon_i}|_{\epsilon_j = \epsilon}$$

$$\frac{\mathrm{d}H(X \mid Y(\epsilon))}{\mathrm{d}\epsilon} = \sum_{i=1}^{N} \underbrace{P\{\hat{x}_i^{\mathrm{MAP}}(Y_{\sim i}) = ?\}}_{h_i(\epsilon)}$$



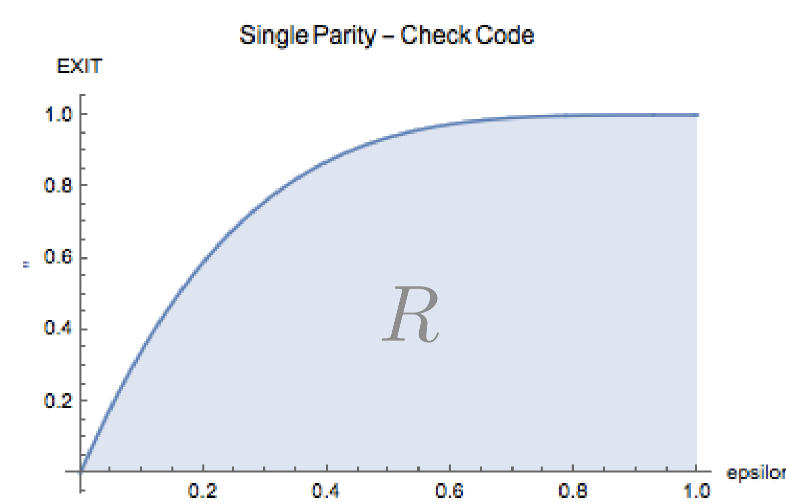
EXIT Functions satisfy the Area Theorem

$$= \sum_{i=1} \frac{\partial H(X_i \mid Y_{\sim i})\epsilon_i}{\partial \epsilon_i}\Big|_{\epsilon_j=\epsilon}$$

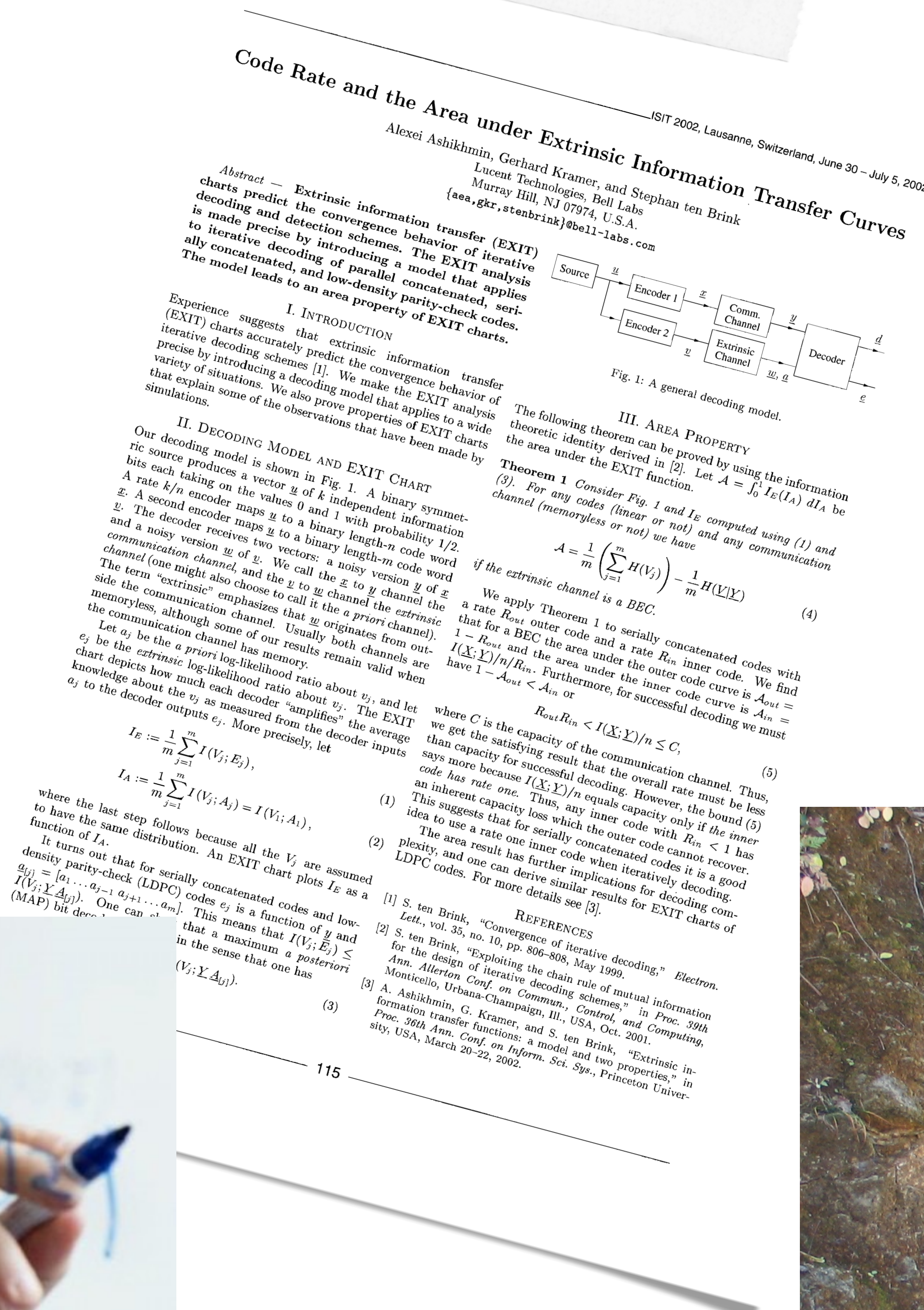
$$= \sum_{i=1}^{N} H(X_i \mid Y_{\sim i})$$

$$= \sum_{i=1}^{N} P\{\hat{x}_i^{\text{MAP}}(Y_{\sim i}) = ?\}$$

$$\frac{\mathrm{d}H(X \mid Y(\epsilon))}{\mathrm{d}\epsilon} = \sum_{i=1}^{N} \underbrace{P\{\hat{x}_i^{\text{MAP}}(Y_{\sim i}) = ?\}}_{h_i(\epsilon)}$$



EXIT Functions satisfy the Area Theorem



ISIT 2002, Lausanne, Switzerland, June 30 – July 5, 2002

Code Rate and the Area under Extrinsic Information Transfer Curves

Alexei Ashikhmin, Gerhard Kramer, and Stephan ten Brink
Lucent Technologies, Bell Labs
Murray Hill, NJ 07974, U.S.A.
{aea,gkr,stenbrink}@bell-labs.com

Abstract — **Extrinsic information transfer (EXIT) charts predict the convergence behavior of iterative decoding and detection schemes. The EXIT analysis is made precise by introducing a model that applies to iterative decoding of parallel concatenated, serially concatenated, and low-density parity-check codes. The model leads to an area property of EXIT charts.**

I. Introduction

Experience suggests that extrinsic information transfer (EXIT) charts accurately predict the convergence behavior of iterative decoding schemes [1]. We make the EXIT analysis precise by introducing a decoding model that applies to a wide variety of situations. We also prove properties of EXIT charts that explain some of the observations that have been made by simulations.

II. Decoding Model and EXIT Chart

Our decoding model is shown in Fig. 1. A binary symmetric source produces a vector $\underline{u}$ of k independent information bits each taking on the values 0 and 1 with probability 1/2. A rate k/n encoder maps $\underline{u}$ to a binary length-n code word $\underline{x}$. A second encoder maps $\underline{u}$ to a binary length-m code word $\underline{v}$. The decoder receives two vectors: a noisy version $\underline{y}$ of $\underline{x}$ and a noisy version $\underline{w}$ of $\underline{v}$. We call the $\underline{x}$ to $\underline{y}$ channel the *communication channel*, and the $\underline{v}$ to $\underline{w}$ channel the *extrinsic channel* (one might also choose to call it the *a priori* channel). The term "extrinsic" emphasizes that $\underline{w}$ originates from outside the communication channel. Usually both channels are memoryless, although some of our results remain valid when the communication channel has memory.

Let a_j be the *a priori* log-likelihood ratio about v_j, and let e_j be the *extrinsic* log-likelihood ratio about v_j. The EXIT chart depicts how much each decoder "amplifies" the average knowledge about the v_j as measured from the decoder inputs a_j to the decoder outputs e_j. More precisely, let

$$I_E := \frac{1}{m}\sum_{j=1}^{m} I(V_j; E_j), \tag{1}$$

$$I_A := \frac{1}{m}\sum_{j=1}^{m} I(V_j; A_j) = I(V_1; A_1), \tag{2}$$

where the last step follows because all the V_j are assumed to have the same distribution. An EXIT chart plots I_E as a function of I_A.

It turns out that for serially concatenated codes and low-density parity-check (LDPC) codes e_j is a function of $\underline{y}$ and $\underline{a}_{[j]} = [a_1 \ldots a_{j-1}\, a_{j+1} \ldots a_m]$. This means that $I(V_j; \underline{E}_j) \leq$ $I(V_j; \underline{Y}\, \underline{A}_{[j]})$. One can … that a maximum *a posteriori* (MAP) bit dec… in the sense that one has

… $(V_j; \underline{Y}\, \underline{A}_{[j]})$. (3)

Fig. 1: A general decoding model.

III. Area Property

The following theorem can be proved by using the information theoretic identity derived in [2]. Let $\mathcal{A} = \int_0^1 I_E(I_A)\, dI_A$ be the area under the EXIT function.

Theorem 1 *Consider Fig. 1 and I_E computed using (1) and (3). For any codes (linear or not) and any communication channel (memoryless or not) we have*

$$\mathcal{A} = \frac{1}{m}\left(\sum_{j=1}^{m} H(V_j)\right) - \frac{1}{m} H(\underline{V}|\underline{Y}) \tag{4}$$

if the extrinsic channel is a BEC.

We apply Theorem 1 to serially concatenated codes with a rate R_{out} outer code and a rate R_{in} inner code. We find that for a BEC the area under the outer code curve is $\mathcal{A}_{out} = 1 - R_{out}$ and the area under the inner code curve is $\mathcal{A}_{in} = I(\underline{X};\underline{Y})/n/R_{in}$. Furthermore, for successful decoding we must have $1 - \mathcal{A}_{out} < \mathcal{A}_{in}$ or

$$R_{out}R_{in} < I(\underline{X};\underline{Y})/n \leq C, \tag{5}$$

where C is the capacity of the communication channel. Thus, we get the satisfying result that the overall rate must be less than capacity for successful decoding. However, the bound (5) says more because $I(\underline{X};\underline{Y})/n$ equals capacity only if *the inner code has rate one*. Thus, any inner code with $R_{in} < 1$ has an inherent capacity loss which the outer code cannot recover. This suggests that for serially concatenated codes it is a good idea to use a rate one inner code when iteratively decoding.

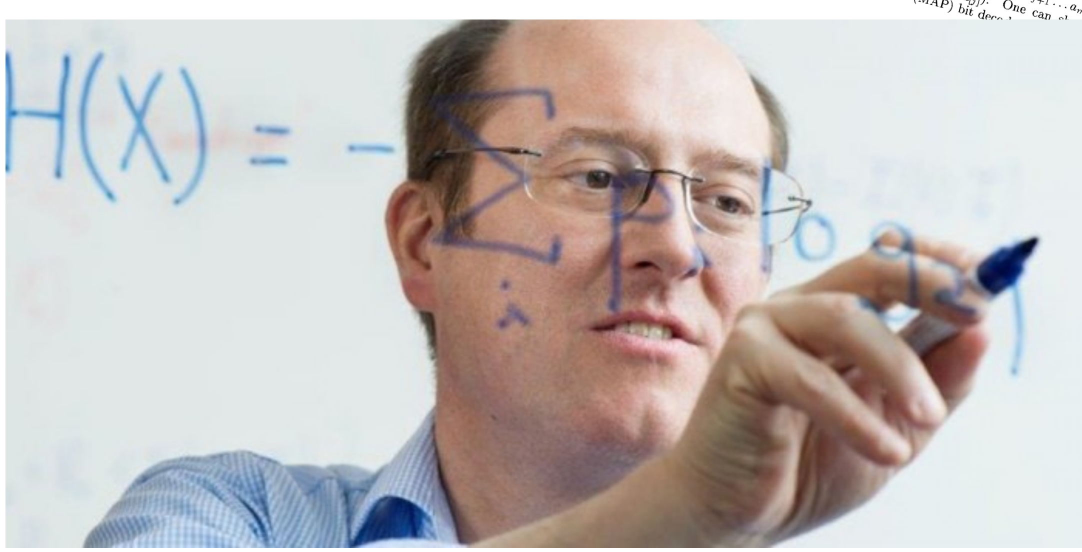
The area result has further implications for decoding complexity, and one can derive similar results for EXIT charts of LDPC codes. For more details see [3].

References

[1] S. ten Brink, "Convergence of iterative decoding," *Electron. Lett.*, vol. 35, no. 10, pp. 806–808, May 1999.

[2] S. ten Brink, "Exploiting the chain rule of mutual information for the design of iterative decoding schemes," in *Proc. 39th Ann. Allerton Conf. on Commun., Control, and Computing*, Monticello, Urbana-Champaign, Ill., USA, Oct. 2001.

[3] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: a model and two properties," in *Proc. 36th Ann. Conf. on Inform. Sci. Sys.*, Princeton University, USA, March 20–22, 2002.



EXIT Functions satisfy the Area Theorem

Life Above Threshold:
From List Decoding to Area Theorem and MSE[1]

Cyril Méasson and Rüdiger Urbanke
EPFL, I&C
CH-1015 Lausanne, Switzerland
e-mail: cyril.measson@epfl.ch
ruediger.urbanke@epfl.ch

Andrea Montanari
LPTENS (UMR 8549, CNRS et ENS)
24, rue Lhomond, 75231
Paris CEDEX 05, France
e-mail: montanar@lpt.ens.fr

Tom Richardson
Flarion Technologies
Bedminster, NJ, USA-07921
e-mail: richardson@flarion.com

v2 [cs.IT] 14 Nov 2004

Abstract — **We consider communication over memoryless channels using low-density parity-check code ensembles above the iterative (belief propagation) threshold. What is the computational complexity of decoding (i.e., of reconstructing *all* the typical input codewords for a given channel output) in this regime? We define an algorithm accomplishing this task and analyze its typical performance. The behavior of the new algorithm can be expressed in purely information-theoretical terms. Its analysis provides an alternative proof of the area theorem for the binary erasure channel. Finally, we explain how the area theorem is generalized to arbitrary memoryless channels. We note that the recently discovered relation between mutual information and minimal square error is an instance of the area theorem in the setting of Gaussian channels.**

I. Introduction

The analysis of iterative coding systems has been extremely effective in determining the conditions for successful communication. The single most important prediction in this context is the existence of a threshold noise level below which the bit error rate vanishes (as the blocklength and the number of iterations diverge). The threshold can be computed for a large ariety of code ensembles using density evolution.

On the other hand, understanding the behavior of these tems *above threshold* is largely an open issue. Since in this me the bit error rate remains bounded away from zero, one wonder about the motivation for such an investigation. an think of three possible answers: (*i*) It is intellectu-ustrating to have an "half-complete" theory of iterative ng. Moreover this theory has poor connections with l issues such as the behavior of the same codes under m likelihood (ML) decoding. (*ii*) Loopy belief prop-has stimulated a considerable interest as a general-nference algorithm for graphical models. However, very few applications where its effectiveness can be athematically. Decoding below threshold is prob-ost prominent of such examples and one may hope upon this success. (*iii*) There are communication contexts in which one is is interested in reproducing some in-

[1]Copyright 2004 IEEE. Published in the 2004 IEEE Information Theory Workshop (ITW 2004), scheduled for October 24-29, 2004 at the Riverwalk Marriott in San Antonio, Texas, USA. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works, must be obtained from the IEEE. Contact: Manager, Copyrights and Permissions / IEEE Service Center / 445 Hoes Lane / P.O. Box 1331 / Piscataway, NJ 08855-1331, USA. Telephone: + Intl. 732-562-3966.

formation within a pre-established tolerance, rather than exactly. There are indications that iterative methods can play an important role also in such contexts. If this is the case, one will necessarily operate in the above-threshold regime.

Consider, for the sake of simplicity, communication over a memoryless channel using random elements from a standard low-density parity-check (LDPC) code ensemble. Assume moreover that the noise level is greater than the threshold one. There are two natural theoretical problems one can address in this regime: (A) How many channel inputs correspond to a given typical output? (B) How hard is to construct *all* of them? Answering question (A) amounts co computing the conditional entropy $H(X_1^n|Y_1^n)$ of the chann input given the output (here n is the blocklength). We ex pect this entropy to become of order $O(n)$ at large enough noise. We call the minimum noise level for this to be the case, the ML threshold. ML decoding is bound to fail above this threshold.

The second question is apparently far from Information Theory and in any case very difficult to answer. The naive expectation would be that reconstructing all the typical codewords becomes harder as their conditional entropy gets larger.

In this paper we report some recent progress on both of the questions outlined above. In Secs. II and III we reconsider the binary erasure channel (BEC). We define a natural extension of the belief propagation decoder which reconstruct all the codewords compatible with a given channel output. The new algorithm ('Maxwell decoder') thus performs a 'complete' list decoding, and is based on the general message-passing philosophy. Below the iterative threshold, it coincides with belief propagation decoding and its complexity is linear in the blocklength. Above the iterative threshold, its complexity becomes exponential. Its behavior can be analyzed precisely, and provides answers both questions (A) and (B) above (within this circumscribed context). Surprisingly, the resulting picture is most easily conveyed using a well-known information theoretic characterization of the code: the EXIT curve. As a byproduct, we obtain an alternative proof of the area theorem for the BEC.

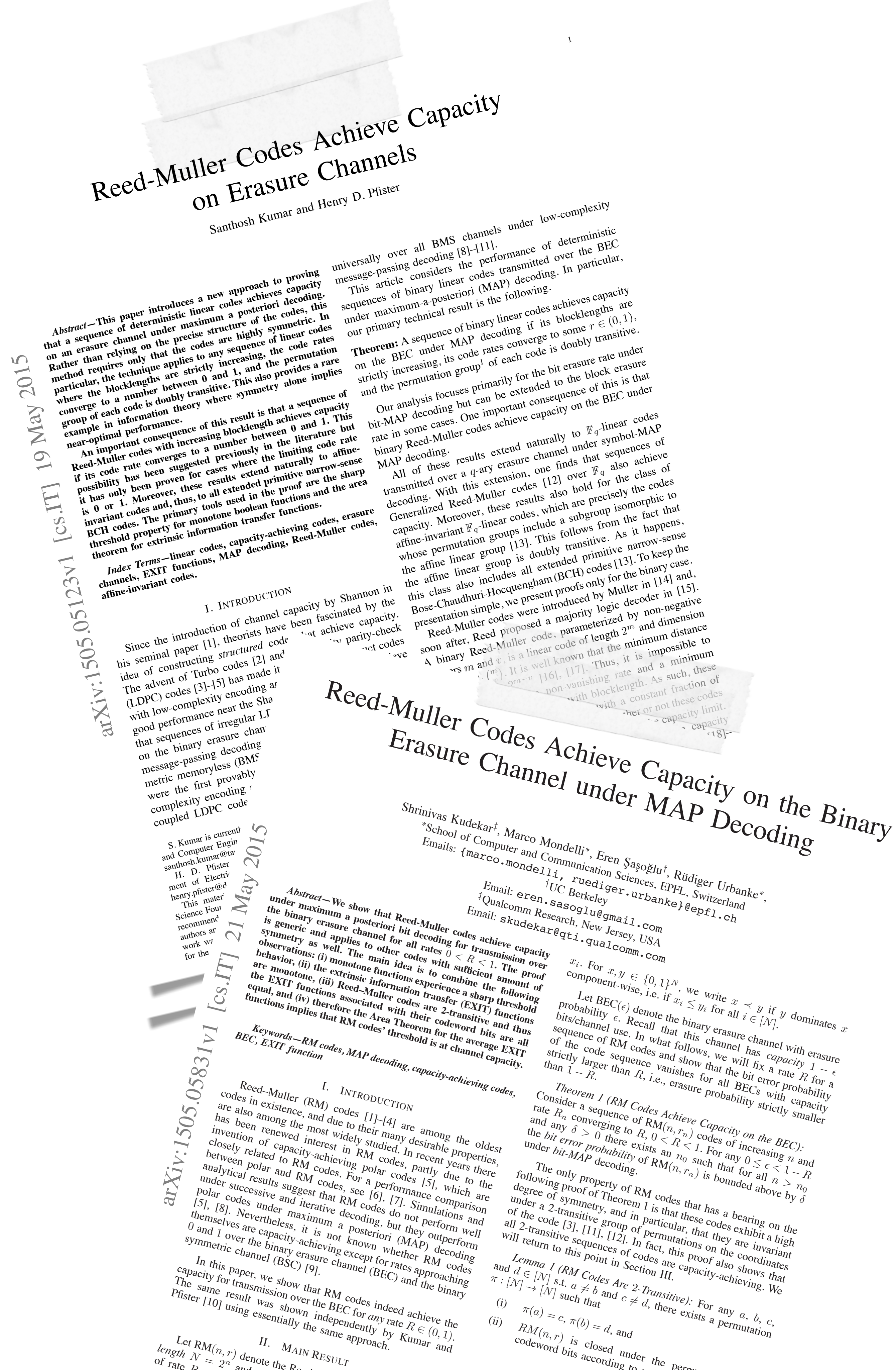
The connection between the EXIT curve and Maxwell decoder is not a peculiarity of the binary erasure channel, and has instead a rather fundamental origin. The algorithm progressively reduces the uncertainty on the transmitted bits. This can be regarded as an effective change of the noise level of the communication channel. The EXIT curve describe the response of the bits (i.e., the change of the bit uncertainty) to a change in the noise level. The area theorem is obtained when integrating this response: the total bit uncertainty at maximal noise level (the code rate) is thus given by an integral of the EXIT curve.

In Sec. IV, we explain how to generalize these ideas to ar-

Ingredients

- *RM codes are 2-transitive*
- *symmetric monotone sets have sharp thresholds*
- *EXIT functions satisfy the Area Theorem*

The Proof


Reed-Muller Codes Achieve Capacity on Erasure Channels
Santhosh Kumar and Henry D. Pfister
arXiv:1505.05123v1 [cs.IT] 19 May 2015
Abstract—This paper introduces a new approach to proving that a sequence of deterministic linear codes achieves capacity on an erasure channel under maximum a posteriori decoding. Rather than relying on the precise structure of the codes, this method requires only that the codes are highly symmetric. In particular, the technique applies to any sequence of linear codes where the blocklengths are strictly increasing, the code rates converge to a number between 0 and 1, and the permutation group of each code is doubly transitive. This also provides a rare example in information theory where symmetry alone implies near-optimal performance.
I. Introduction
Reed-Muller Codes Achieve Capacity on the Binary Erasure Channel under MAP Decoding
Shrinivas Kudekar, Marco Mondelli, Eren Şaşoğlu, Rüdiger Urbanke
arXiv:1505.05831v1 [cs.IT] 21 May 2015
I. Introduction
II. Main Result

Ω_i ... set of erasure patterns that cause trouble for bit i

$$\hat{x}_i^{\mathrm{MAP}}(Y_{\sim i}) = ?$$

Ω_i ... set of erasure patterns that cause trouble for bit i

$$\hat{x}_i^{\mathrm{MAP}}(Y_{\sim i}) = ?$$

$$\Omega_i \subseteq \{0,1\}^{N-1}$$

Ω_i ... set of erasure patterns that cause trouble for bit i

$$\hat{x}_i^{\mathrm{MAP}}(Y_{\sim i}) = ?$$

$$\Omega_i \subseteq \{0,1\}^{N-1}$$

$$\begin{array}{lcccccccc} & & & & & i & & & \\ & & & & & \downarrow & & & \\ \omega \in \{0,1\}^{N-1} & 0 & 1 & 1 & 0 & & 1 & 0 & 1 \\ c \in \mathcal{C} \subset \{0,1\}^N & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}$$

$\omega \in \Omega_i$ iff $\exists c \in \mathcal{C} : c_i = 1 \wedge c_{\sim i} \prec \omega$

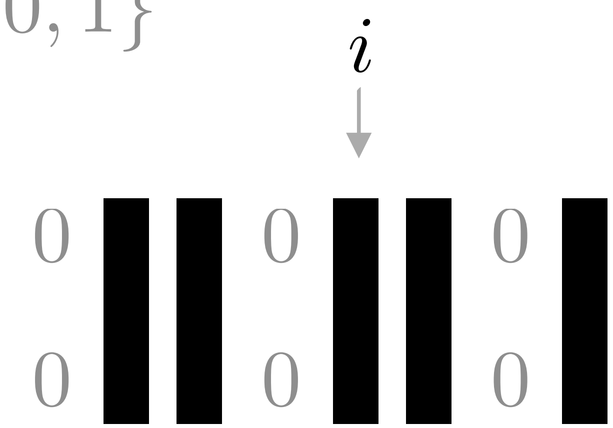
Ω_i ... set of erasure patterns that cause trouble for bit i

$$\hat{x}_i^{\mathrm{MAP}}(Y_{\sim i}) = ?$$

$$\Omega_i \subseteq \{0,1\}^{N-1}$$

$\omega \in \{0,1\}^{N-1}$

$c \in \mathcal{C} \subset \{0,1\}^N$



$$\omega \in \Omega_i \text{ iff } \exists c \in \mathcal{C} : c_i = 1 \wedge c_{\sim i} \prec \omega$$

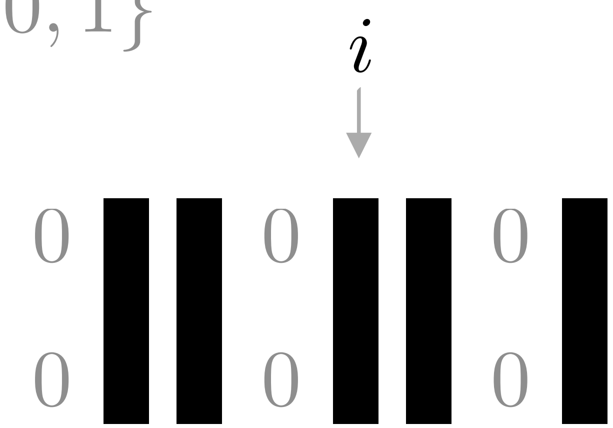
Ω_i ... set of erasure patterns that cause trouble for bit i

$$\hat{x}_i^{\mathrm{MAP}}(Y_{\sim i}) = ?$$

$$\Omega_i \subseteq \{0,1\}^{N-1}$$

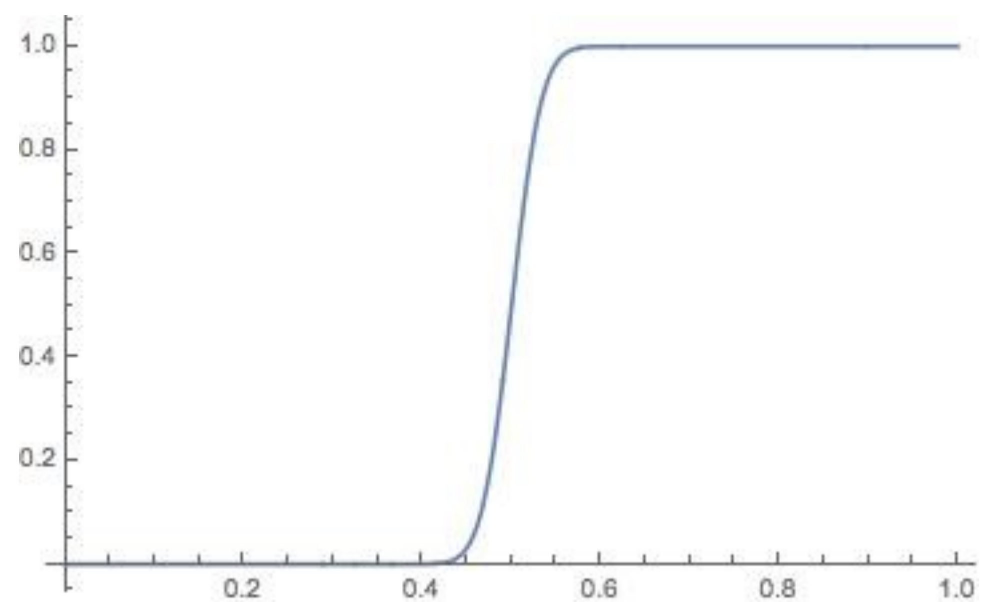
$\omega \in \{0,1\}^{N-1}$

$c \in \mathcal{C} \subset \{0,1\}^N$



$$\omega \in \Omega_i \ \text{ iff } \ \exists c \in \mathcal{C} : c_i = 1 \wedge c_{\sim i} \prec \omega$$

$$h_i(\epsilon) = \mu_\epsilon(\Omega_i) = P\{\hat{x}_i^{\mathrm{MAP}}(Y_{\sim i}) = ?\}$$



Monotonicity: Ω_i is monotone

Symmetry: if $\mathcal{C}$ is 2-transitive then Ω_i is symmetric

Independence: if $\mathcal{C}$ is transitive then $h_i(\epsilon)$ does not depend on i

Monotonicity: Ω_i is monotone

$$
\begin{array}{lcccccccc}
 & & & & & i & & & \\
 & & & & & \downarrow & & & \\
\omega \in \{0,1\}^{N-1} & 0 & 1 & 1 & 0 & & 1 & 0 & 1 \\
c \in \mathcal{C} \subset \{0,1\}^{N} & 0 & 1 & 1 & 0 & \mathbf{1} & 0 & 0 & 1
\end{array}
$$

$$\omega \in \Omega_i \text{ iff } \exists c \in \mathcal{C} : c_i = 1 \wedge c_{\sim i} \prec \omega$$

Monotonicity: Ω_i is monotone

$$\begin{array}{lcccccccccc} & & & & & i & & & & \\ & & & & & \downarrow & & & & \\ \omega \in \{0,1\}^{N-1} & 0 & 1 & 1 & 0 & & 1 & 1 & 1 & \omega' \\ c \in \mathcal{C} \subset \{0,1\}^N & 0 & 1 & 1 & 0 & \mathbf{1} & 0 & 0 & 1 & \end{array}$$

$$\omega \in \Omega_i \text{ iff } \exists c \in \mathcal{C} : c_i = 1 \wedge c_{\sim i} \prec \omega \prec \omega'$$

Monotonicity: Ω_i is monotone

$$
\begin{array}{lcccccccccc}
 & & & & & i & & & & \\
 & & & & & \downarrow & & & & \\
\omega \in \{0,1\}^{N-1} & 0 & 1 & 1 & 0 & & 1 & 1 & 1 & \omega' \\
c \in \mathcal{C} \subset \{0,1\}^{N} & 0 & 1 & 1 & 0 & \mathbf{1} & 0 & 0 & 1 &
\end{array}
$$

$$\omega \in \Omega_i \ \text{ iff } \ \exists c \in \mathcal{C} : c_i = 1 \wedge c_{\sim i} \prec \omega \prec \omega' \Rightarrow \omega' \in \Omega_i$$

Symmetry: if $\mathcal{C}$ is 2-transitive then Ω_i is symmetric

$$
\begin{array}{lcccccccccl}
 & & & & & i & & & & \\
 & & & & & \downarrow & & & & \\
\omega \in \{0,1\}^{N-1} & 0 & 1 & 1 & 0 & & 1 & 0 & 1 & \quad \omega \in \Omega_i \\
c \in \mathcal{C} \subset \{0,1\}^N & 0 & 1 & 1 & 0 & \mathbf{1} & 0 & 0 & 1 &
\end{array}
$$

Symmetry: if $\mathcal{C}$ is 2-transitive then Ω_i is symmetric

$$\begin{array}{lccccccccl}
 & & j & & & i & & & k & \\
 & & \downarrow & & & \downarrow & & & \downarrow & \\
\omega \in \{0,1\}^{N-1} & 0 & 1 & 1 & 0 & & 1 & 0 & 1 & \omega \in \Omega_i \\
c \in \mathcal{C} \subset \{0,1\}^N & 0 & 1 & 1 & 0 & \mathbf{1} & 0 & 0 & 1 &
\end{array}$$

$$\pi : [N] \to [N] \qquad \pi(i) = i \qquad \pi(j) = k$$

$$\pi(\mathcal{C}) = \mathcal{C}$$

$$\hat{\pi} : [N] \setminus \{i\} \to [N] \setminus \{i\} \qquad \hat{\pi}(j) = k$$

Symmetry: if $\mathcal{C}$ is 2-transitive then Ω_i is symmetric

$$
\begin{array}{lcccccccc}
 & & j & & & i & & & k \\
 & & \downarrow & & & \downarrow & & & \downarrow \\
\omega \in \{0,1\}^{N-1} & 0 & 1 & 1 & 0 & & 1 & 0 & 1 \\
c \in \mathcal{C} \subset \{0,1\}^{N} & 0 & 1 & 1 & 0 & \mathbf{1} & 0 & 0 & 1
\end{array}
$$

$$\omega \in \Omega_i \Rightarrow \hat{\pi}(\omega) \in \Omega_i$$

$$\pi : [N] \to [N] \qquad \pi(i) = i \qquad \pi(j) = k$$

$$\hat{\pi} : [N] \setminus \{i\} \to [N] \setminus \{i\} \qquad \hat{\pi}(j) = k$$

$$\pi(\mathcal{C}) = \mathcal{C}$$

Symmetry: if $\mathcal{C}$ is 2-transitive then Ω_i is symmetric

$$
\begin{array}{lcccccccccl}
 & & j & & & i & & & k & \\
 & & \downarrow & & & \downarrow & & & \downarrow & \\
\omega \in \{0,1\}^{N-1} & 0 & 1 & 1 & 0 & & 1 & 0 & 1 & & \omega \in \Omega_i \Rightarrow \hat{\pi}(\omega) \in \Omega_i \\
c \in \mathcal{C} \subset \{0,1\}^N & 0 & 1 & 1 & 0 & \mathbf{1} & 0 & 0 & 1 & &
\end{array}
$$

$$\pi : [N] \to [N] \quad \pi(i) = i \quad \pi(j) = k \qquad \pi(\mathcal{C}) = \mathcal{C}$$

$$\hat{\pi} : [N] \setminus \{i\} \to [N] \setminus \{i\} \quad \hat{\pi}(j) = k$$

$$\omega \in \Omega_i \Rightarrow \exists c \in \mathcal{C} : c_i = 1 \wedge c_{\sim i} \prec \omega$$

Symmetry: if $\mathcal{C}$ is 2-transitive then Ω_i is symmetric

$$\begin{array}{lcccccccc}
 & & j & & & i & & & k \\
 & & \downarrow & & & \downarrow & & & \downarrow \\
\omega \in \{0,1\}^{N-1} & 0 & 1 & 1 & 0 & & 1 & 0 & 1 \\
c \in \mathcal{C} \subset \{0,1\}^{N} & 0 & 1 & 1 & 0 & \mathbf{1} & 0 & 0 & 1
\end{array}
\qquad \omega \in \Omega_i \Rightarrow \hat{\pi}(\omega) \in \Omega_i$$

$$\pi : [N] \to [N] \quad \pi(i) = i \quad \pi(j) = k$$

$$\pi(\mathcal{C}) = \mathcal{C}$$

$$\hat{\pi} : [N] \setminus \{i\} \to [N] \setminus \{i\} \quad \hat{\pi}(j) = k$$

$$\omega \in \Omega_i \Rightarrow \exists c \in \mathcal{C} : c_i = 1 \wedge c_{\sim i} \prec \omega$$

$$\hat{c} = \pi(c) \Rightarrow \hat{c} \in \mathcal{C} : \hat{c}_i = 1 \wedge \hat{c}_{\sim i} \prec \hat{\pi}(\omega) \Rightarrow \hat{\pi}(\omega) \in \Omega_i$$

Independence: if $\mathcal{C}$ is transitive then $h_i(\epsilon)$ does not depend on i

$$\begin{array}{lcccccccccl} & & & & & \overset{i}{\downarrow} & & & & & \\ \omega \in \{0,1\}^{N-1} & 0 & 1 & 1 & 0 & & 1 & 0 & 1 & & \omega \in \Omega_i \\ c \in \mathcal{C} \subset \{0,1\}^N & 0 & 1 & 1 & 0 & \mathbf{1} & 0 & 0 & 1 & & \end{array}$$

$$\pi : [N] \to [N] \qquad \pi(i) = j \qquad \pi(\mathcal{C}) = \mathcal{C}$$

$$c' = \pi(c) \qquad c' \in \mathcal{C} \qquad \omega' = \hat{\pi}(\omega)$$

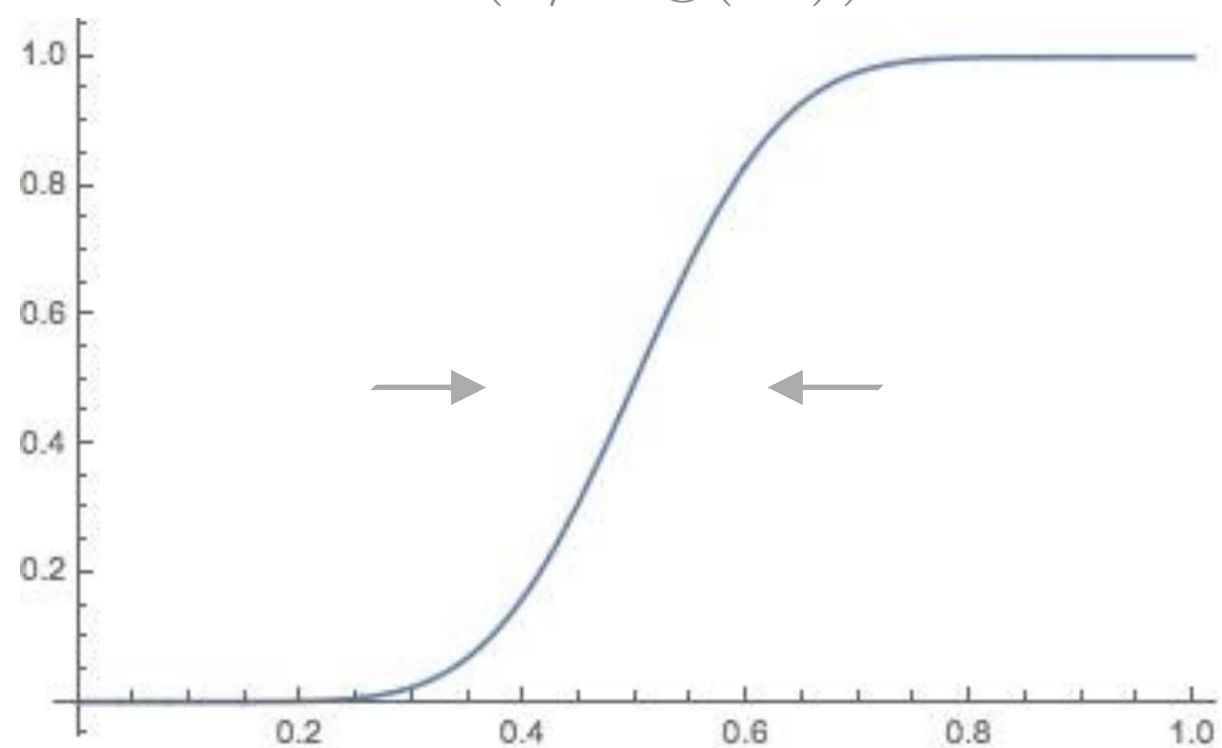
$$c'_j = 1 \wedge c'_{\sim j} \prec \omega' \Rightarrow \omega' \in \Omega_j$$

$$\Omega_j = \hat{\pi}(\Omega_i) \Rightarrow h_j(\epsilon) = h_i(\epsilon)$$

monotonicity + symmetry + Friedgut-Kalai

monotonicity + symmetry + Friedgut-Kalai

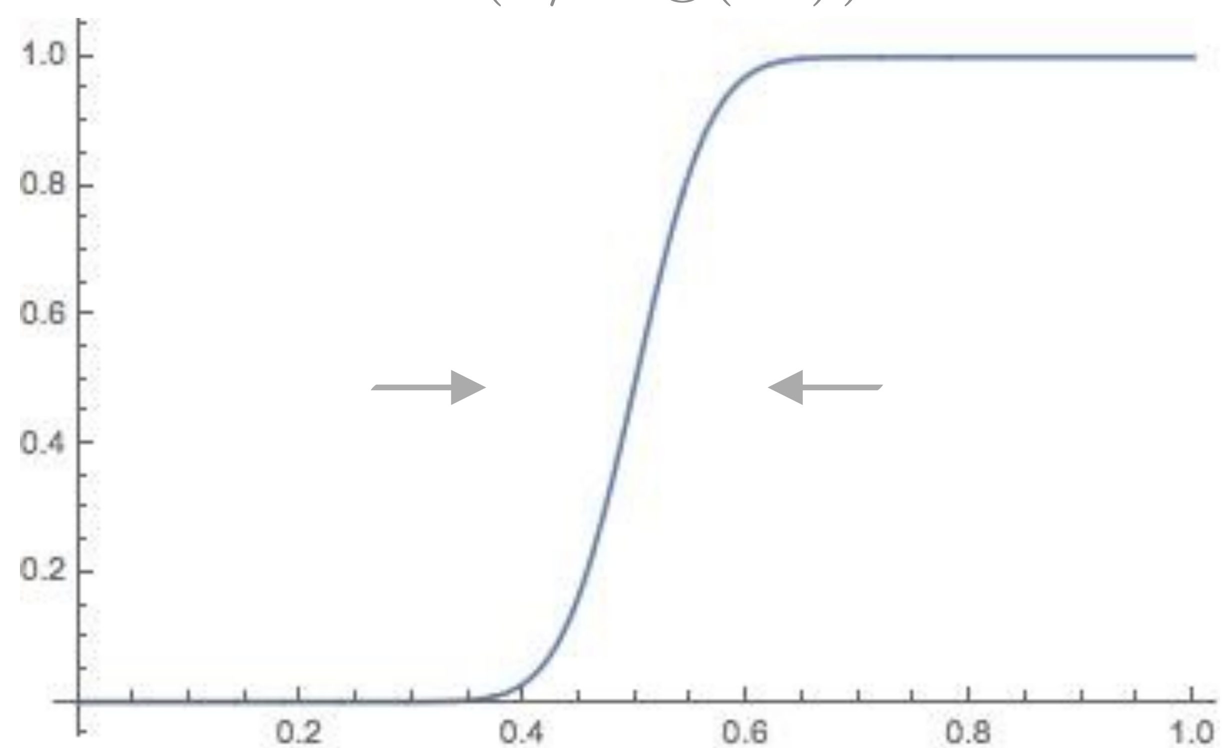
$O(1/\log(N))$



$h_i(\epsilon)$ has a sharp threshold of width $O(1/\log(N))$

monotonicity + symmetry + Friedgut-Kalai

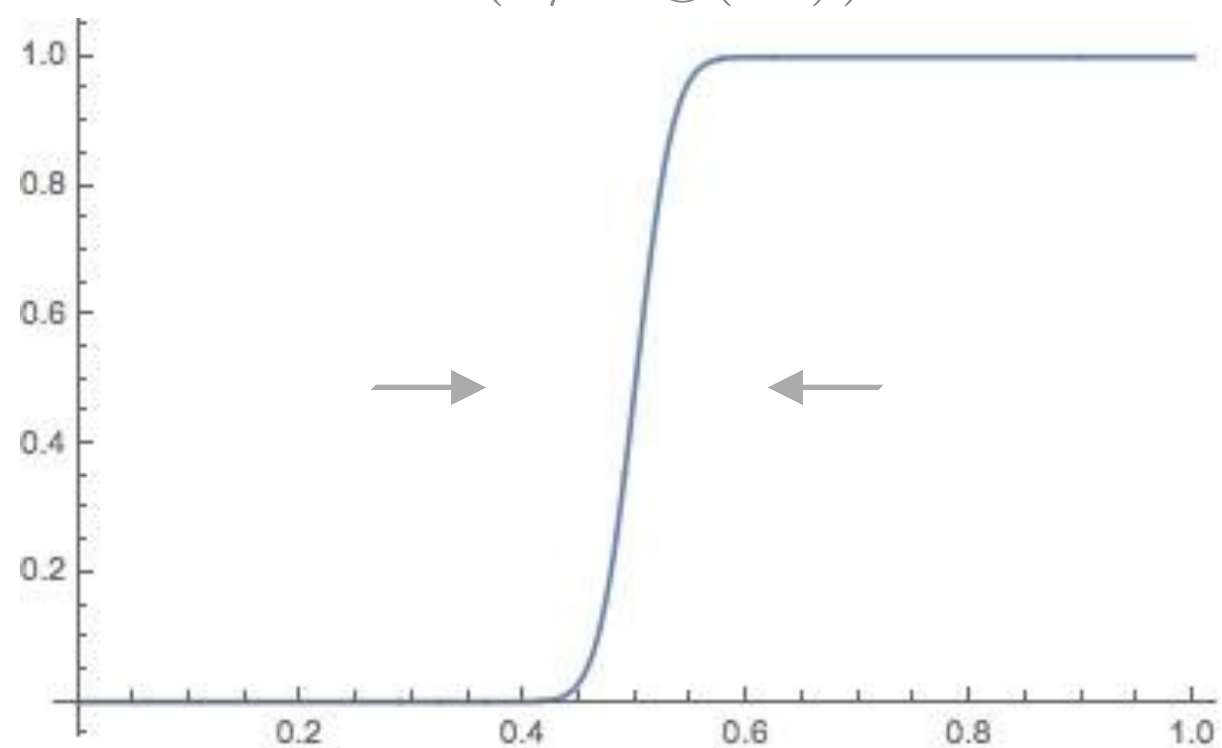
$O(1/\log(N))$



$h_i(\epsilon)$ has a sharp threshold of width $O(1/\log(N))$

monotonicity + symmetry + Friedgut-Kalai

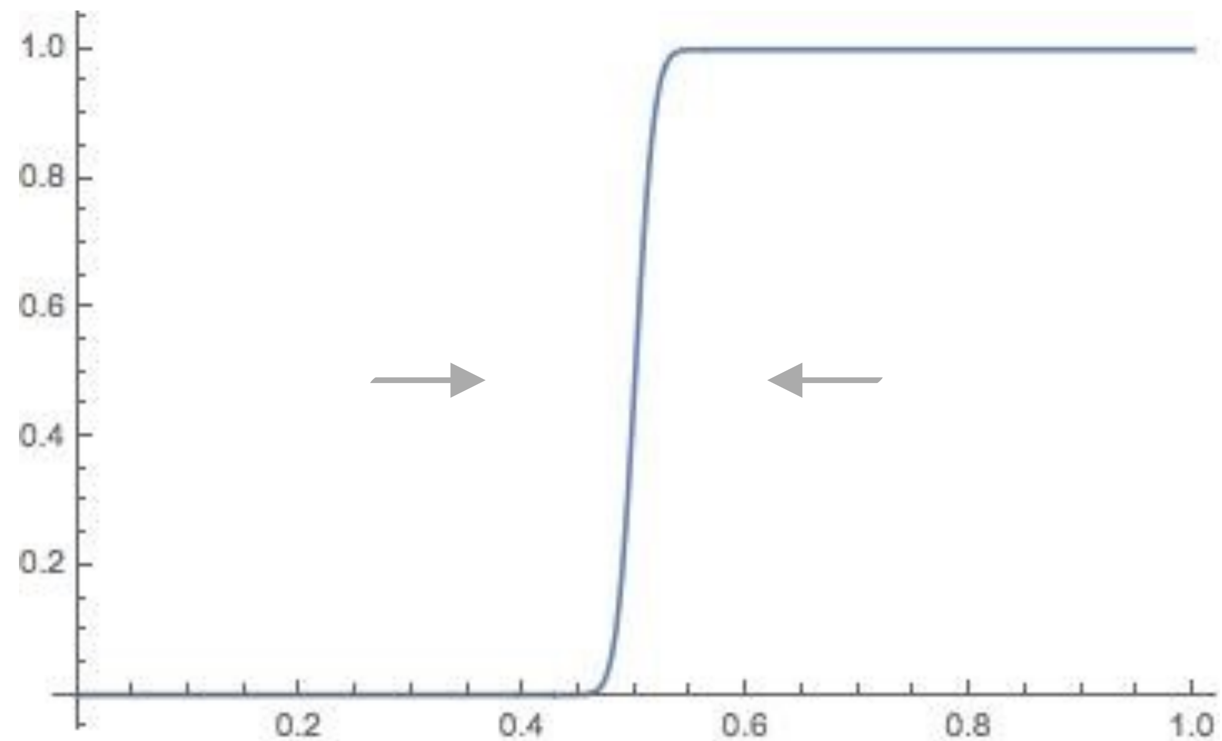
$O(1/\log(N))$



$h_i(\epsilon)$ has a sharp threshold of width $O(1/\log(N))$

$monotonicity + symmetry + Friedgut\text{-}Kalai$

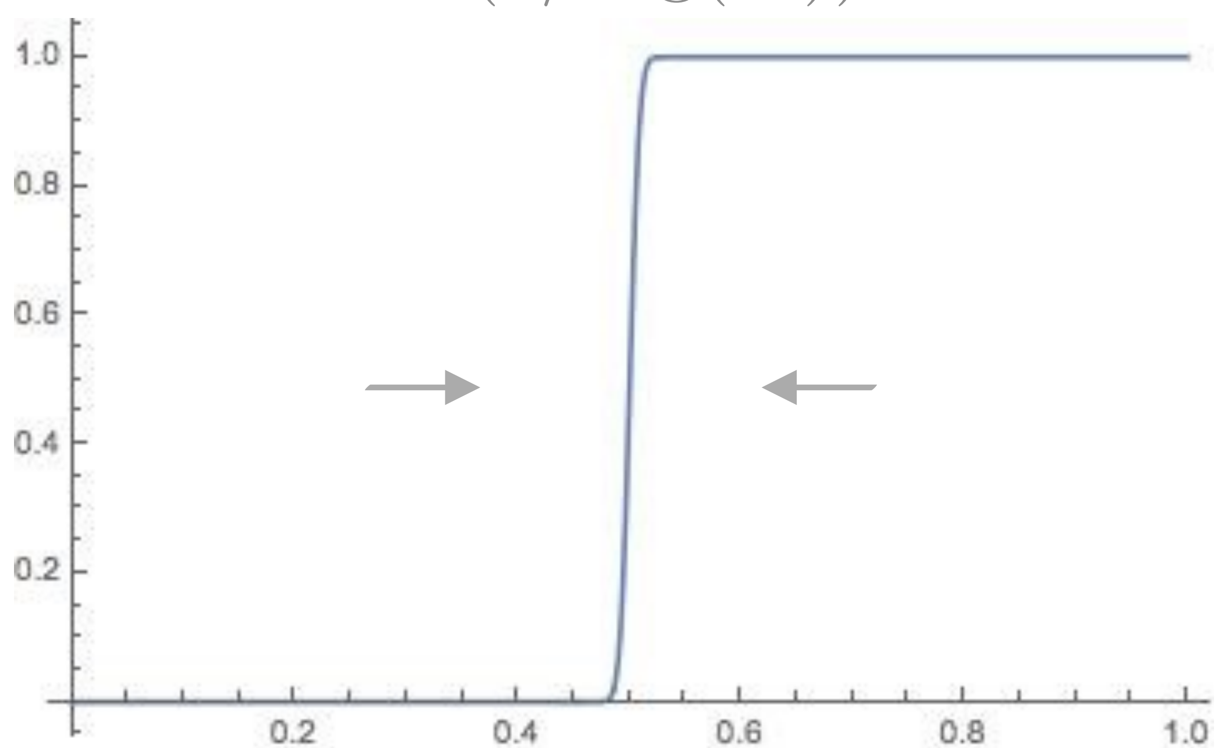
$O(1/\log(N))$



$h_i(\epsilon)$ has a sharp threshold of width $O(1/\log(N))$

monotonicity + symmetry + Friedgut-Kalai

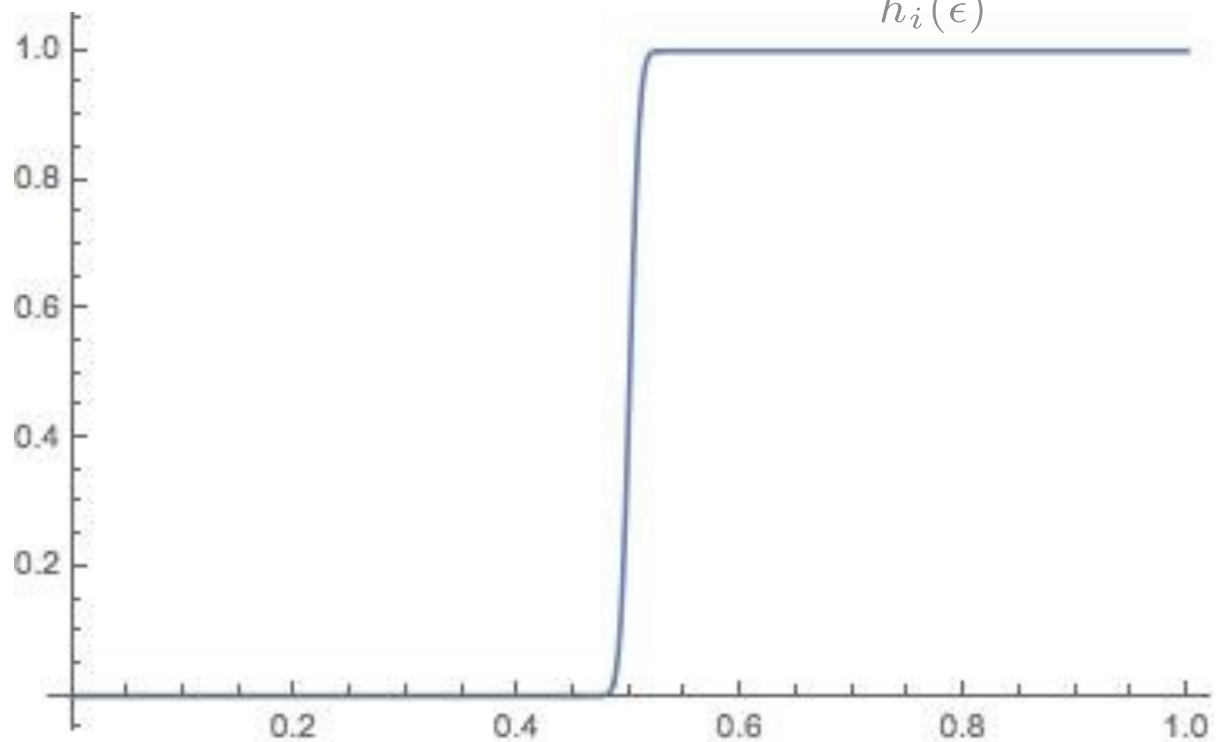
$O(1/\log(N))$



$h_i(\epsilon)$ has a sharp threshold of width $O(1/\log(N))$

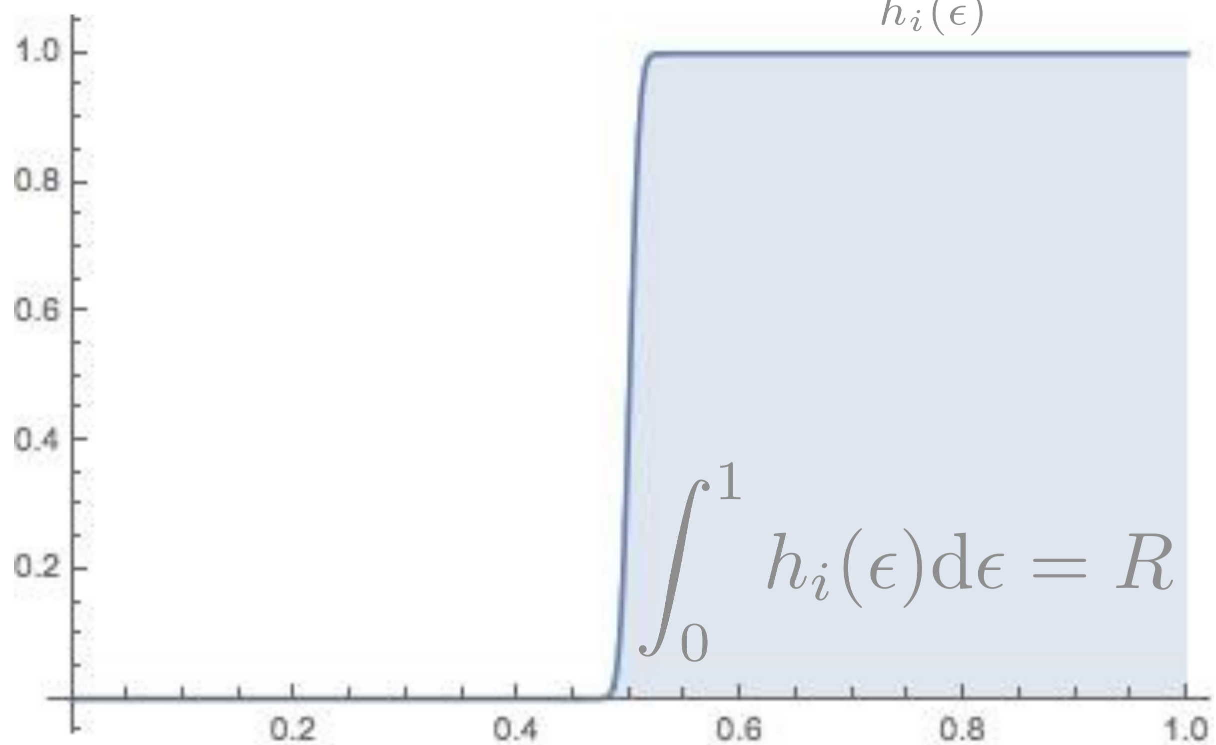
independence + Area Theorem

$$\frac{\mathrm{d}H(X \mid Y(\epsilon))}{\mathrm{d}\epsilon} = \sum_{i=1}^{N} \underbrace{P\{\hat{x}_i^{\mathrm{MAP}}(Y_{\sim i}) = ?\}}_{h_i(\epsilon)}$$



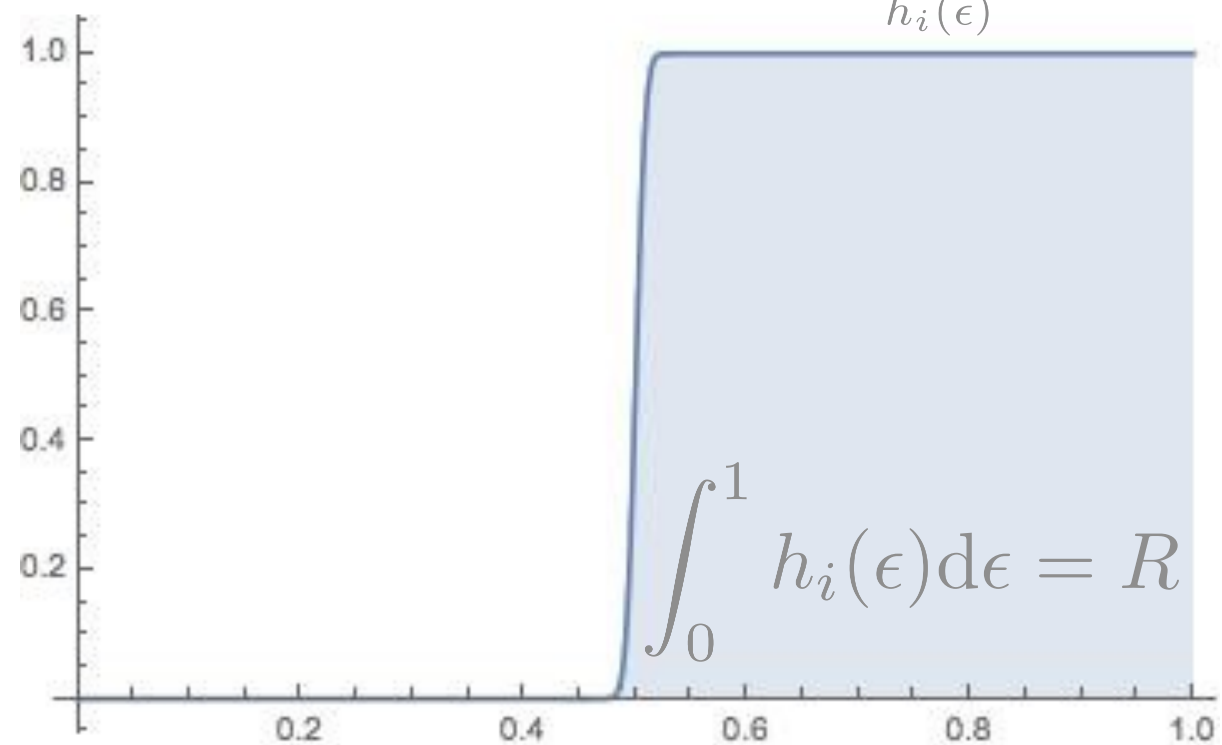
independence + Area Theorem

$$\frac{\mathrm{d}H(X \mid Y(\epsilon))}{\mathrm{d}\epsilon} = \sum_{i=1}^{N} \underbrace{P\{\hat{x}_i^{\mathrm{MAP}}(Y_{\sim i}) = ?\}}_{h_i(\epsilon)}$$



$$independence + Area\ Theorem$$

$$\frac{\mathrm{d}H(X \mid Y(\epsilon))}{\mathrm{d}\epsilon} = \sum_{i=1}^{N} \underbrace{P\{\hat{x}_i^{\mathrm{MAP}}(Y_{\sim i}) = ?\}}_{h_i(\epsilon)}$$



$\epsilon = 1 - R$

threshold

DONE!

Ingredients

- *RM codes are 2-transitive*
- *symmetric monotone sets have sharp thresholds*
- *EXIT functions satisfy the Area Theorem*

block versus bit $\cdots$

block versus bit $\cdots$ Friedgut-Kalai $\rightarrow$ Bourgain-Kalai

block versus bit	$\cdots$	Friedgut-Kalai $\to$ Bourgain-Kalai
other codes	$\cdots$	

block versus bit	$\cdots$	Friedgut-Kalai $\rightarrow$ Bourgain-Kalai
other codes	$\cdots$	BCH codes

block versus bit	$\cdots$	Friedgut-Kalai $\to$ Bourgain-Kalai
other codes	$\cdots$	BCH codes
general channels	$\cdots$	

block versus bit	$\cdots$	Friedgut-Kalai $\rightarrow$ Bourgain-Kalai
other codes	$\cdots$	BCH codes
general channels	$\cdots$	cautiously optimistic

Summary

polar codes
spatially coupled
back to classics — symmetry!

finite length