Spatially Coupled LDPC Codes: Is This What Shannon Had In Mind?

Daniel J. Costello, Jr.

Dept. of Electrical Engineering,
University of Notre Dame

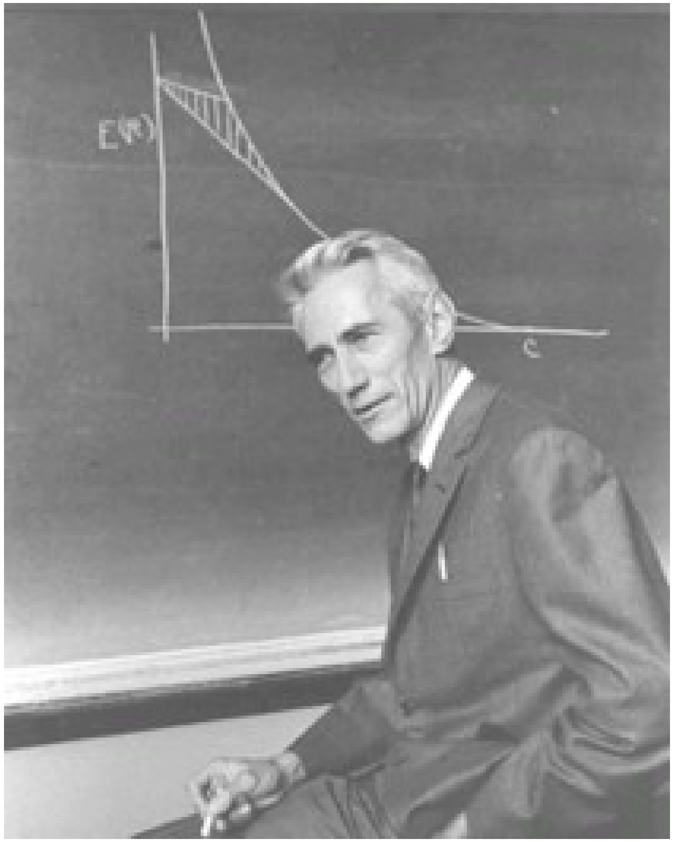
University of Michigan, Nov. 17, 2016

Research Collaborators: David Mitchell, Michael Lentmaier, and Ali Pusane

Outline

- **From Shannon to Modern Coding Theory**
  - ➡ Channel capacity, structured codes, random codes, LDPC codes
- **LDPC Block Codes**
  - ➡ Parity-check matrix and Tanner graph representations, minimum distance bounds, iterative decoding thresholds, protograph-based constructions
- **Spatially Coupled LDPC Codes**
  - ➡ Protograph representation, edge-spreading construction, termination
  - ➡ Iterative decoding thresholds, threshold saturation, minimum distance
- **Practical Considerations**
  - ➡ Window decoding, performance, latency, and complexity comparisons to LDPC block codes, rate-compatibility, implementation aspects

Shannon's Legacy



Claude Elwood Shannon
Apr. 30, 1916 – Feb. 24, 2001
Father of **Information Theory**

Shannon's Legacy

UNIVERSITY OF NOTRE DAME

Shannon's Theory Was Invented at Bell Labs

Bell Labs in Murray Hill,
New Jersey

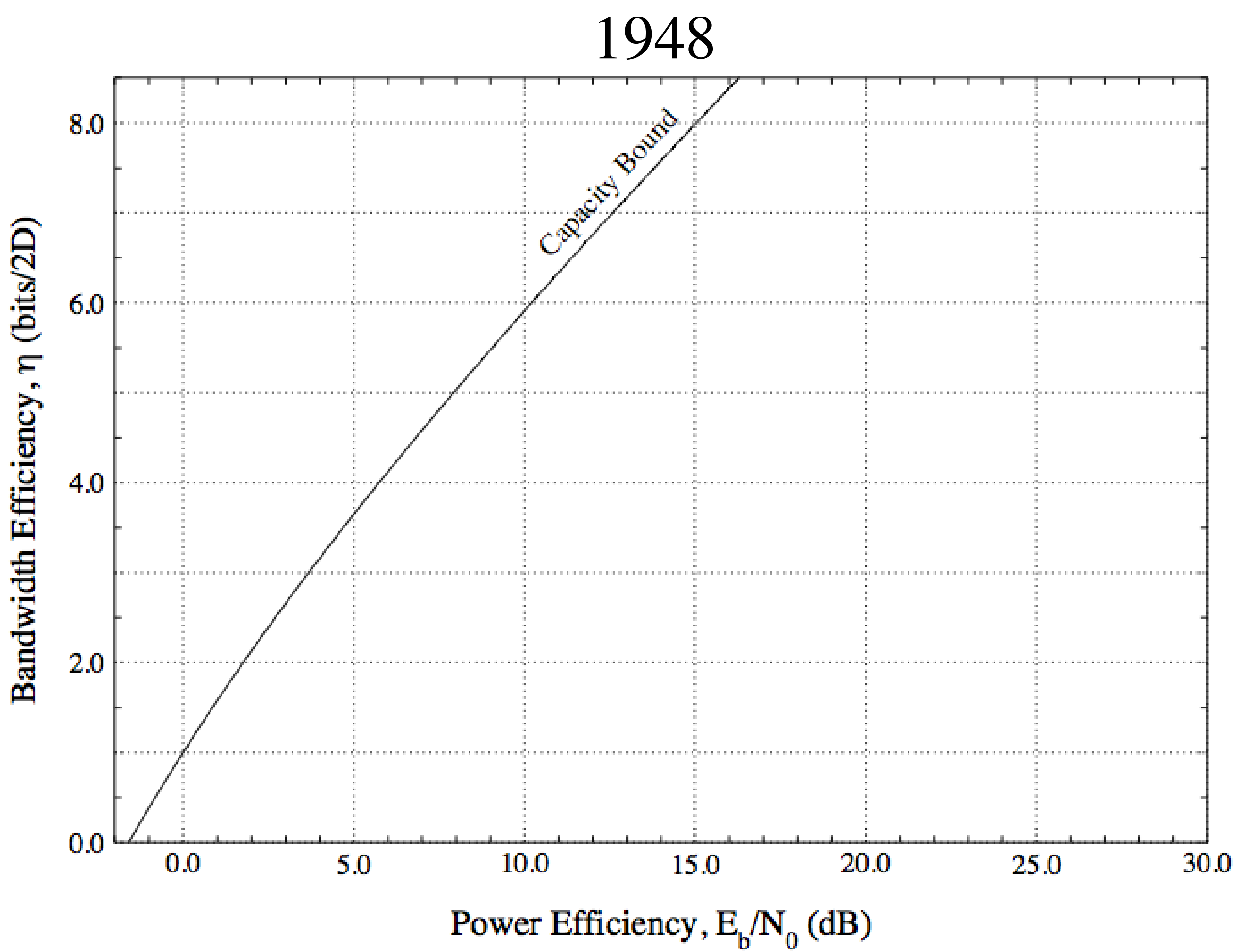
Shannon's Legacy

UNIVERSITY OF NOTRE DAME

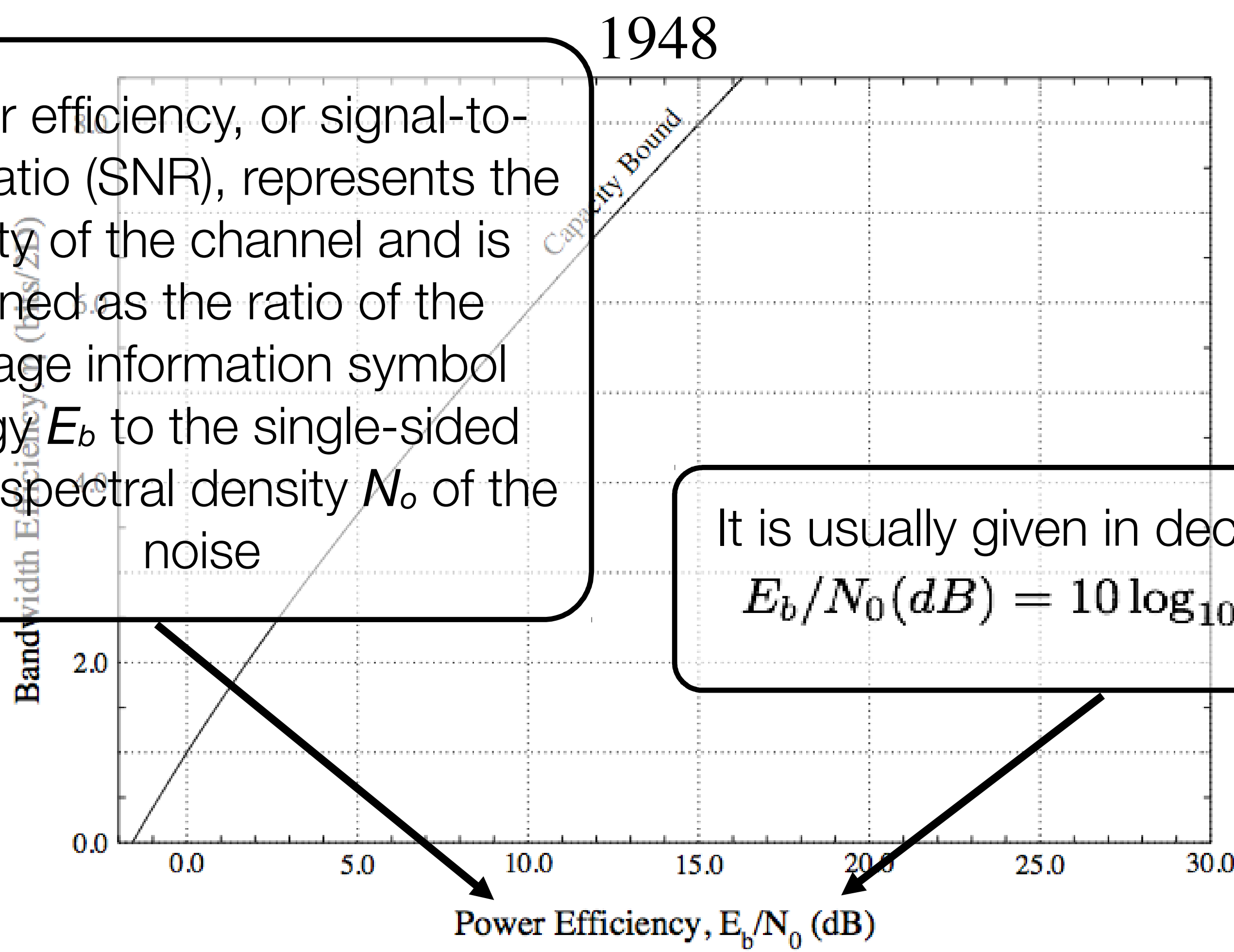
Three Great Successes of Information Theory

- Source Coding for Data Compression
- Secret Coding (Cryptography) for Data Security
- Channel Coding for Data Reliability (*the focus of this presentation*)

Shannon's Legacy



Shannon's Legacy

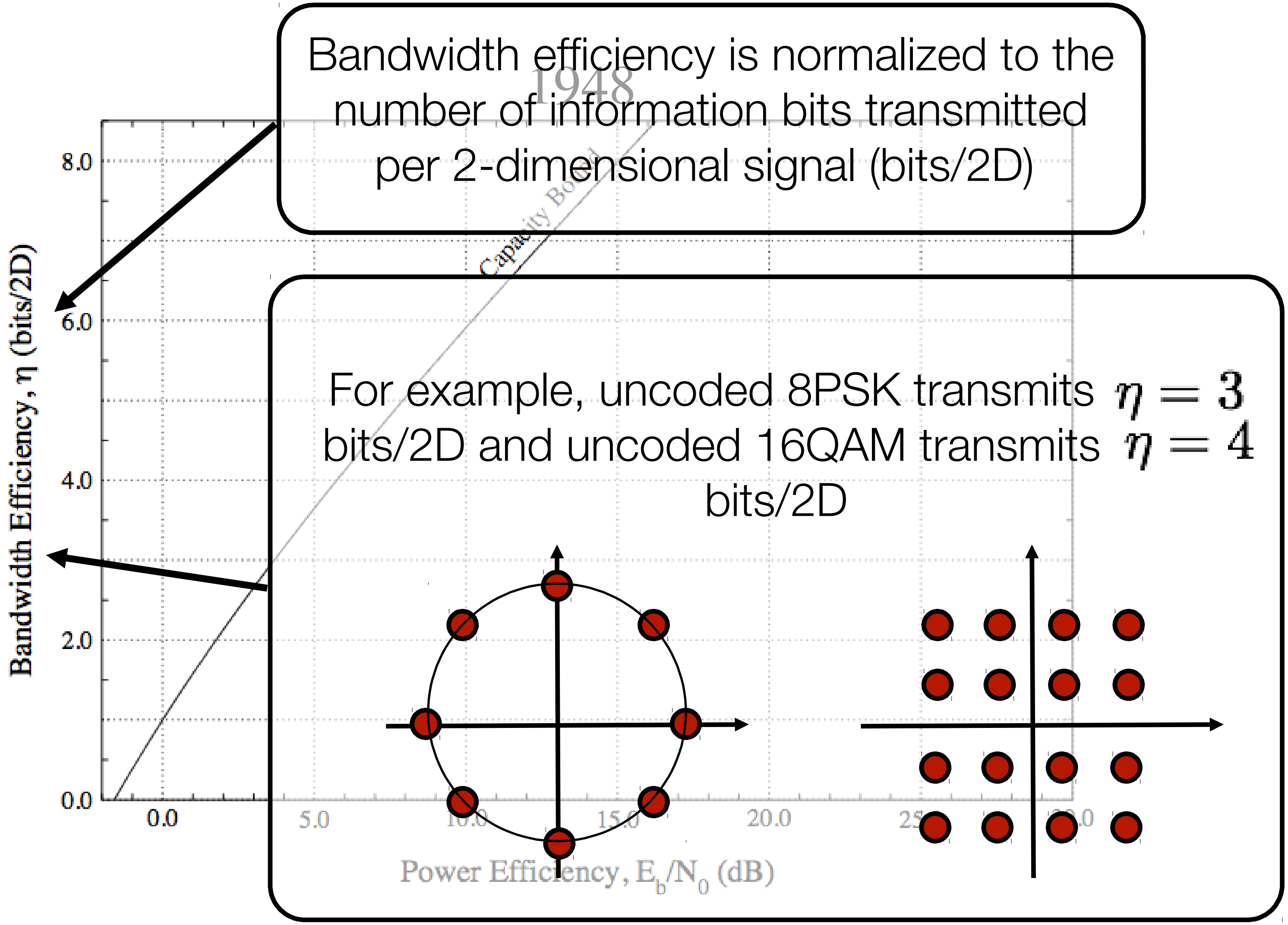


Power efficiency, or signal-to-noise ratio (SNR), represents the quality of the channel and is defined as the ratio of the average information symbol energy E_b to the single-sided power spectral density N_o of the noise

It is usually given in decibels (dB).

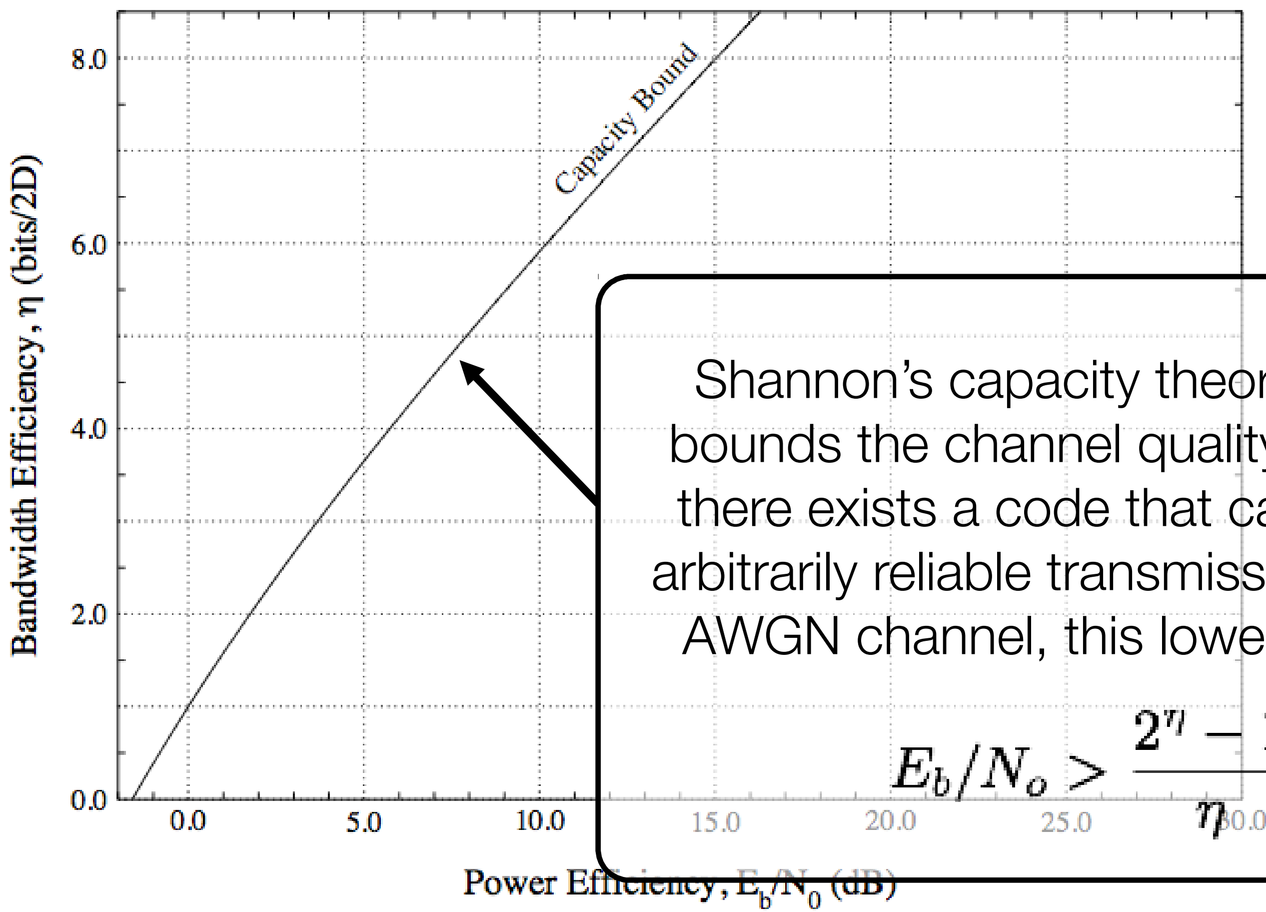
$$E_b/N_0(dB) = 10\log_{10} E_b/N_o$$

Shannon's Legacy



Shannon's Legacy

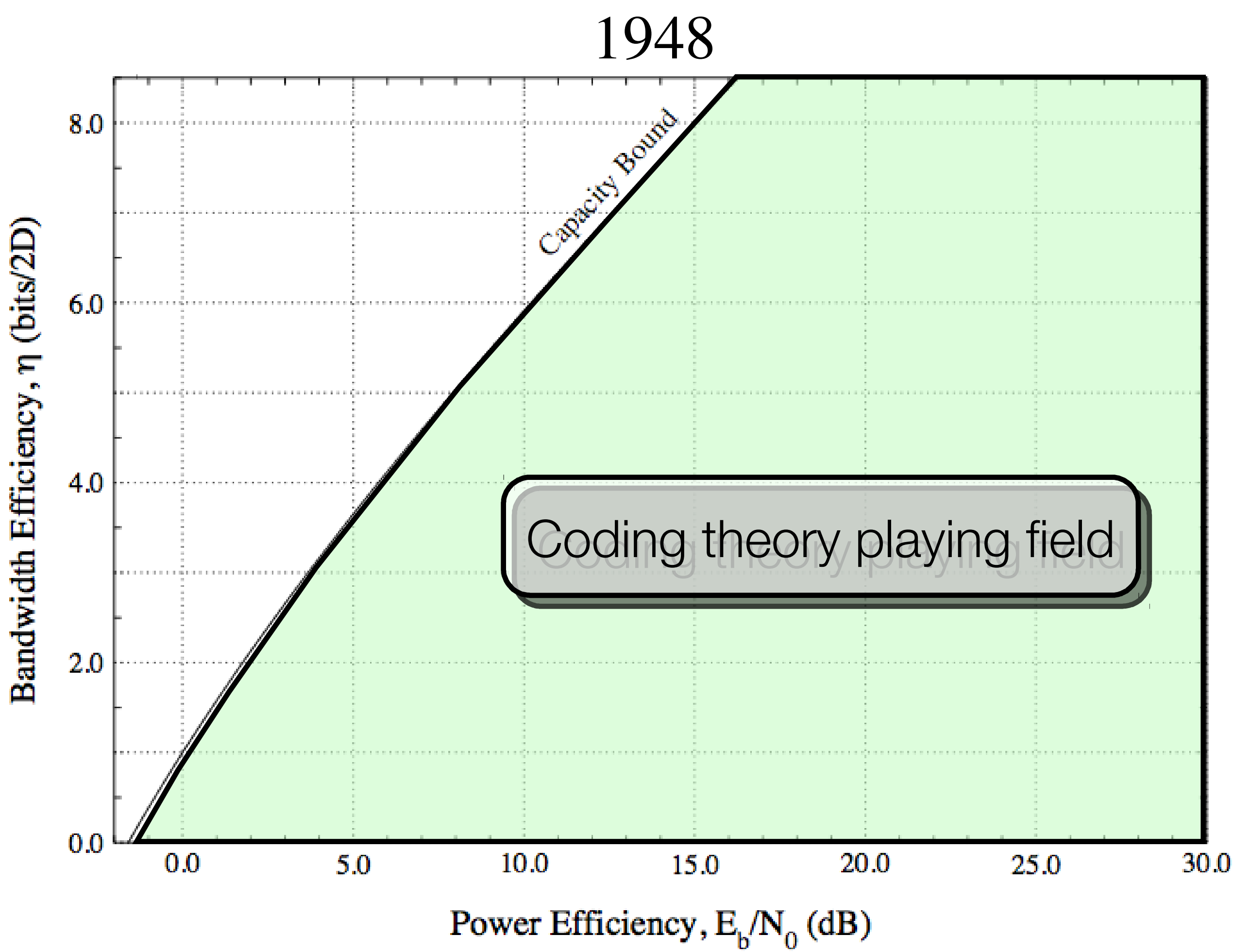
1948



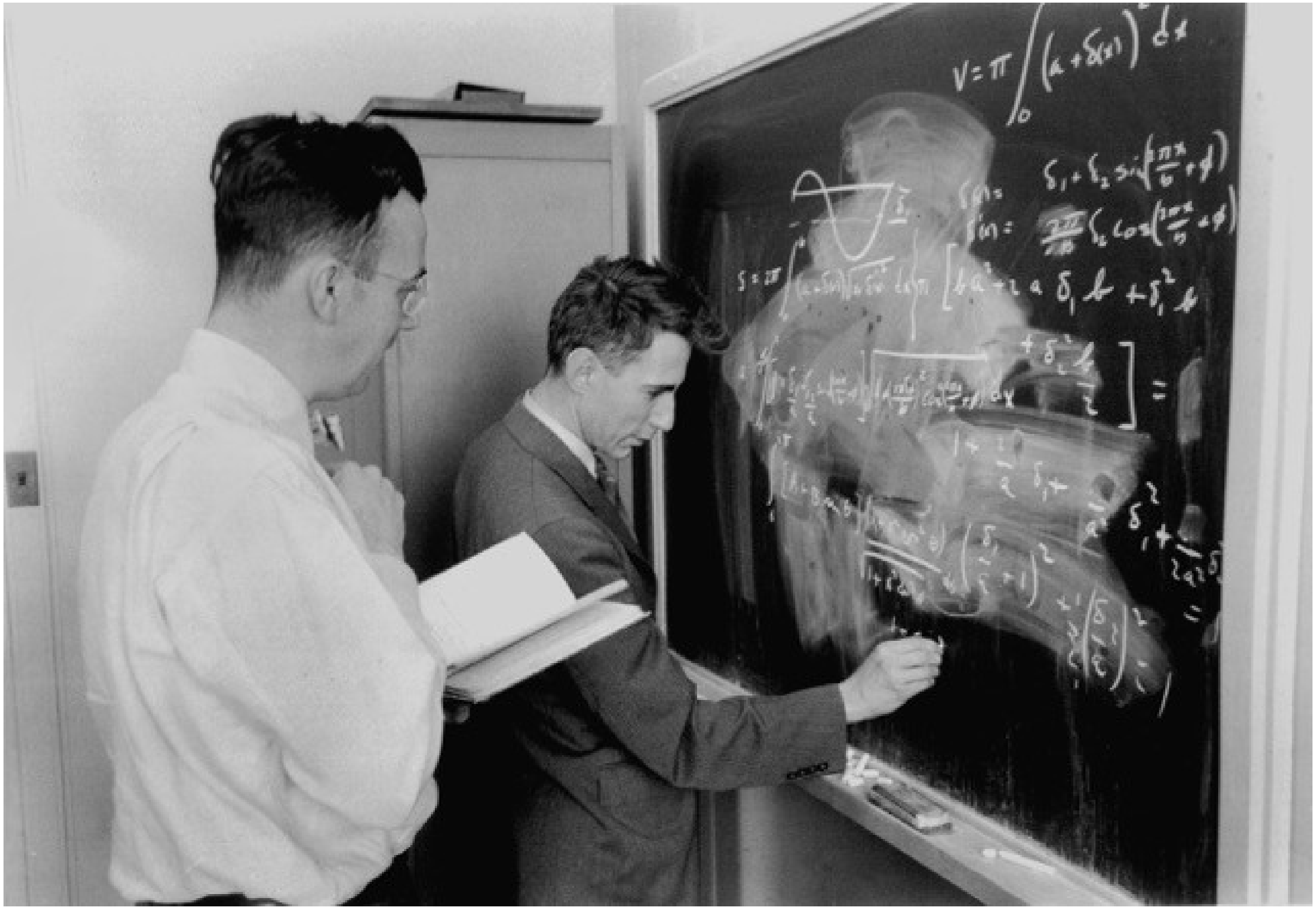
Shannon's capacity theorem lower bounds the channel quality for which there exists a code that can achieve arbitrarily reliable transmission. For the AWGN channel, this lower bound is

$$E_b/N_o > \frac{2^\eta - 1}{\eta}$$

Shannon's Legacy

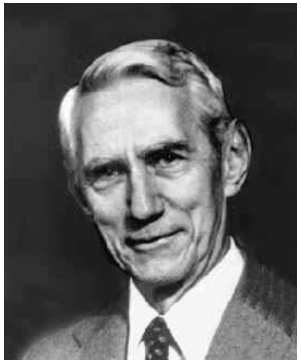


Shannon's Legacy



The coding dilemma

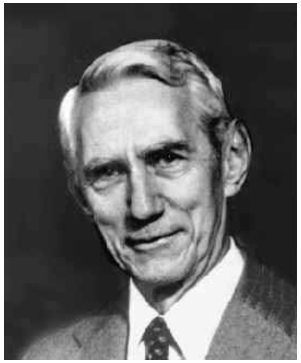
- Shannon showed that **random codes** with large block length can **achieve capacity**, but...



The coding dilemma

- Shannon showed that **random codes** with large block length can **achieve capacity**, but...

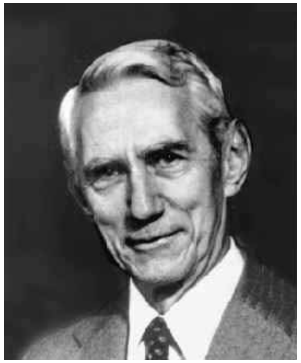
  … code **structure** (algebraic/topological) is required in order to permit decoding with reasonable complexity.



The coding dilemma

- Shannon showed that **random codes** with large block length can **achieve capacity**, but...

  … code **structure** (algebraic/topological) is required in order to permit decoding with reasonable complexity.

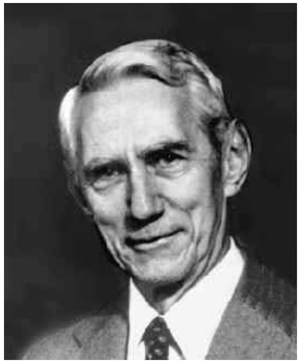


“Almost all codes are good...

The coding dilemma

- Shannon showed that **random codes** with large block length can **achieve capacity**, but...

  … code **structure** (algebraic/topological) is required in order to permit decoding with reasonable complexity.

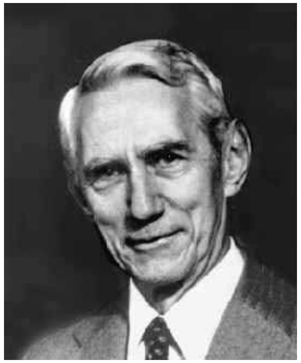


"Almost all codes are good... except those we can think of."

The coding dilemma

- Shannon showed that **random codes** with large block length can **achieve capacity**, but...

  … code **structure** (algebraic/topological) is required in order to permit decoding with reasonable complexity.

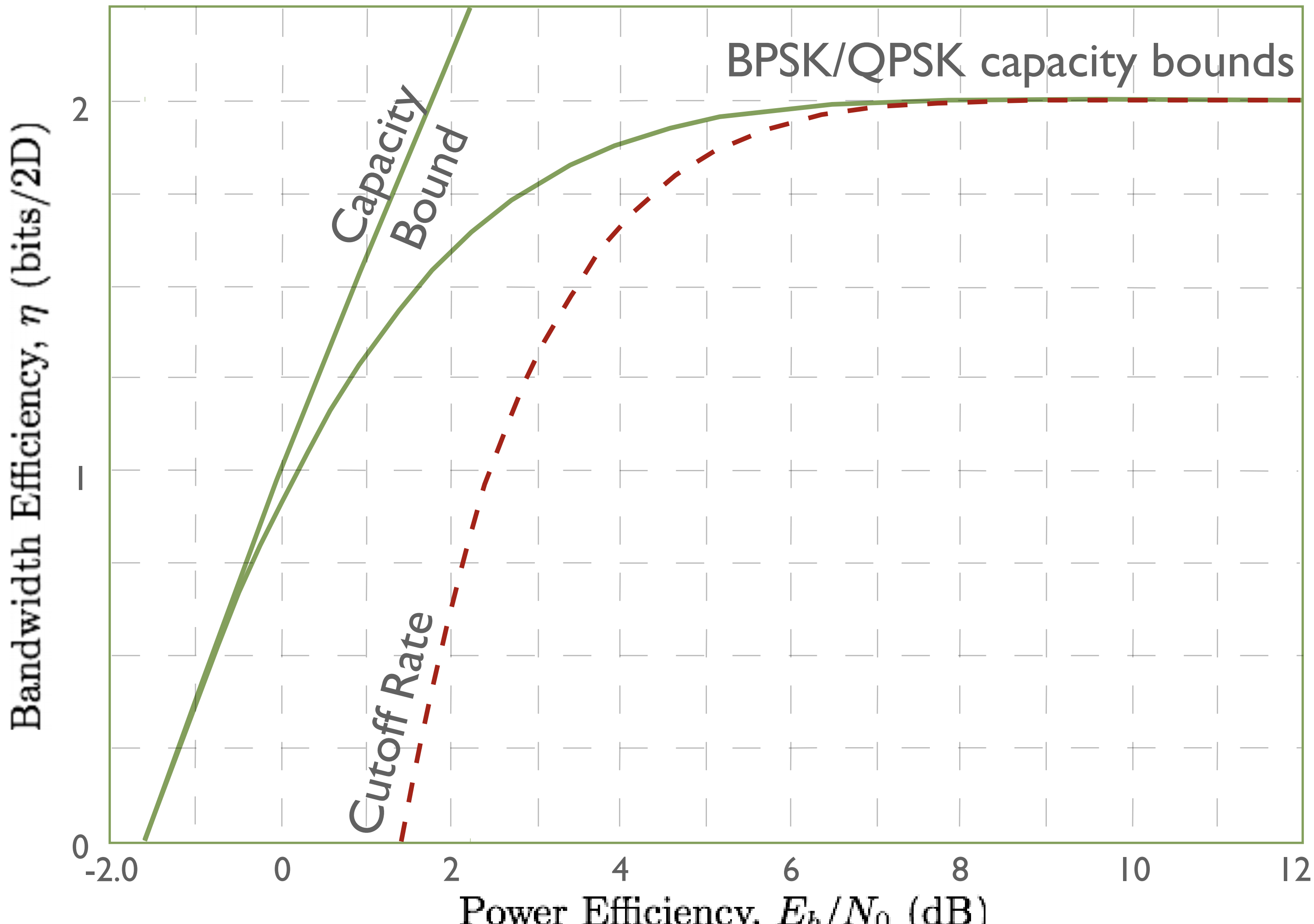


"Almost all codes are good... except those we can think of."

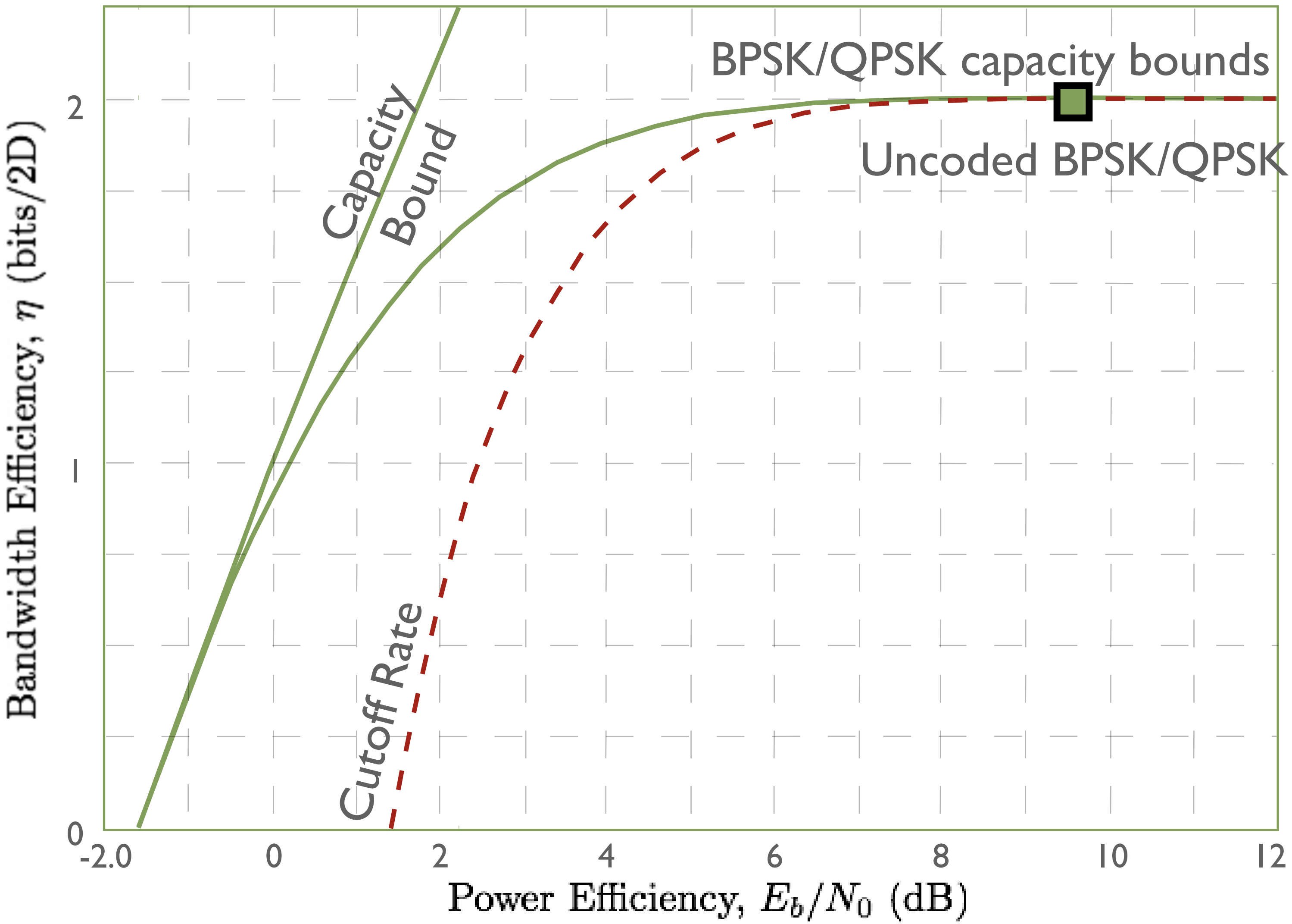
Solution: Construct random-like codes with just enough structure to allow efficient decoding

➡ Modern Coding Theory

LDPC Codes: motivation (for a target BER 10^{-5})

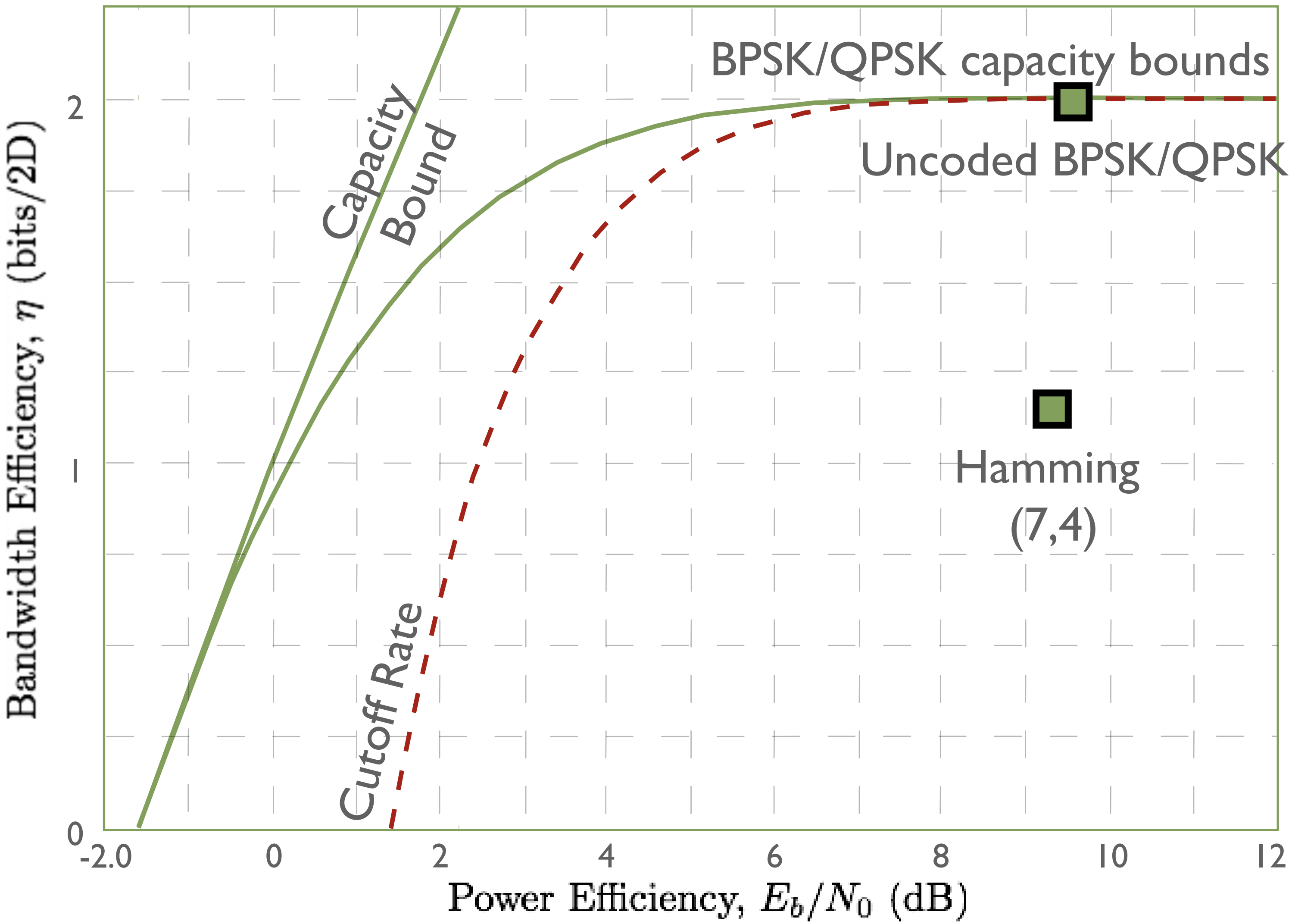


LDPC Codes: motivation (for a target BER 10^{-5})

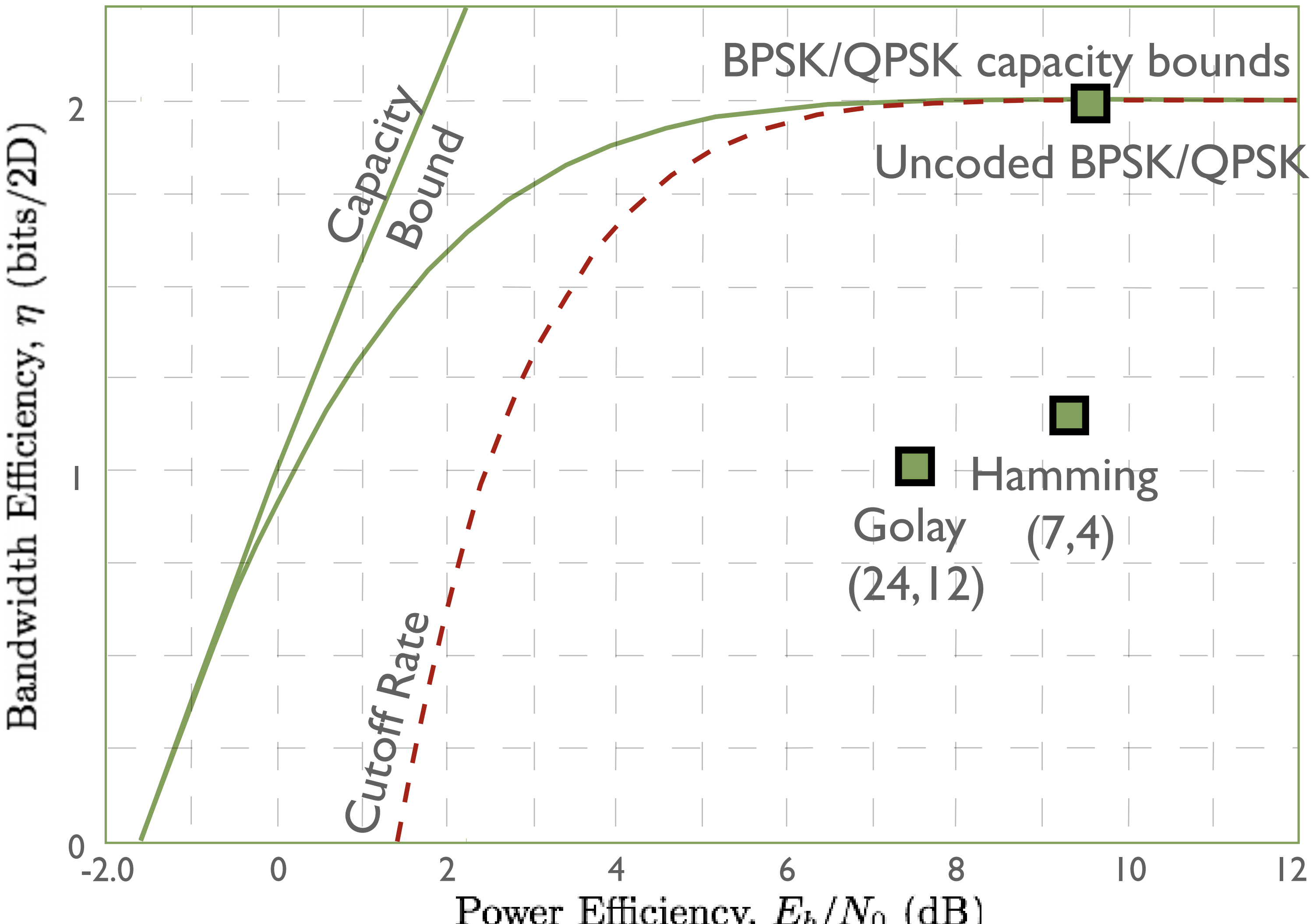


LDPC Codes: motivation (for a target BER 10^{-5})

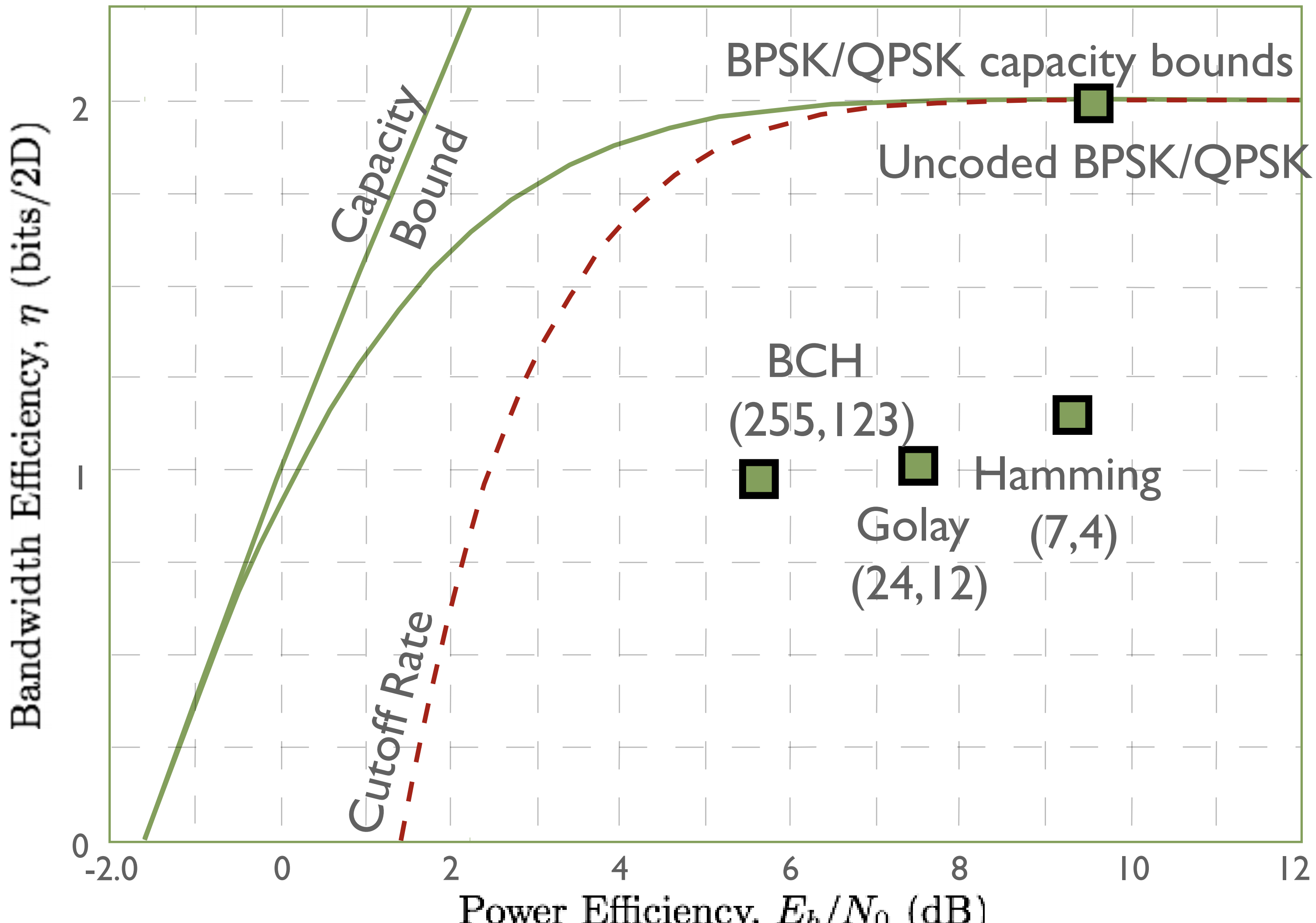
UNIVERSITY OF NOTRE DAME



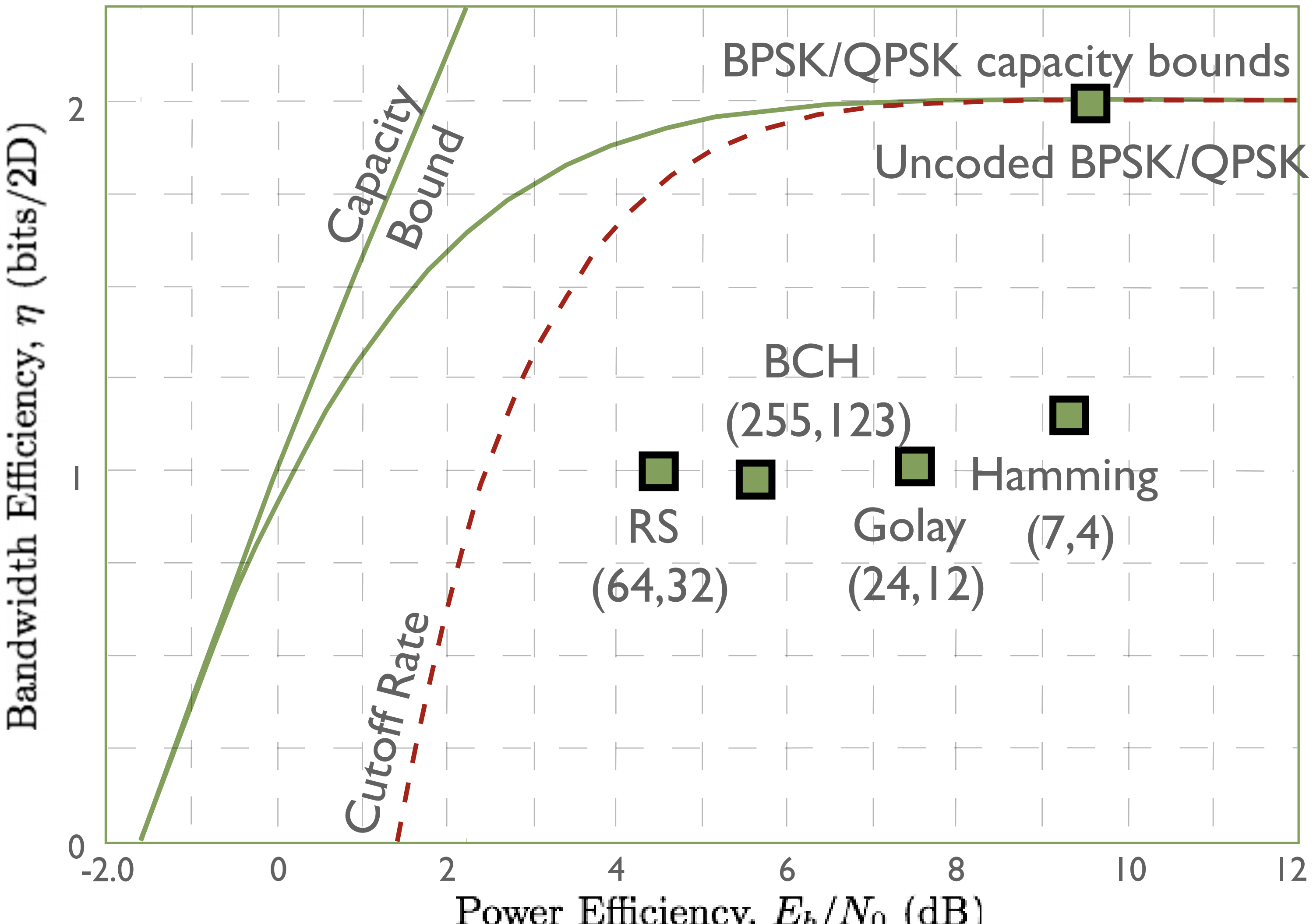
LDPC Codes: motivation (for a target BER 10^{-5})



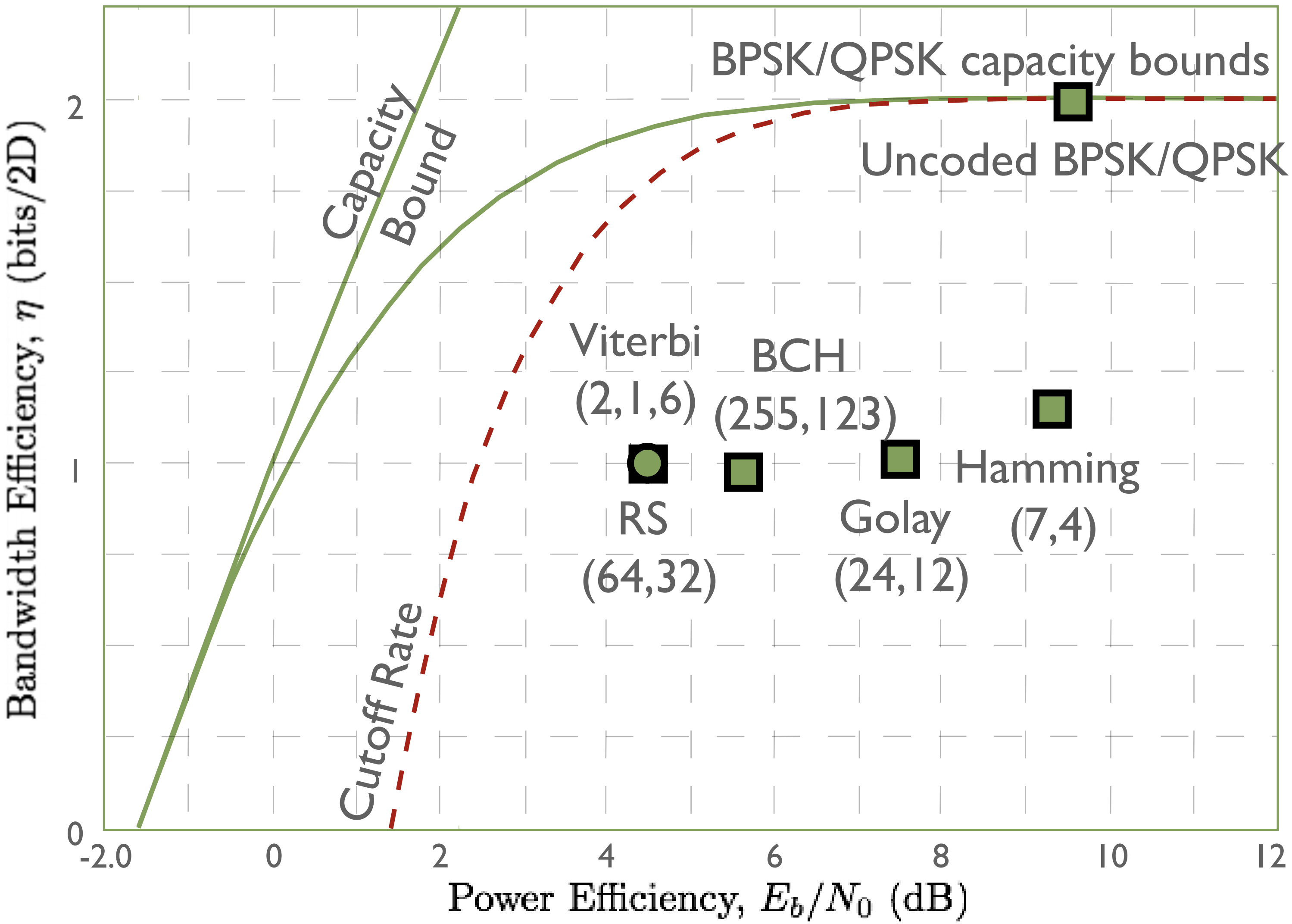
LDPC Codes: motivation (for a target BER 10^{-5})



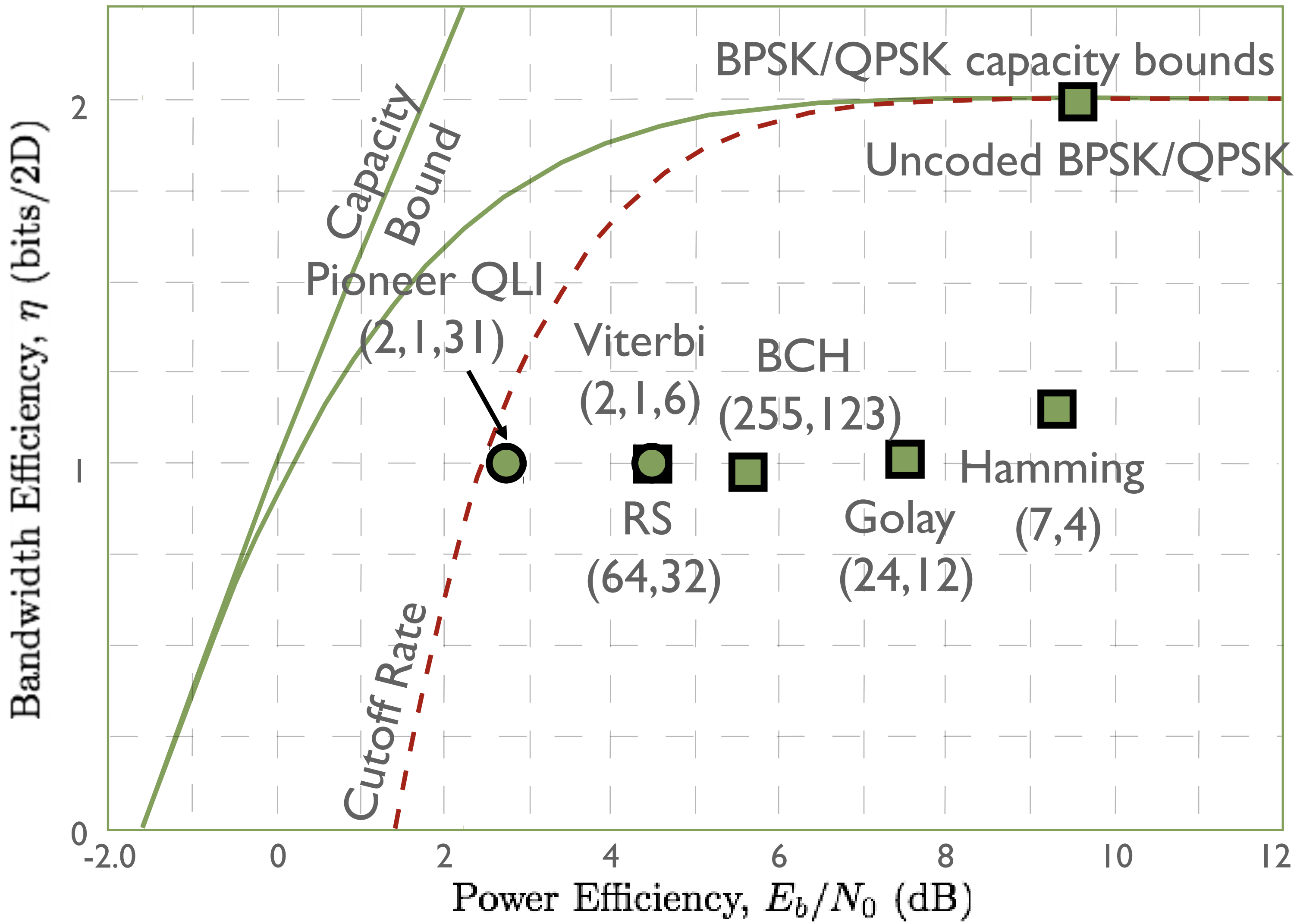
LDPC Codes: motivation (for a target BER 10^{-5})



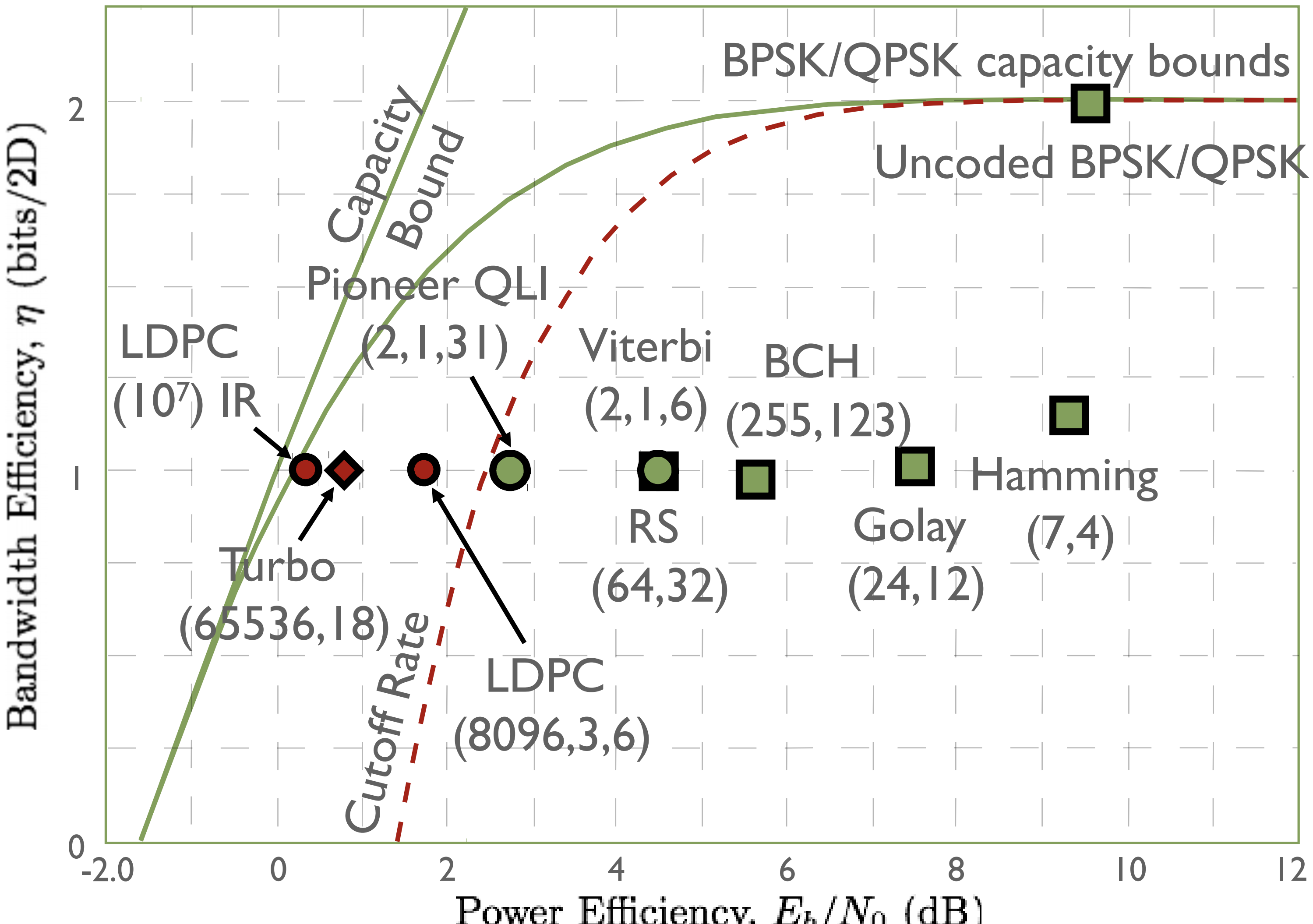
LDPC Codes: motivation (for a target BER 10^{-5})



LDPC Codes: motivation (for a target BER 10^{-5})



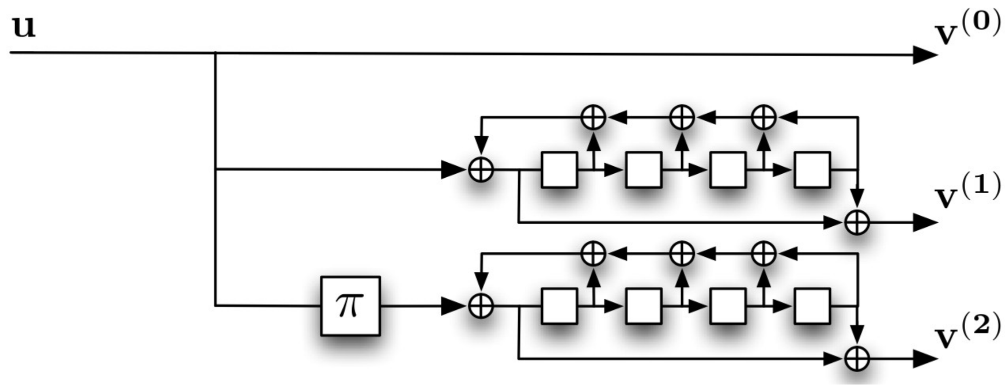
LDPC Codes: motivation (for a target BER 10^{-5})



Random-like codes (2000s - today)

- Turbo codes use a long pseudorandom interleaver

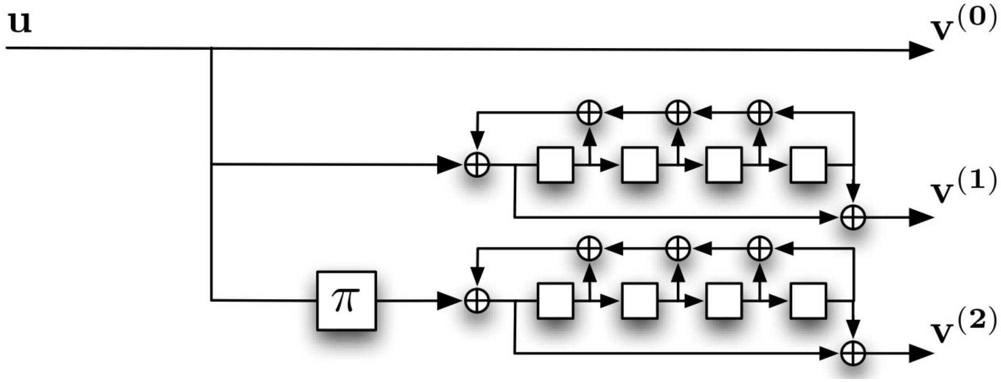
➡ 3G and 4G telephony standards HSPA, EV-DO, LTE, satellite DVB-RCS, Mars Reconnaissance Rover, WiMAX, and so on.



Random-like codes (2000s - today)

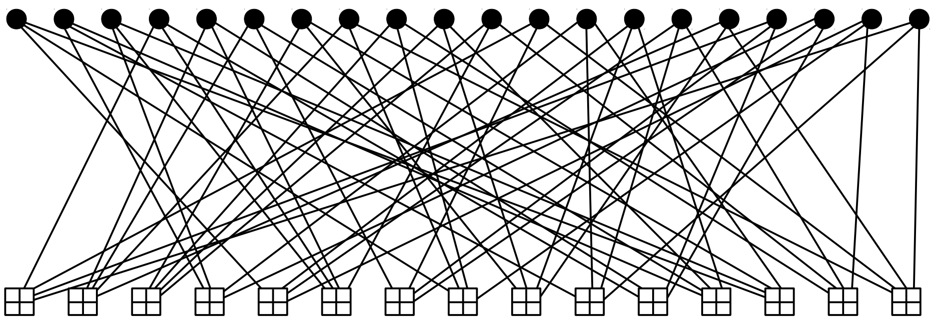
- Turbo codes use a long pseudorandom interleaver

➡ 3G and 4G telephony standards HSPA, EV-DO, LTE, satellite DVB-RCS, Mars Reconnaissance Rover, WiMAX, and so on.



- Low-density parity-check (LDPC) codes are defined on a large sparse graph

➡ DVB-S2, ITU-T G.hn standard (data networking over power lines, phone lines, and coaxial cables), 10GBase-T Ethernet, Wi-Fi standards 802.11, and so on.



Outline

- From Shannon to Modern Coding Theory
  - Channel capacity, structured codes, random codes, LDPC codes
- **LDPC Block Codes**
  - Parity-check matrix and Tanner graph representations, minimum distance bounds, iterative decoding thresholds, protograph-based constructions
- Spatially Coupled LDPC Codes
  - Protograph representation, edge-spreading construction, termination
  - Iterative decoding thresholds, threshold saturation, minimum distance
- Practical Considerations
  - Window decoding, performance, latency, and complexity comparisons to LDPC block codes, rate-compatibility, implementation aspects

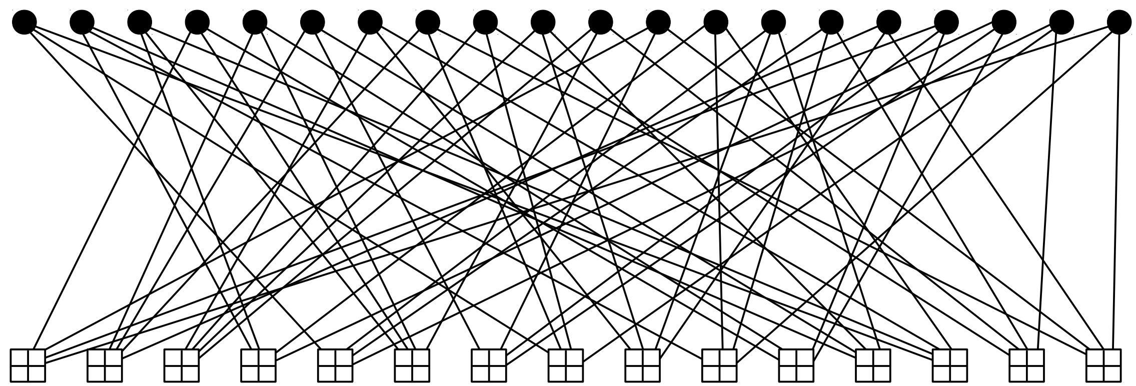
LDPC Block Codes

Definition by parity-check matrix: [Gallager, '62]

Bipartite graph representation: [Tanner, '81]

$$
\mathbf{H} = \left[\begin{array}{cccccccccccccccccccc}
0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&1\\
0&0&0&0&1&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&1&0&1&1&1&0&0&0&0&0&0&0&0&0\\
0&1&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0\\
1&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&1&0\\
\hline
0&0&1&1&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&1&1&0&0\\
1&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&1&0&0&1&0&0&0&1&0&1&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&1\\
\hline
0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0\\
0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0\\
1&0&1&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&1&0&0&0&1&0&0&0&1&0\\
0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1
\end{array}\right]_{15 \times 20}
$$

$n = 20$ **variable nodes** of degree $J = 3$



$l = 15$ **check nodes** of degree $K = 4$

Code: $\{\mathbf{v} \mid \mathbf{v}\mathbf{H}^{\mathrm{T}} = \mathbf{0}\}$

(J,K)-regular LDPC block code: $R \geq 1 - \dfrac{J}{K}$

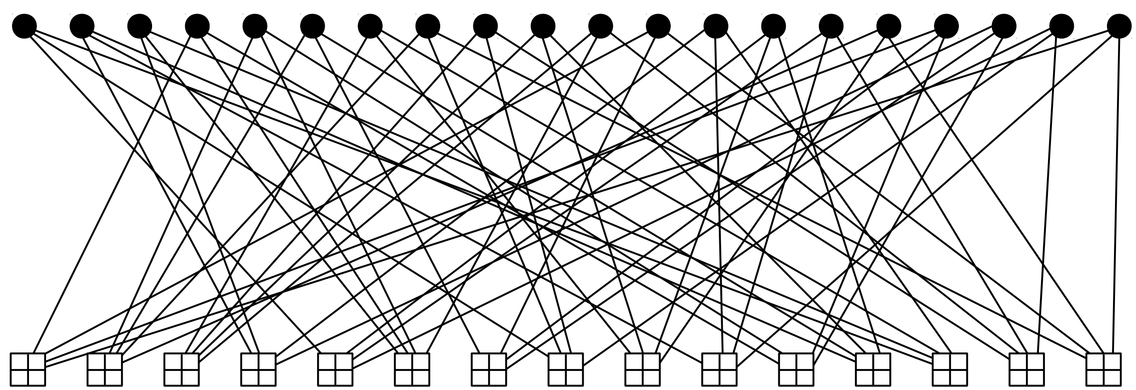
LDPC Block Codes

Definition by parity-check matrix: [Gallager, '62]

Bipartite graph representation: [Tanner, '81]

$$
\boldsymbol{H} = \left[\begin{array}{cccccccccccccccccccc}
0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&1\\
0&0&0&0&1&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&1&0&1&1&1&0&0&0&0&0&0&0&0&0\\
0&1&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0\\
1&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&1&0\\
\hline
0&0&1&1&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&1&1&0&0\\
1&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&1&0&0&1&0&0&0&1&0&1&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&1\\
\hline
0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0\\
0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0\\
1&0&1&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&1&0&0&0&1&0&0&0&1&0\\
0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1
\end{array}\right]_{15 \times 20}
$$

$n = 20$ **variable nodes** of degree $J = 3$



$l = 15$ **check nodes** of degree $K = 4$

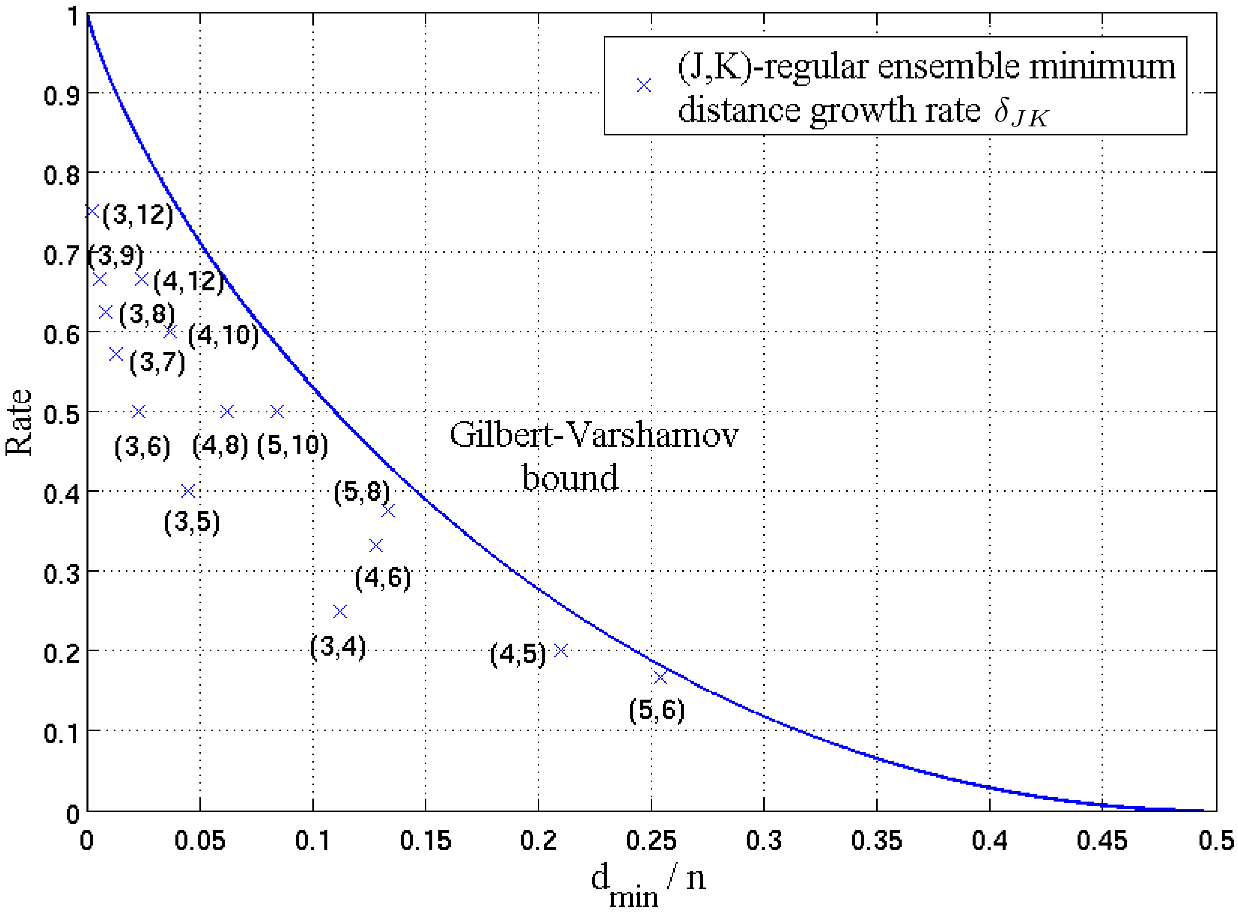
Code: $\{\mathbf{v} \mid \mathbf{v}\mathbf{H}^{\mathrm{T}} = \mathbf{0}\}$

(J,K)-regular LDPC block code: $R \geq 1 - \frac{J}{K}$

- Graph-based codes can be decoded **iteratively** with **low complexity** by exchanging messages in the graph using **Belief Propagation (BP).**

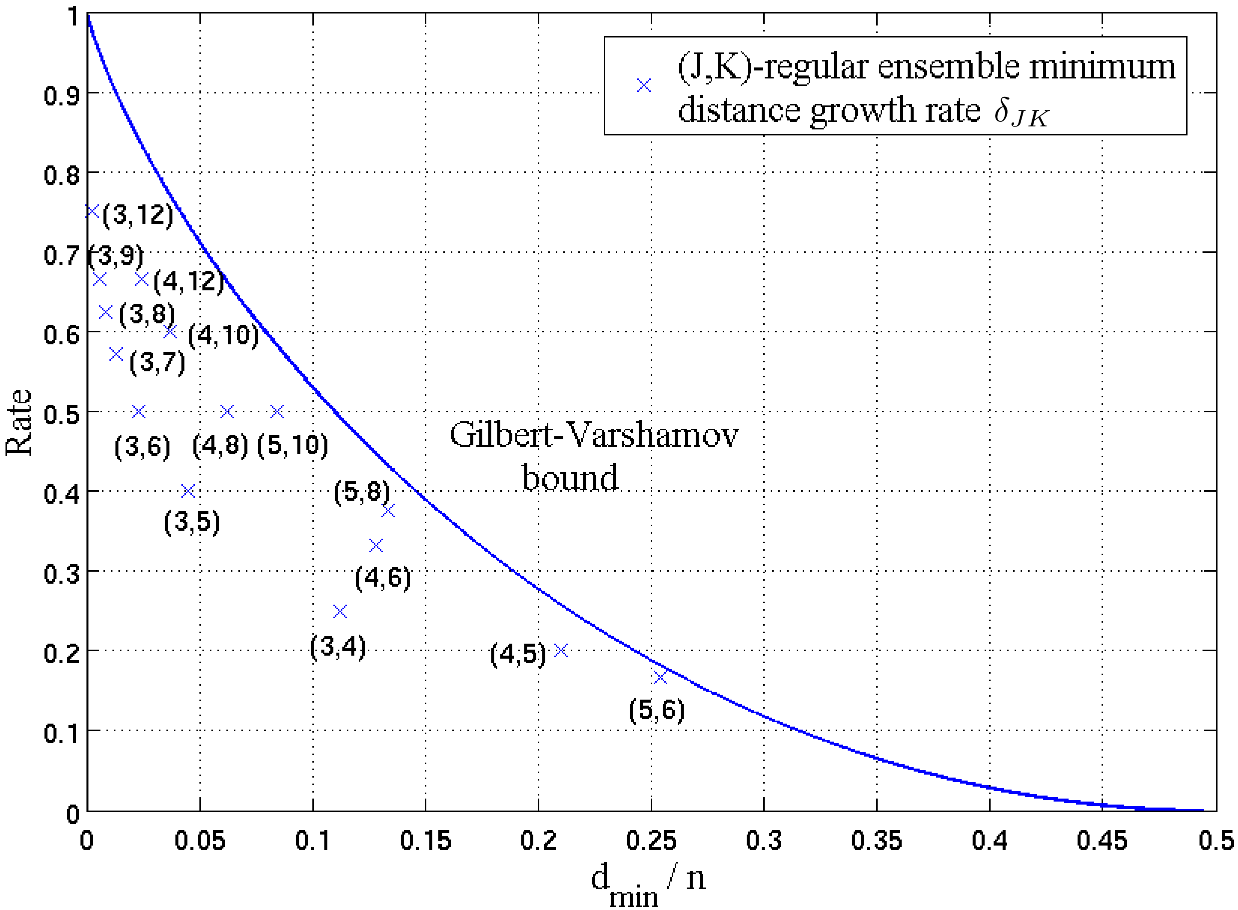
Minimum Distance Growth Rates of (J,K)-Regular LDPC Block Code Ensembles

- For an **asymptotically good** code ensemble, the minimum distance grows linearly with the block length n



Minimum Distance Growth Rates of (J,K)-Regular LDPC Block Code Ensembles

- For an **asymptotically good** code ensemble, the minimum distance grows linearly with the block length n



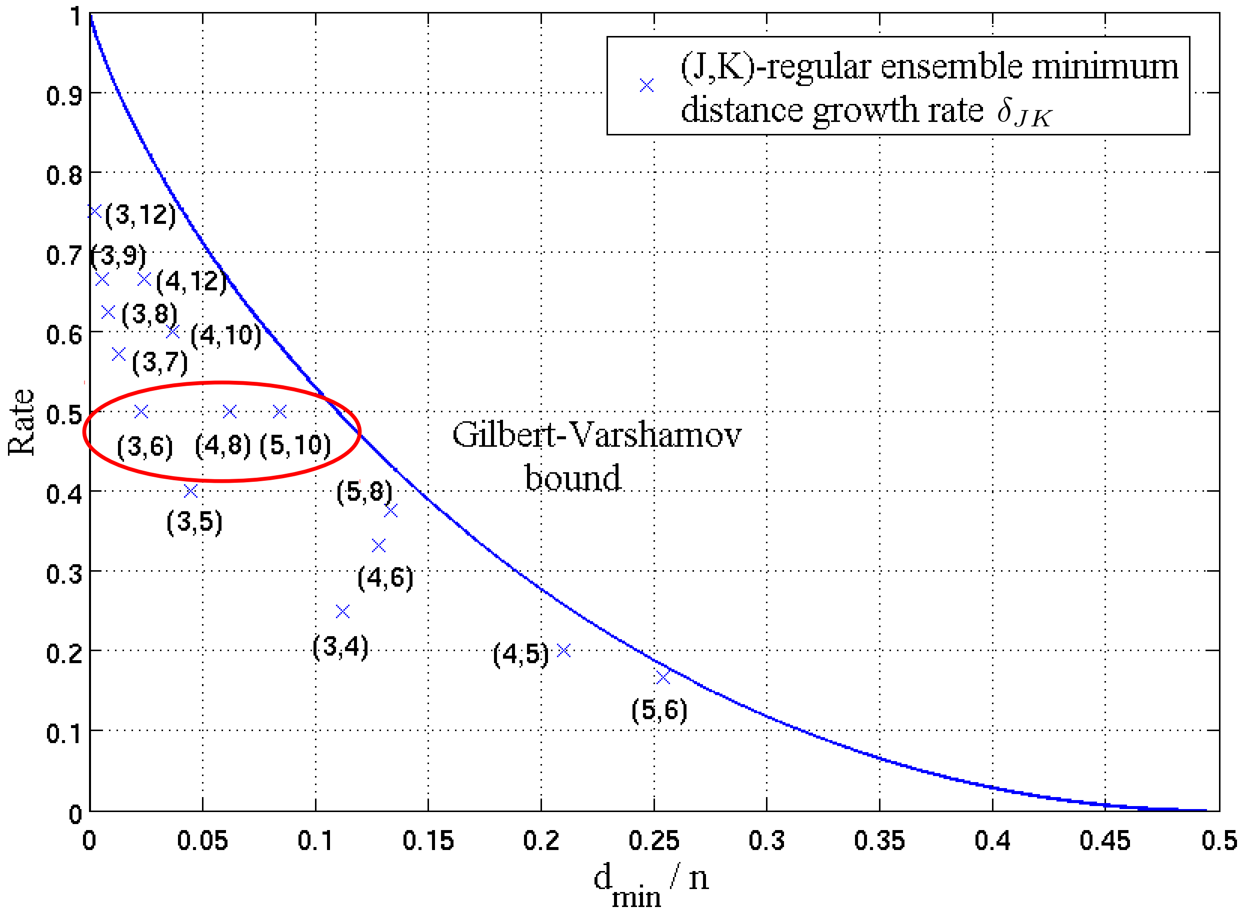
- (J,K)-regular block code ensembles are asymptotically good, i.e.,

$$d_{\min} \geq n\delta_{JK}$$

where δ_{JK} is called the **typical minimum distance ratio**, or **minimum distance growth rate**.

Minimum Distance Growth Rates of (J,K)-Regular LDPC Block Code Ensembles

- For an **asymptotically good** code ensemble, the minimum distance grows linearly with the block length n



- (J,K)-regular block code ensembles are asymptotically good, i.e.,

$$d_{\min} \geq n\delta_{JK}$$

  where δ_{JK} is called the **typical minimum distance ratio**, or **minimum distance growth rate**.

- As the density of (J,K)-regular ensembles increases, δ_{JK} **approaches the Gilbert-Varshamov bound**.

Thresholds of (J,K)-regular LDPC Block Code Ensembles

UNIVERSITY OF NOTRE DAME

- Iterative decoding thresholds can be calculated for (J,K)-regular LDPC block code ensembles using density evolution (DE).

BEC thresholds

J	K	Rate	ε^*	ε_{Sh}
3	6	0.5	0.429	0.5
4	8	0.5	0.383	0.5
5	10	0.5	0.341	0.5
3	5	0.4	0.517	0.6
4	6	0.333	0.506	0.667
3	4	0.25	0.647	0.75

AWGNC thresholds

J	K	Rate	$(E_b/N_0)^*$	$(E_b/N_0)_{\text{Sh}}$
3	6	0.5	1.11	0.184
4	8	0.5	1.61	0.184
5	10	0.5	2.04	0.184
3	5	0.4	0.96	-0.229
4	6	0.333	1.67	-0.480
3	4	0.25	1.00	-0.790

[RU01] T. J. Richardson, and R. Urbanke, "The capacity of low-density parity-check codes under message passing decoding", *IEEE Transactions on Information Theory*, vol. 47 no. 2, Feb. 2001.

Thresholds of (J,K)-regular LDPC Block Code Ensembles

UNIVERSITY OF NOTRE DAME

- Iterative decoding thresholds can be calculated for (J,K)-regular LDPC block code ensembles using density evolution (DE).

BEC thresholds

J	K	Rate	ε^*	$\varepsilon_{\mathrm{Sh}}$
3	6	0.5	0.429	0.5
4	8	0.5	0.383	0.5
5	10	0.5	0.341	0.5
3	5	0.4	0.517	0.6
4	6	0.333	0.506	0.667
3	4	0.25	0.647	0.75

AWGNC thresholds

J	K	Rate	$(E_b/N_0)^*$	$(E_b/N_0)_{\mathrm{Sh}}$
3	6	0.5	1.11	0.184
4	8	0.5	1.61	0.184
5	10	0.5	2.04	0.184
3	5	0.4	0.96	−0.229
4	6	0.333	1.67	−0.480
3	4	0.25	1.00	−0.790

- There exists a relatively **large gap to capacity**.

[RU01] T. J. Richardson, and R. Urbanke, “The capacity of low-density parity-check codes under message passing decoding”, *IEEE Transactions on Information Theory*, vol. 47 no. 2, Feb. 2001.

Thresholds of (J,K)-regular LDPC Block Code Ensembles

- Iterative decoding thresholds can be calculated for (J,K)-regular LDPC block code ensembles using density evolution (DE).

BEC thresholds

J	K	Rate	ε^*	$\varepsilon_{\mathrm{Sh}}$
3	6	0.5	0.429	0.5
4	8	0.5	0.383	0.5
5	10	0.5	0.341	0.5
3	5	0.4	0.517	0.6
4	6	0.333	0.506	0.667
3	4	0.25	0.647	0.75

AWGNC thresholds

J	K	Rate	$(E_b/N_0)^*$	$(E_b/N_0)_{\mathrm{Sh}}$
3	6	0.5	1.11	0.184
4	8	0.5	1.61	0.184
5	10	0.5	2.04	0.184
3	5	0.4	0.96	−0.229
4	6	0.333	1.67	−0.480
3	4	0.25	1.00	−0.790

- There exists a relatively **large gap to capacity**.
- Iterative decoding thresholds get **further from capacity** as the graph **density increases**.

[RU01] T. J. Richardson, and R. Urbanke, "The capacity of low-density parity-check codes under message passing decoding", *IEEE Transactions on Information Theory*, vol. 47 no. 2, Feb. 2001.

Protographs (Matrix Description)

- Large LDPC codes can be obtained from a small **base parity-check matrix B** by replacing each nonzero entry in **B** with an M x M **permutation matrix**, where M is the **lifting factor**.

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}_{3\times 6} \Rightarrow \mathbf{H} = \begin{bmatrix} \mathbf{\Pi}_{1,1} & \mathbf{\Pi}_{1,2} & \mathbf{\Pi}_{1,3} & \mathbf{\Pi}_{1,4} & \mathbf{\Pi}_{1,5} & \mathbf{0} \\ \mathbf{\Pi}_{2,1} & \mathbf{0} & \mathbf{\Pi}_{2,3} & \mathbf{\Pi}_{2,4} & \mathbf{\Pi}_{2,5} & \mathbf{\Pi}_{2,6} \\ \mathbf{0} & \mathbf{\Pi}_{3,2} & \mathbf{\Pi}_{3,3} & \mathbf{0} & \mathbf{0} & \mathbf{\Pi}_{3,6} \end{bmatrix}_{3M\times 6M}$$

Protographs (Matrix Description)

UNIVERSITY OF NOTRE DAME

- Large LDPC codes can be obtained from a small **base parity-check matrix B** by replacing each nonzero entry in **B** with an M x M **permutation matrix**, where M is the **lifting factor**.

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}_{3\times 6} \rightarrow \mathbf{H} = \begin{bmatrix} \boldsymbol{\Pi}_{1,1} & \boldsymbol{\Pi}_{1,2} & \boldsymbol{\Pi}_{1,3} & \boldsymbol{\Pi}_{1,4} & \boldsymbol{\Pi}_{1,5} & \mathbf{0} \\ \boldsymbol{\Pi}_{2,1} & \mathbf{0} & \boldsymbol{\Pi}_{2,3} & \boldsymbol{\Pi}_{2,4} & \boldsymbol{\Pi}_{2,5} & \boldsymbol{\Pi}_{2,6} \\ \mathbf{0} & \boldsymbol{\Pi}_{3,2} & \boldsymbol{\Pi}_{3,3} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Pi}_{3,6} \end{bmatrix}_{3M\times 6M}$$

Example: Irregular code with M = 4

$$\mathbf{H} = \left[\begin{array}{cccc|cccc|cccc|cccc|cccc|cccc}
0&0&1&0&1&0&0&0&0&0&1&0&1&0&0&0&0&0&1&0&0&0&0&0\\
1&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0\\
0&0&0&1&0&0&1&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0\\
0&1&0&0&0&1&0&0&1&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0\\
\hline
1&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&1&0&0&0&0&1&0\\
0&1&0&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&1&0&0&0&1\\
0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&1&0&0&0&1&0&0&0\\
0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&1&0&0&1&0&0\\
\hline
0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0
\end{array}\right]$$

- length $6M$ = 24
- rate R = 1/2

Protographs (Matrix Description)

UNIVERSITY OF NOTRE DAME

- Large LDPC codes can be obtained from a small **base parity-check matrix B** by replacing each nonzero entry in **B** with an M x M **permutation matrix**, where M is the **lifting factor**.

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}_{3\times 6} \Rightarrow \mathbf{H} = \begin{bmatrix} \mathbf{\Pi}_{1,1} & \mathbf{\Pi}_{1,2} & \mathbf{\Pi}_{1,3} & \mathbf{\Pi}_{1,4} & \mathbf{\Pi}_{1,5} & \mathbf{0} \\ \mathbf{\Pi}_{2,1} & \mathbf{0} & \mathbf{\Pi}_{2,3} & \mathbf{\Pi}_{2,4} & \mathbf{\Pi}_{2,5} & \mathbf{\Pi}_{2,6} \\ \mathbf{0} & \mathbf{\Pi}_{3,2} & \mathbf{\Pi}_{3,3} & \mathbf{0} & \mathbf{0} & \mathbf{\Pi}_{3,6} \end{bmatrix}_{3M\times 6M}$$

Example: Irregular code with M = 4

$$\mathbf{H} = \left[\begin{array}{cccc|cccc|cccc|cccc|cccc|cccc}
0&0&1&0&1&0&0&0&0&0&1&0&1&0&0&0&0&0&1&0&0&0&0&0 \\
1&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0 \\
0&0&0&1&0&0&1&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0 \\
0&1&0&0&0&1&0&0&1&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0 \\ \hline
1&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&1&0&0&0&0&1&0 \\
0&1&0&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&1&0&0&0&1 \\
0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&1&0&0&0&1&0&0&0 \\
0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&1&0&0&1&0&0 \\ \hline
0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0 \\
0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1 \\
0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0 \\
0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0
\end{array}\right]$$

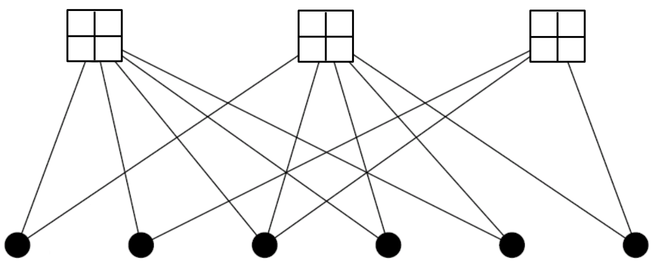
- length $6M$ = 24
  rate R = 1/2
- Irregular codes have variable row and column weights (check node and variable node degrees)

Protographs (Graphical Description)

- Protographs are often represented using a bipartite **Tanner graph**

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}_{3\times 6} \Longleftrightarrow$$

3 check nodes



6 variable nodes

[Tho05] J. Thorpe, "Low-Density Parity-Check (LDPC) codes constructed from protographs", *Jet Propulsion Laboratory INP Progress Report*, Vol. 42-154 Aug. 2003.

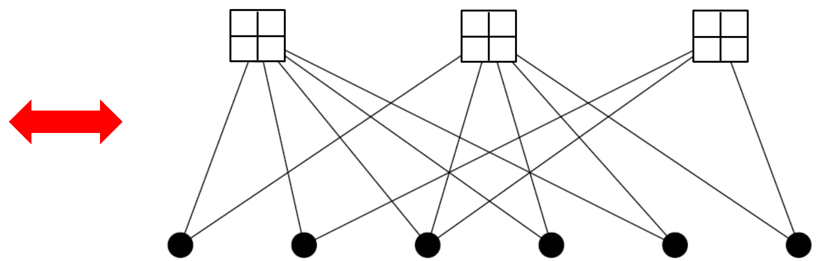
Protographs (Graphical Description)

UNIVERSITY OF NOTRE DAME

- Protographs are often represented using a bipartite **Tanner graph**

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}_{3\times 6}$$

3 check nodes



6 variable nodes

- The collection of all possible parity-check matrices with lifting factor M forms a **code ensemble**, where all the codes share a **common structure**

$$\mathbf{H} = \begin{bmatrix} \mathbf{\Pi}_{1,1} & \mathbf{\Pi}_{1,2} & \mathbf{\Pi}_{1,3} & \mathbf{\Pi}_{1,4} & \mathbf{\Pi}_{1,5} & \mathbf{0} \\ \mathbf{\Pi}_{2,1} & \mathbf{0} & \mathbf{\Pi}_{2,3} & \mathbf{\Pi}_{2,4} & \mathbf{\Pi}_{2,5} & \mathbf{\Pi}_{2,6} \\ \mathbf{0} & \mathbf{\Pi}_{3,2} & \mathbf{\Pi}_{3,3} & \mathbf{0} & \mathbf{0} & \mathbf{\Pi}_{3,6} \end{bmatrix}$$

[Tho05] J. Thorpe, "Low-Density Parity-Check (LDPC) codes constructed from protographs", *Jet Propulsion Laboratory INP Progress Report*, Vol. 42-154 Aug. 2003.

Quasi-Cyclic LDPC Codes

- **Quasi-cyclic** (QC) LDPC codes are of great interest in practice, since they have **efficient encoder and decoder implementations**

Quasi-Cyclic LDPC Codes

- **Quasi-cyclic** (QC) LDPC codes are of great interest in practice, since they have **efficient encoder and decoder implementations**

Example: protograph construction of a $(2,3)$**-regular** QC-LDPC block code

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}_{2 \times 3}$$

Quasi-Cyclic LDPC Codes

UNIVERSITY OF NOTRE DAME

- **Quasi-cyclic** (QC) LDPC codes are of great interest in practice, since they have **efficient encoder and decoder implementations**

Example: protograph construction of a **$(2,3)$-regular** QC-LDPC block code

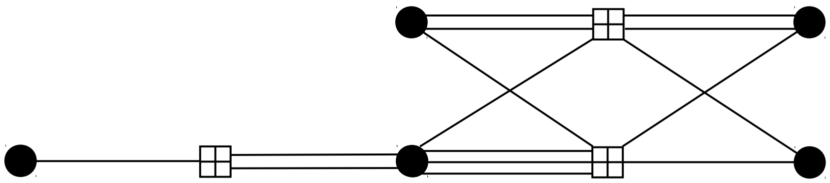
$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}_{2\times 3} \Rightarrow \mathbf{H} = \left[\begin{array}{ccccccc|ccccccc|ccccccc}
0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&1&0&0&0\\
1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0\\
0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0\\
0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&1&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0\\
\hline
0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0\\
0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1&0\\
0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1\\
0&0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0\\
0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&1&0&0&0&0\\
1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0
\end{array}\right]_{14\times 21}$$

$$M = 7$$
$$n = 3M = 21$$
$$R = 1/3$$

For QC codes, the permutation matrices are **shifted identities**

Multi-Edge Protographs

- Protographs can have repeated edges (corresponding to integer values greater than one in **B**)



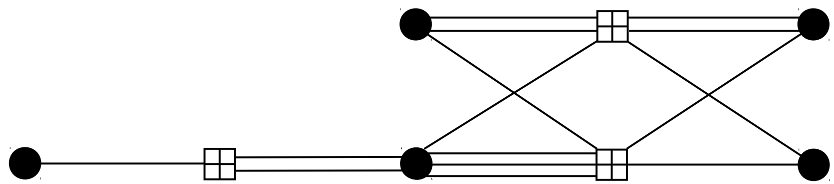
$$\mathbf{B} = \begin{bmatrix} 2 & 2 & 0 & 1 & 1 \\ 0 & 0 & 1 & 2 & 0 \\ 1 & 1 & 0 & 3 & 1 \end{bmatrix}_{3 \times 5}$$

- Note that this makes no sense without lifting

[DDJA09] D. Divsalar, S. Dolinar, C. Jones, and K. Andrews, "Capacity-approaching protograph codes", *IEEE Journal on Select Areas in Communications*, vol. 27, no. 6 Aug. 2009.

Multi-Edge Protographs

- Protographs can have repeated edges (corresponding to integer values greater than one in **B**)



$$\mathbf{B} = \begin{bmatrix} 2 & 2 & 0 & 1 & 1 \\ 0 & 0 & 1 & 2 & 0 \\ 1 & 1 & 0 & 3 & 1 \end{bmatrix}_{3 \times 5}$$

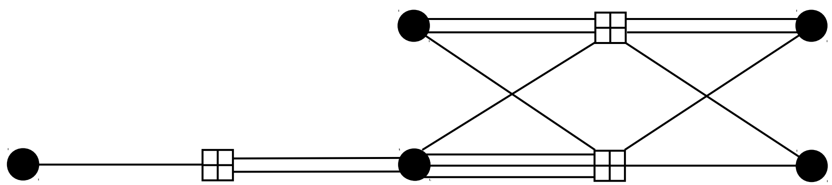
- Note that this makes no sense without lifting
- Repeated edges in a protograph correspond to using sums of permutation matrices to form LDPC code ensembles

$$\mathbf{H} = \left[\begin{array}{ccc|ccc|ccc|ccc|ccc} 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \end{array}\right]$$

[DDJA09] D. Divsalar, S. Dolinar, C. Jones, and K. Andrews, "Capacity-approaching protograph codes", *IEEE Journal on Select Areas in Communications*, vol. 27, no. 6 Aug. 2009.

Multi-Edge Protographs

- Protographs can have repeated edges (corresponding to integer values greater than one in **B**)



$$\mathbf{B} = \begin{bmatrix} 2 & 2 & 0 & 1 & 1 \\ 0 & 0 & 1 & 2 & 0 \\ 1 & 1 & 0 & 3 & 1 \end{bmatrix}_{3 \times 5}$$

- Note that this makes no sense without lifting

- Repeated edges in a protograph correspond to using sums of permutation matrices to form LDPC code ensembles

$$\mathbf{H} = \left[\begin{array}{ccc|ccc|ccc|ccc|ccc} 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \end{array}\right]$$

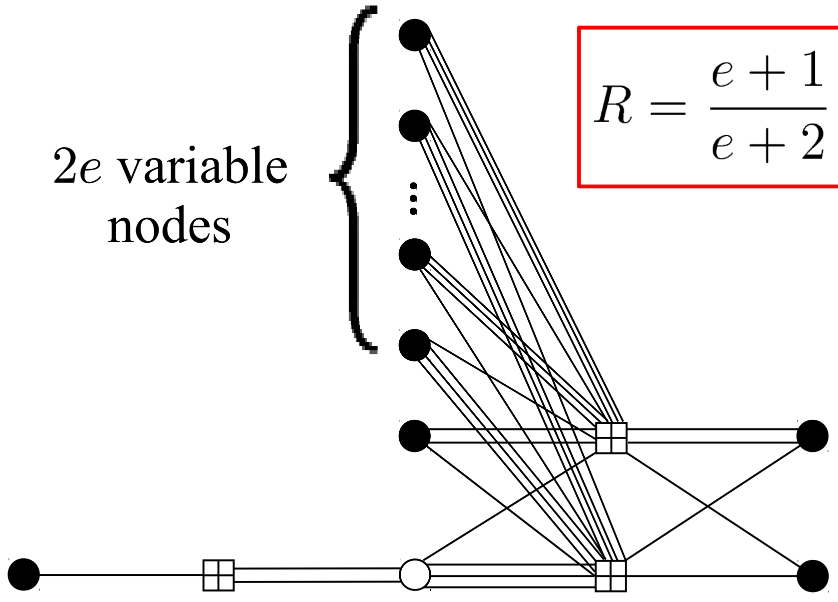
➡ denser graphs!

➡ can also be QC (using **circulant matrices**)!

[DDJA09] D. Divsalar, S. Dolinar, C. Jones, and K. Andrews, "Capacity-approaching protograph codes", *IEEE Journal on Select Areas in Communications*, vol. 27, no. 6 Aug. 2009.

'Good' Protographs

- Ensemble average properties can be easily calculated from a protograph, thus simplifying the construction of 'good' code ensembles.
  - ➡ Iterative decoding thresholds close to capacity for irregular protographs
  - ➡ Minimum distance growing linearly with block length (**asymptotically good**) for regular and some irregular protographs

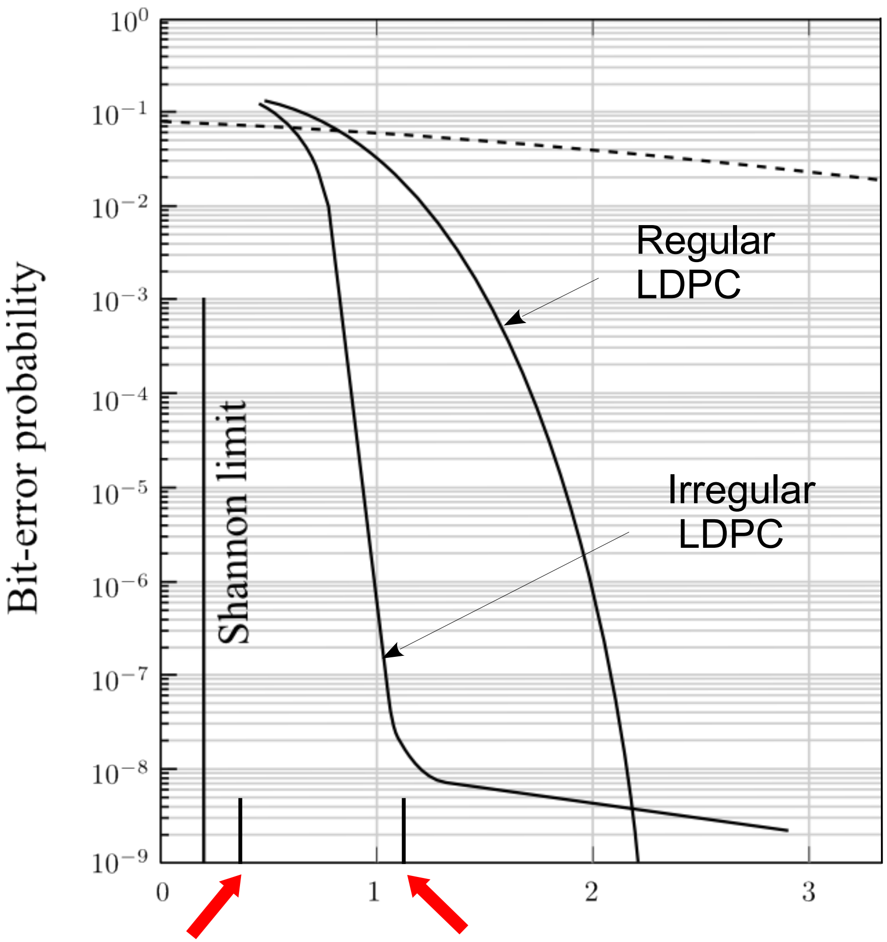


$$R = \frac{e+1}{e+2}$$

Rate	Threshold $(E_b/N_0)^*$	Capacity $(E_b/N_0)_{\rm Sh}$	Distance growth rate
1/2	0.628	0.187	0.01450
2/3	1.450	1.059	0.00582
3/4	2.005	1.628	0.00323
4/5	2.413	2.040	0.00207
5/6	2.733	2.362	0.00145
6/7	2.993	2.625	0.00108

[DDJA09] D. Divsalar, S. Dolinar, C. Jones, and K. Andrews, "Capacity-approaching protograph codes", *IEEE Journal on Select Areas in Communications*, vol. 27, no. 6 Aug. 2009.

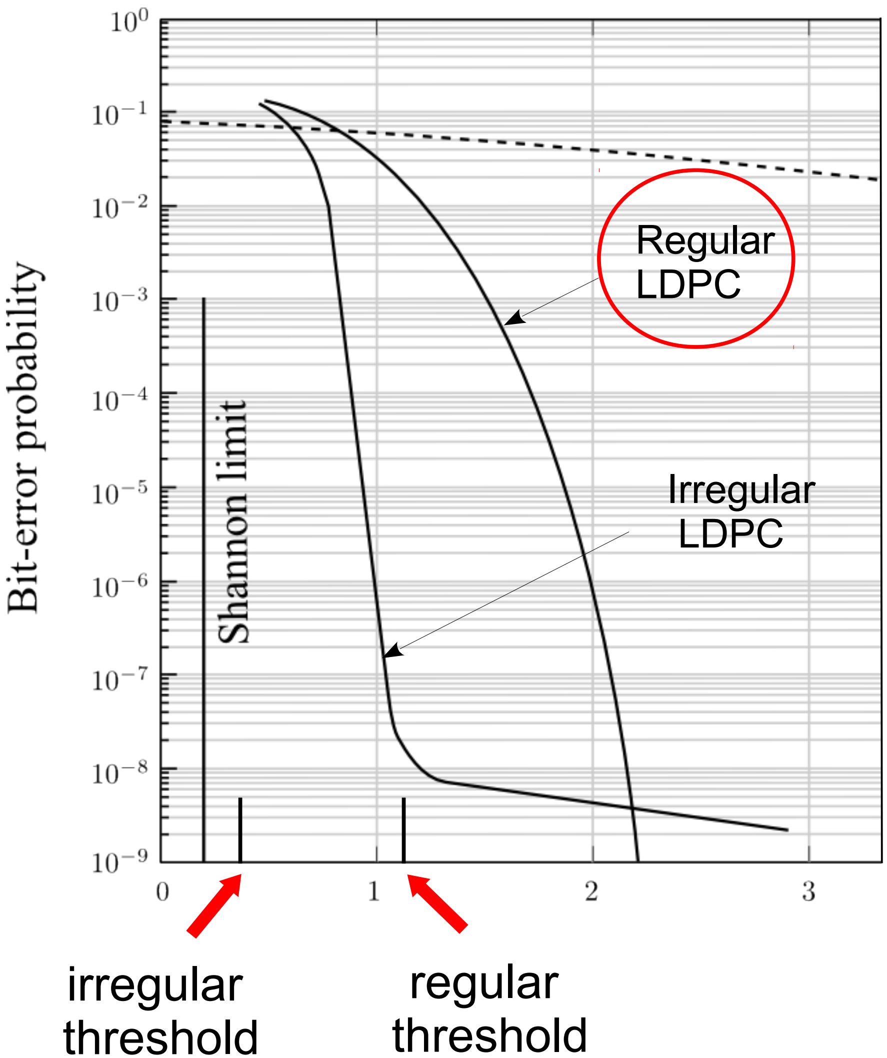
Regular vs. Irregular LDPC codes



irregular threshold

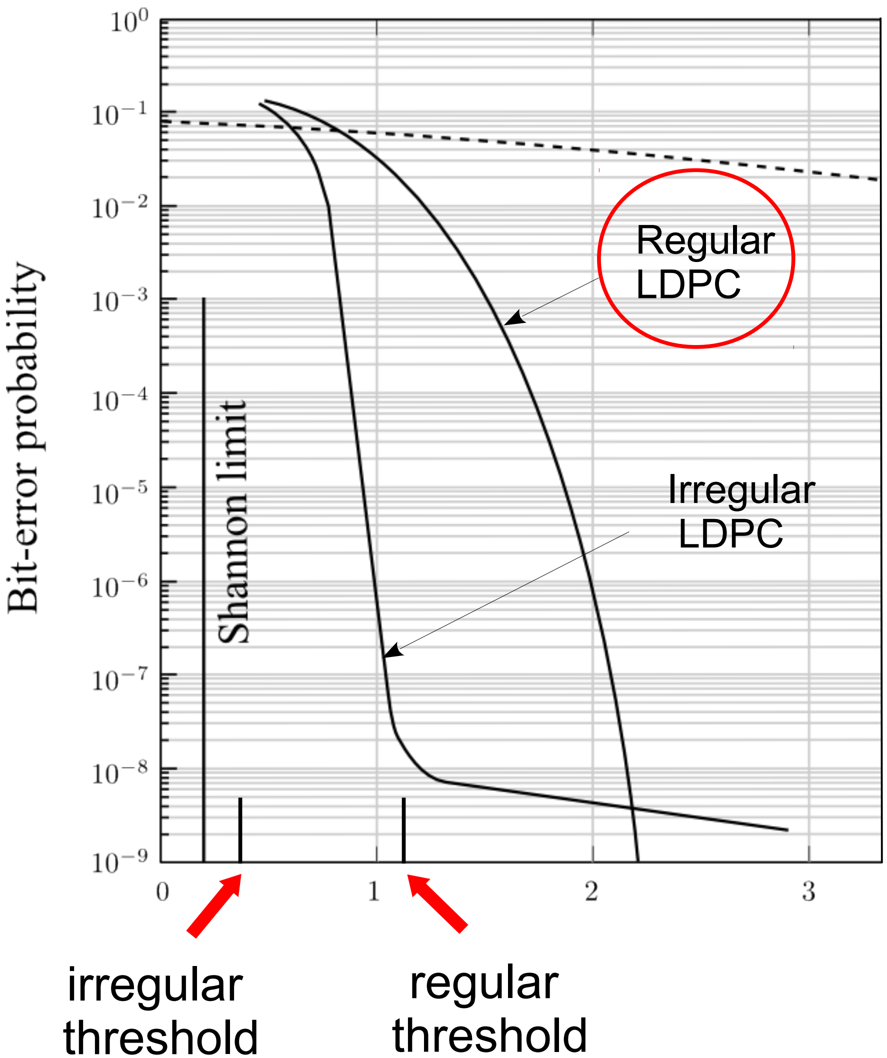
regular threshold

Regular vs. Irregular LDPC codes



- "Regular" LDPC codes:
  - ✓ structure **aids implementation**
  - ✓ low **error floors**
  - ✗ thresholds **far from capacity**

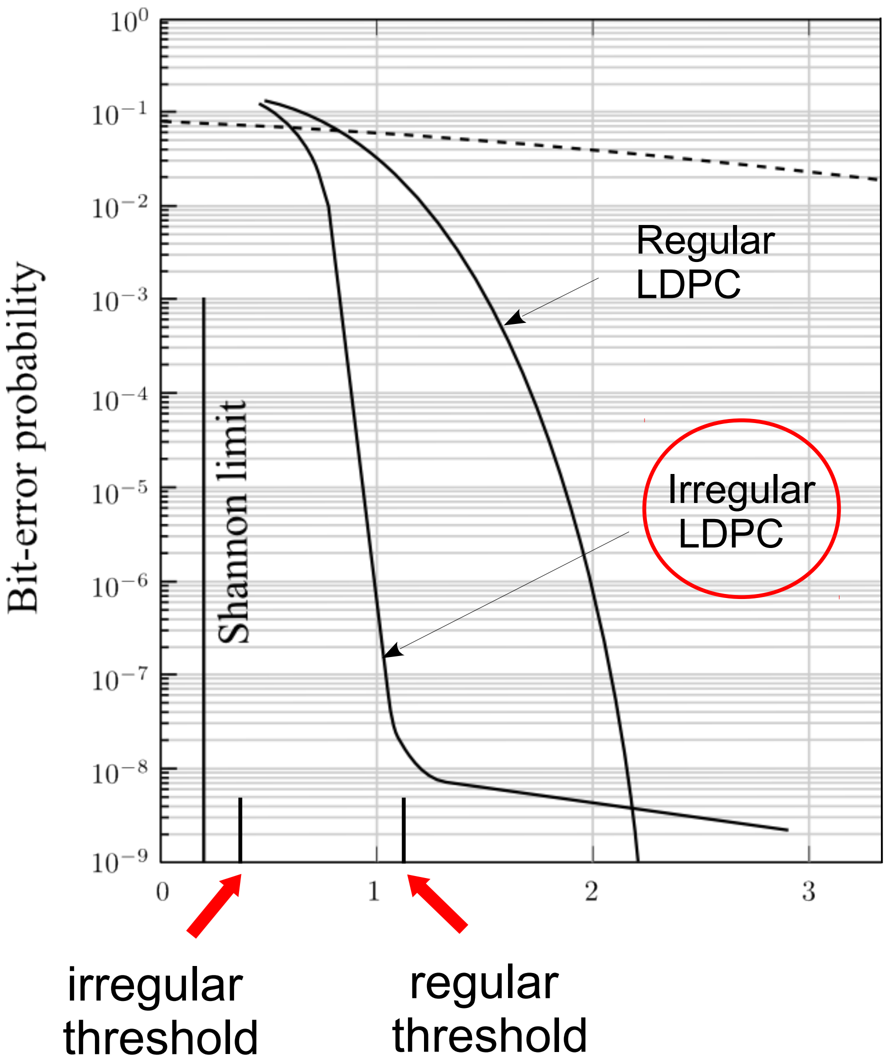
Regular vs. Irregular LDPC codes



- "Regular" LDPC codes:
  - ✓ structure **aids implementation**
  - ✓ low **error floors**
  - ✗ thresholds **far from capacity**
  - ➡ not suitable for **severely power constrained** applications

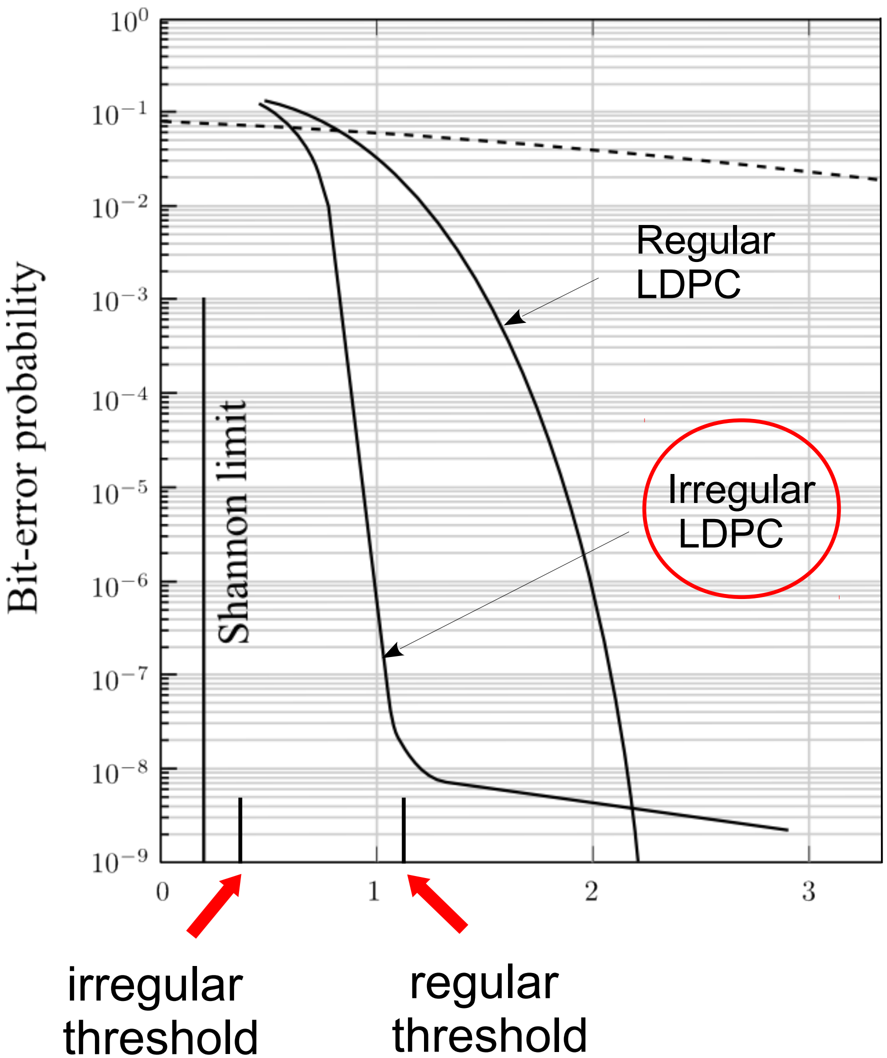
Regular vs. Irregular LDPC codes

UNIVERSITY OF NOTRE DAME



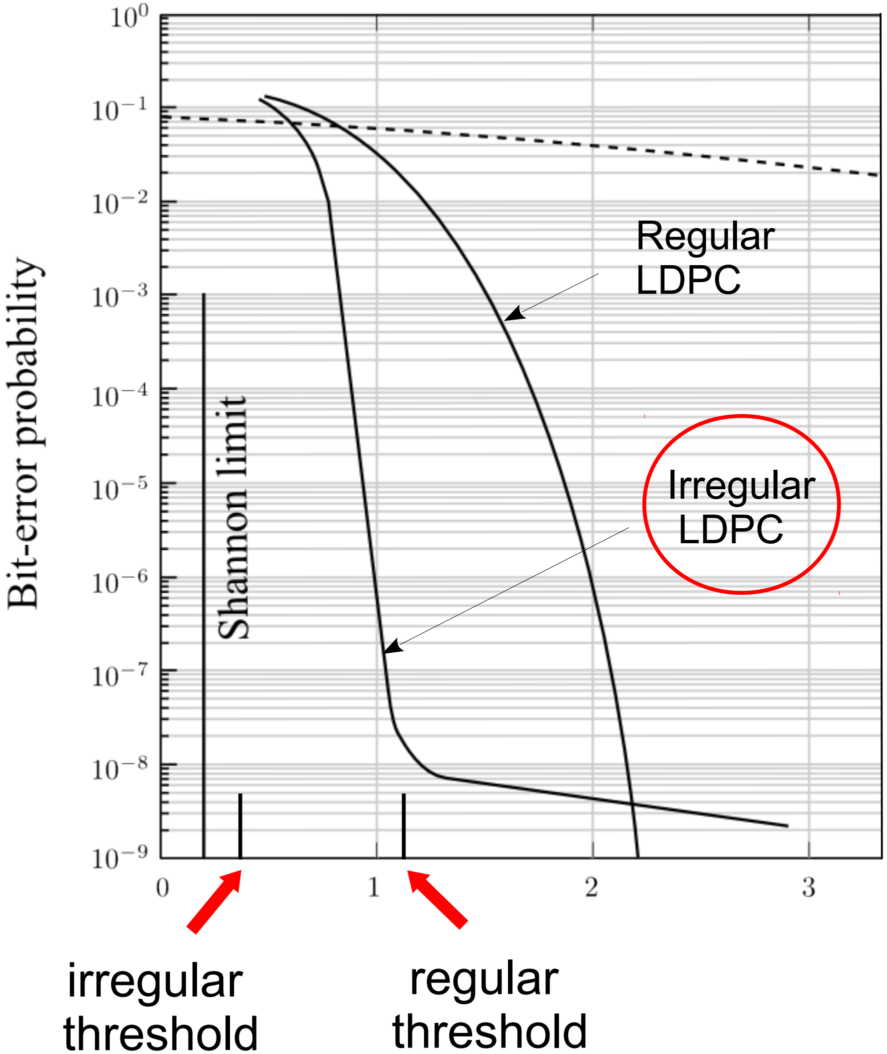
- "Regular" LDPC codes:
  - ✓ structure **aids implementation**
  - ✓ low **error floors**
  - x thresholds **far from capacity**
  - ➡ not suitable for **severely power constrained** applications
- "Irregular" LDPC codes:
  - x Less desirable structure
  - ✓ thresholds **close to capacity**
  - x visible **error floors**

Regular vs. Irregular LDPC codes



- "Regular" LDPC codes:
  - ✓ structure **aids implementation**
  - ✓ low **error floors**
  - ✗ thresholds **far from capacity**
  - ➡ not suitable for **severely power constrained** applications
- "Irregular" LDPC codes:
  - ✗ Less desirable structure
  - ✓ thresholds **close to capacity**
  - ✗ visible **error floors**
  - ➡ not suitable for applications that require **very low error rates**

Regular vs. Irregular LDPC codes



- "Regular" LDPC codes:
  - ✓ structure **aids implementation**
  - ✓ low **error floors**
  - x thresholds **far from capacity**
  - ➡ not suitable for **severely power constrained** applications
- "Irregular" LDPC codes:
  - x Less desirable structure
  - ✓ thresholds **close to capacity**
  - x visible **error floors**
  - ➡ not suitable for applications that require **very low error rates**
- **Spatially coupled** LDPC codes combine all of the positive features!

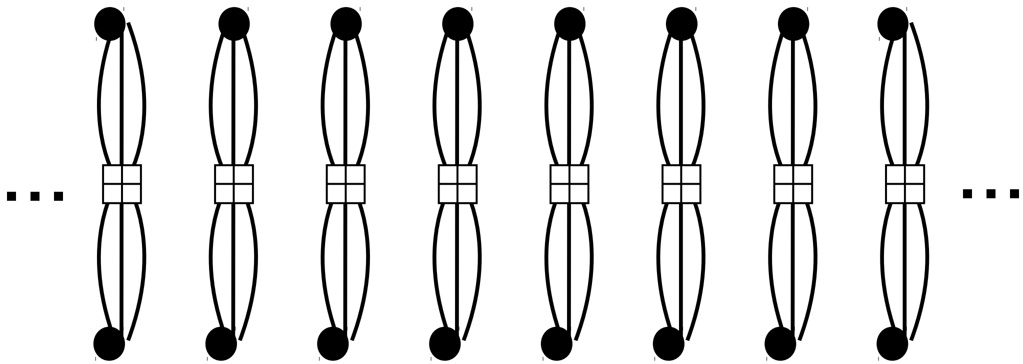
Outline

UNIVERSITY OF NOTRE DAME

- **LDPC Block Codes**
  - Parity-check matrix and Tanner graph representations, minimum distance bounds, iterative decoding thresholds, protograph-based constructions
- **Spatially Coupled LDPC Codes**
  - Protograph representation, edge-spreading construction, termination
  - Iterative decoding thresholds, threshold saturation, minimum distance
- **Practical Considerations**
  - Window decoding; performance, latency, and complexity comparisons to LDPC block codes; rate-compatibility; implementation aspects

Spatially Coupled Protographs

- Consider transmission of consecutive blocks (protograph representation):

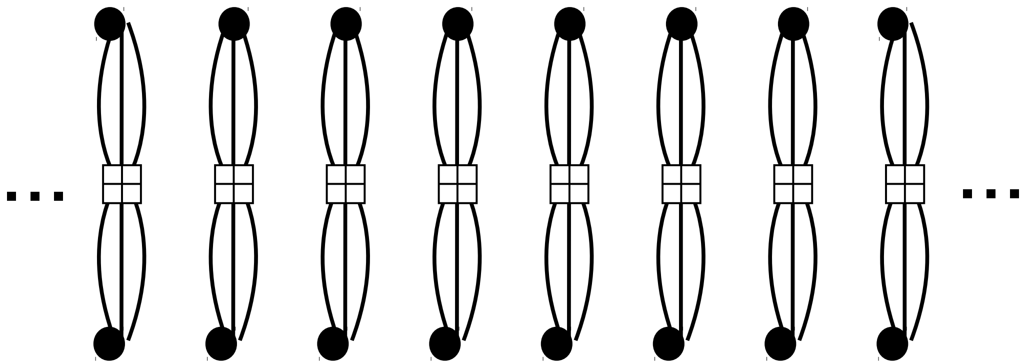


$$\mathbf{B} = \begin{bmatrix} 3 & 3 \end{bmatrix}_{b_c \times b_v}$$

$(3{,}6)$-regular LDPC-BC base matrix

Spatially Coupled Protographs

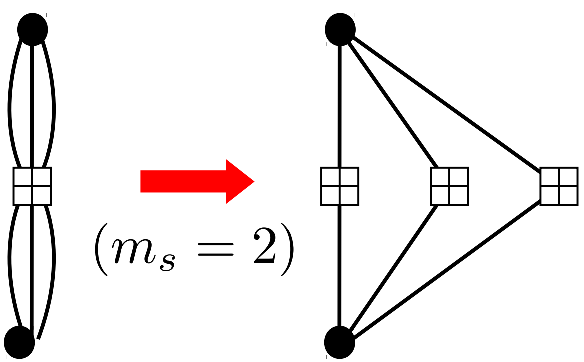
- Consider transmission of consecutive blocks (protograph representation):



$$\mathbf{B} = \begin{bmatrix} 3 & 3 \end{bmatrix}_{b_c \times b_v}$$

(3,6)-regular LDPC-BC base matrix

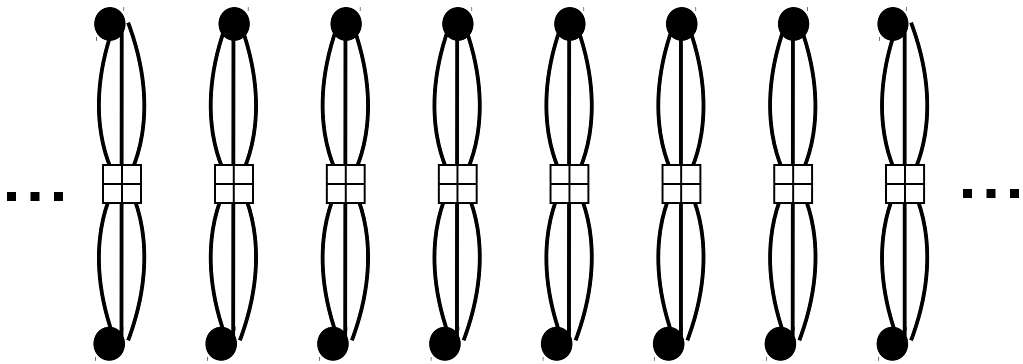
- Blocks are **spatially coupled** (introducing **memory**) by **spreading edges** over time:



$$\mathbf{B} = \begin{bmatrix} 3 & 3 \end{bmatrix} \xrightarrow{(m_s = 2)} \begin{bmatrix} \mathbf{B}_0 \\ \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Spatially Coupled Protographs

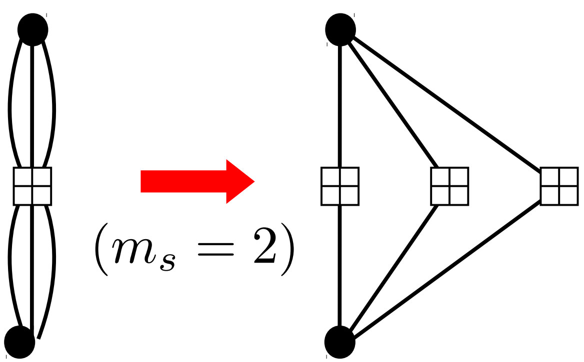
- Consider transmission of consecutive blocks (protograph representation):



$$\mathbf{B} = \begin{bmatrix} 3 & 3 \end{bmatrix}_{b_c \times b_v}$$

(3,6)-regular LDPC-BC base matrix

- Blocks are **spatially coupled** (introducing **memory**) by **spreading edges** over time:

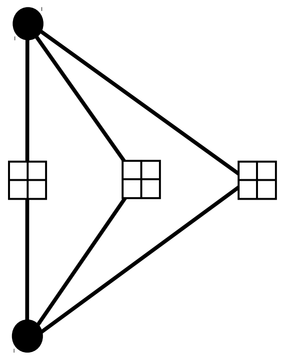


$$\mathbf{B} = \begin{bmatrix} 3 & 3 \end{bmatrix} \xrightarrow{(m_s = 2)} \begin{bmatrix} \mathbf{B}_0 \\ \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}$$

- **Spreading constraint:** $\sum_{i=0}^{m_s} \mathbf{B}_i = \mathbf{B}$ ($\mathbf{B}_i$ has size $b_c \times b_v$)

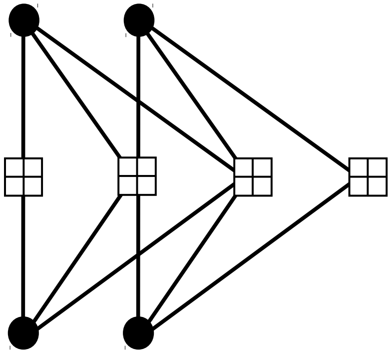
Spatially Coupled Protographs

- Transmission of consecutive spatially coupled (SC) blocks results in a **convolutional protograph**:



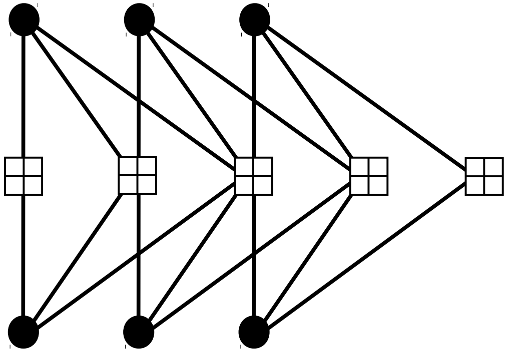
Spatially Coupled Protographs

- Transmission of consecutive spatially coupled (SC) blocks results in a **convolutional protograph**:



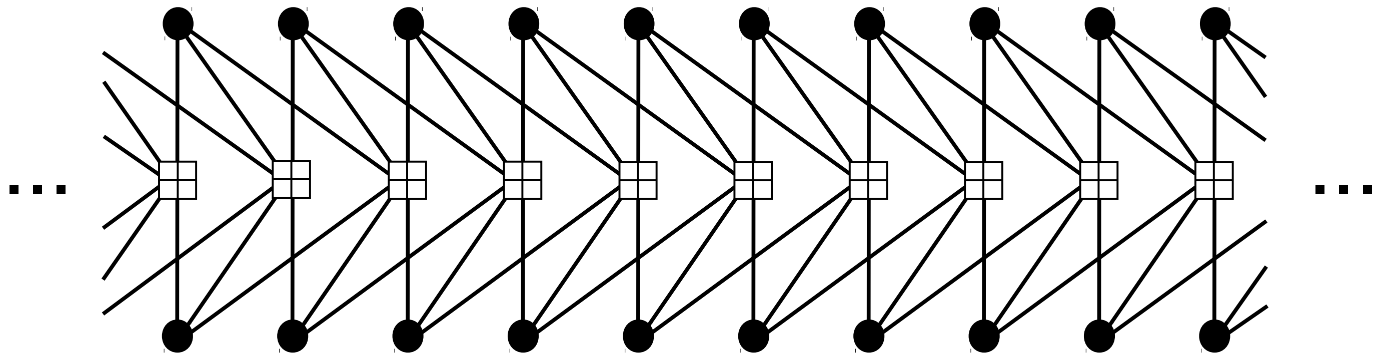
Spatially Coupled Protographs

- Transmission of consecutive spatially coupled (SC) blocks results in a **convolutional protograph**:



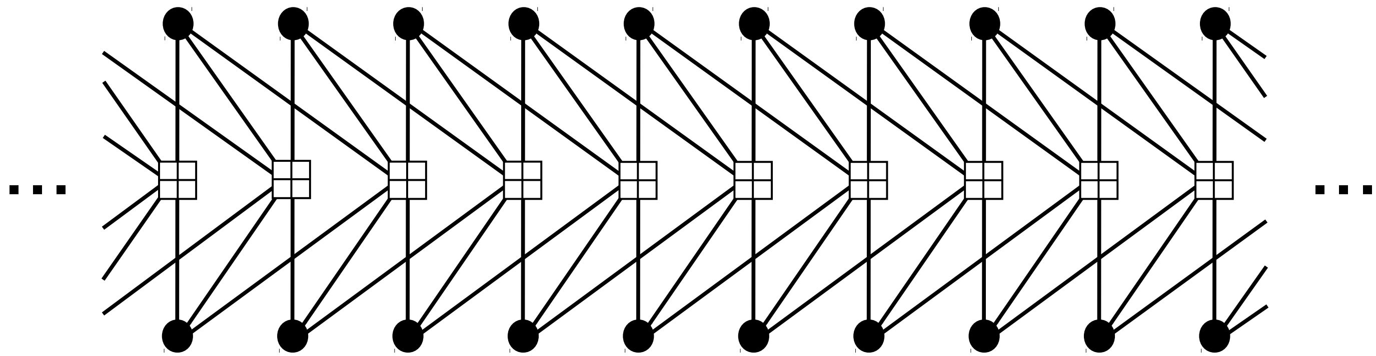
Spatially Coupled Protographs

- Transmission of consecutive spatially coupled (SC) blocks results in a **convolutional protograph**:



Spatially Coupled Protographs

- Transmission of consecutive spatially coupled (SC) blocks results in a **convolutional protograph**:



- The bi-infinite convolutional protograph corresponds to a bi-infinite **convolutional base matrix**:

$$\mathbf{B}_{[-\infty,\infty]} = \begin{bmatrix} \ddots & & \ddots & \ddots & & & & \\ & \mathbf{B}_{m_s} & \cdots & \mathbf{B}_1 & \mathbf{B}_0 & & & \\ & & \ddots & & \ddots & \ddots & & \\ & & & \mathbf{B}_{m_s} & \cdots & \mathbf{B}_1 & \mathbf{B}_0 & \\ & & & & \ddots & & \ddots & \ddots \end{bmatrix}$$

Rate:

$$R = \frac{b_v - b_c}{b_v}$$

Constraint length:

$$\nu_s = b_v(m_s + 1)$$

SC-LDPC Code Ensembles

- An ensemble of (3,6)-regular SC-LDPC codes can be created from the **convolutional protograph** by the graph lifting operation

$$\mathbf{B}_{[-\infty,\infty]} = \begin{bmatrix} \ddots & \ddots & \ddots & & & \\ \mathbf{B}_2 & \mathbf{B}_1 & \mathbf{B}_0 & & & \\ & \mathbf{B}_2 & \mathbf{B}_1 & \mathbf{B}_0 & & \\ & & \mathbf{B}_2 & \mathbf{B}_1 & \mathbf{B}_0 & \\ & & & \ddots & \ddots & \ddots \end{bmatrix}$$

SC-LDPC Code Ensembles

- An ensemble of (3,6)-regular SC-LDPC codes can be created from the **convolutional protograph** by the graph lifting operation

$$\mathbf{B}_{[-\infty,\infty]} = \begin{bmatrix} \ddots & & \ddots & & \ddots & & & & & \\ 1 & 1 & 1 & 1 & 1 & 1 & & & & \\ & & 1 & 1 & 1 & 1 & 1 & 1 & & \\ & & & & 1 & 1 & 1 & 1 & 1 & 1 \\ & & & & & \ddots & & \ddots & & \ddots \end{bmatrix}$$

$$\mathbf{B}_i = [\,1 \quad 1\,]$$
$$b_c = 1$$
$$b_v = 2$$
$$m_s = 2$$

SC-LDPC Code Ensembles

UNIVERSITY OF NOTRE DAME

- An ensemble of (3,6)-regular SC-LDPC codes can be created from the **convolutional protograph** by the graph lifting operation

$$\mathbf{B}_{[-\infty,\infty]} = \begin{bmatrix} \ddots & & \ddots & & \ddots & & & & & & \\ 1 & 1 & 1 & 1 & 1 & 1 & & & & \\ & & 1 & 1 & 1 & 1 & 1 & 1 & & \\ & & & & 1 & 1 & 1 & 1 & 1 & 1 \\ & & & & & \ddots & & \ddots & & \ddots \end{bmatrix}$$

$$\mathbf{B}_i = [1 \quad 1], \quad b_c = 1, \quad b_v = 2, \quad m_s = 2$$

Graph lifting: $\Pi_{i,j}$ is an $M \times M$ permutation matrix

$$\nu_s = M b_v (m_s + 1) = 6M$$

$$\mathbf{H}_{\text{cc}} = \begin{bmatrix} \ddots & & \ddots & & \ddots & & & & & \\ \Pi_{5,t} & \Pi_{4,t} & \Pi_{3,t} & \Pi_{2,t} & \Pi_{1,t} & \Pi_{0,t} & & & & \\ & & \Pi_{5,t+1} & \Pi_{4,t+1} & \Pi_{3,t+1} & \Pi_{2,t+1} & \Pi_{1,t+1} & \Pi_{0,t+1} & & \\ & & & & \Pi_{5,t+2} & \Pi_{4,t+2} & \Pi_{3,t+2} & \Pi_{2,t+2} & \Pi_{1,t+2} & \Pi_{0,t+2} \\ & & & & & \ddots & & \ddots & & \ddots \end{bmatrix}$$

SC-LDPC Code Ensembles

UNIVERSITY OF NOTRE DAME

- An ensemble of (3,6)-regular SC-LDPC codes can be created from the **convolutional protograph** by the graph lifting operation

$$\mathbf{B}_{[-\infty,\infty]} = \begin{bmatrix} \ddots & \ddots & \ddots & & & \\ 1\ \ 1 & 1\ \ 1 & 1\ \ 1 & & & \\ & 1\ \ 1 & 1\ \ 1 & 1\ \ 1 & & \\ & & 1\ \ 1 & 1\ \ 1 & 1\ \ 1 & \\ & & & \ddots & \ddots & \ddots \end{bmatrix} \qquad \begin{aligned} \mathbf{B}_i &= [1\ \ 1] \\ b_c &= 1 \\ b_v &= 2 \\ m_s &= 2 \end{aligned}$$

Graph lifting: $\Pi_{i,j}$ is an $M \times M$ permutation matrix

$$\nu_s = M b_v (m_s + 1) = 6M$$

$$\mathbf{H}_{\text{cc}} = \begin{bmatrix} \ddots & \ddots & \ddots & & & \\ \Pi_{5,t}\ \ \Pi_{4,t} & \Pi_{3,t}\ \ \Pi_{2,t} & \Pi_{1,t}\ \ \Pi_{0,t} & & & \\ & \Pi_{5,t+1}\ \ \Pi_{4,t+1} & \Pi_{3,t+1}\ \ \Pi_{2,t+1} & \Pi_{1,t+1}\ \ \Pi_{0,t+1} & & \\ & & \Pi_{5,t+2}\ \ \Pi_{4,t+2} & \Pi_{3,t+2}\ \ \Pi_{2,t+2} & \Pi_{1,t+2}\ \ \Pi_{0,t+2} & \\ & & & \ddots & \ddots & \ddots \end{bmatrix}$$

- If each permutation matrix $\Pi_{i,j}$ is **circulant**, the codes are **quasi-cyclic**

Terminated Spatially Coupled Codes

- Consider **terminating** $\mathbf{B}_{[-\infty,\infty]}$ to a (block code) **base matrix** of length Lb_v:

$$\mathbf{B}_{[0,L-1]} = \begin{bmatrix} \mathbf{B}_0 & & \\ \vdots & \ddots & \\ \mathbf{B}_{m_s} & & \mathbf{B}_0 \\ & \ddots & \vdots \\ & & \mathbf{B}_{m_s} \end{bmatrix}_{(L+m_s)b_c \times Lb_v}$$

Code rate:

$$R_L = \frac{Lb_v - (L+m_s)b_c}{Lb_v}.$$

Terminated Spatially Coupled Codes

UNIVERSITY OF NOTRE DAME

- Consider **terminating** $\mathbf{B}_{[-\infty,\infty]}$ to a (block code) **base matrix** of length Lb_v:

$$\mathbf{B}_{[0,L-1]} = \begin{bmatrix} \mathbf{B}_0 & & \\ \vdots & \ddots & \\ \mathbf{B}_{m_s} & & \mathbf{B}_0 \\ & \ddots & \vdots \\ & & \mathbf{B}_{m_s} \end{bmatrix}_{(L+m_s)b_c \times Lb_v}$$

Code rate:

$$R_L = \frac{Lb_v - (L+m_s)b_c}{Lb_v}.$$

- For large L, R_L approaches the **unterminated** code rate $R = (b_v - b_c)/b_v$.

Terminated Spatially Coupled Codes

UNIVERSITY OF NOTRE DAME

- Consider **terminating** $\mathbf{B}_{[-\infty,\infty]}$ to a (block code) **base matrix** of length Lb_v:

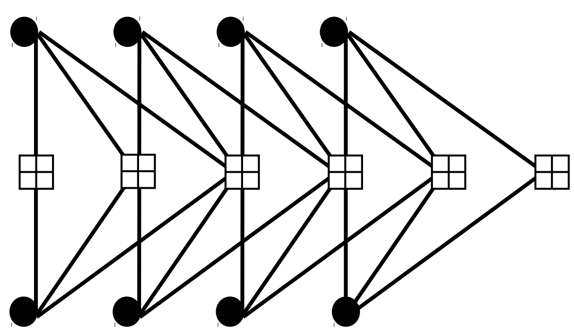
$$\mathbf{B}_{[0,L-1]} = \begin{bmatrix} \mathbf{B}_0 & & \\ \vdots & \ddots & \\ \mathbf{B}_{m_s} & & \mathbf{B}_0 \\ & \ddots & \vdots \\ & & \mathbf{B}_{m_s} \end{bmatrix}_{(L+m_s)b_c \times Lb_v}$$

Code rate:

$$R_L = \frac{Lb_v - (L+m_s)b_c}{Lb_v}.$$

- For large L, R_L approaches the **unterminated** code rate $R = (b_v - b_c)/b_v$.
- **Example:** $(3,6)$-regular base matrix $\mathbf{B} = [3 \;\; 3]$, $m_s = 2$, $L = 4$, $R_4 = 1/4$

$$\mathbf{B}_{[0,3]} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$



(**check node degrees lower at the ends**)

Terminated Spatially Coupled Codes

UNIVERSITY OF NOTRE DAME

- Consider **terminating** $\mathbf{B}_{[-\infty,\infty]}$ to a (block code) **base matrix** of length Lb_v:

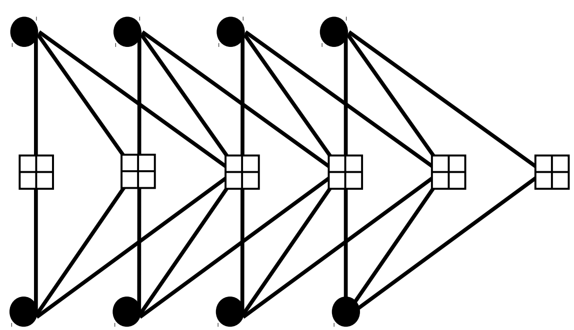
$$\mathbf{B}_{[0,L-1]} = \begin{bmatrix} \mathbf{B}_0 & & \\ \vdots & \ddots & \\ \mathbf{B}_{m_s} & & \mathbf{B}_0 \\ & \ddots & \vdots \\ & & \mathbf{B}_{m_s} \end{bmatrix}_{(L+m_s)b_c \times Lb_v}$$

Code rate:

$$R_L = \frac{Lb_v - (L+m_s)b_c}{Lb_v}.$$

- For large L, R_L approaches the **unterminated** code rate $R = (b_v - b_c)/b_v$.
- **Example:** $(3,6)$-regular base matrix $\mathbf{B} = [3 \;\; 3]$, $m_s = 2$, $L = 4$, $R_4 = 1/4$

$$\mathbf{B}_{[0,3]} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

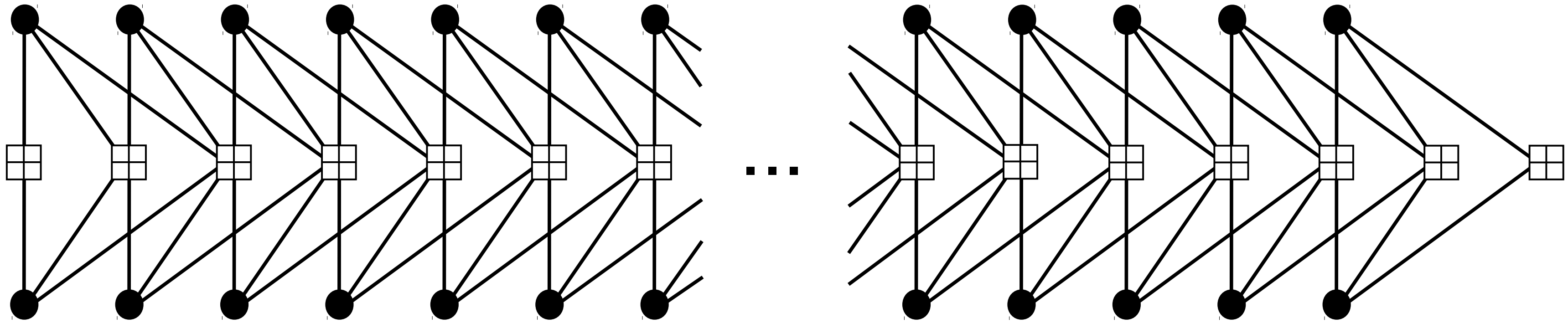


(**check node degrees lower at the ends**)

- Codes can be **lifted** to **different lengths and rates** by varying M and L.

Thresholds of SC-LDPC Codes

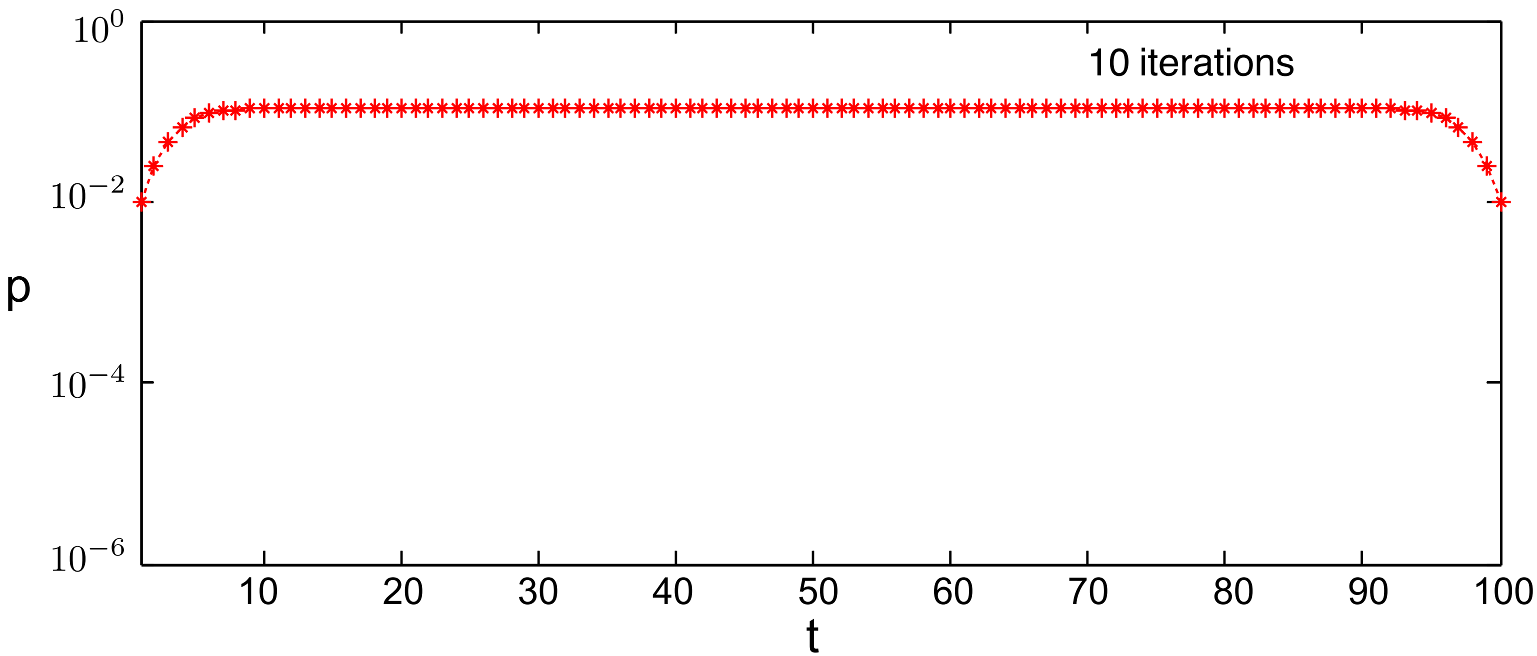
- Variable nodes all have the **same degree** as the block code.
- Check nodes with **lower degrees** (at the ends) improve the BP decoder.



Thresholds of SC-LDPC Codes

- Variable nodes all have the **same degree** as the block code.
- Check nodes with **lower degrees** (at the ends) improve the BP decoder.

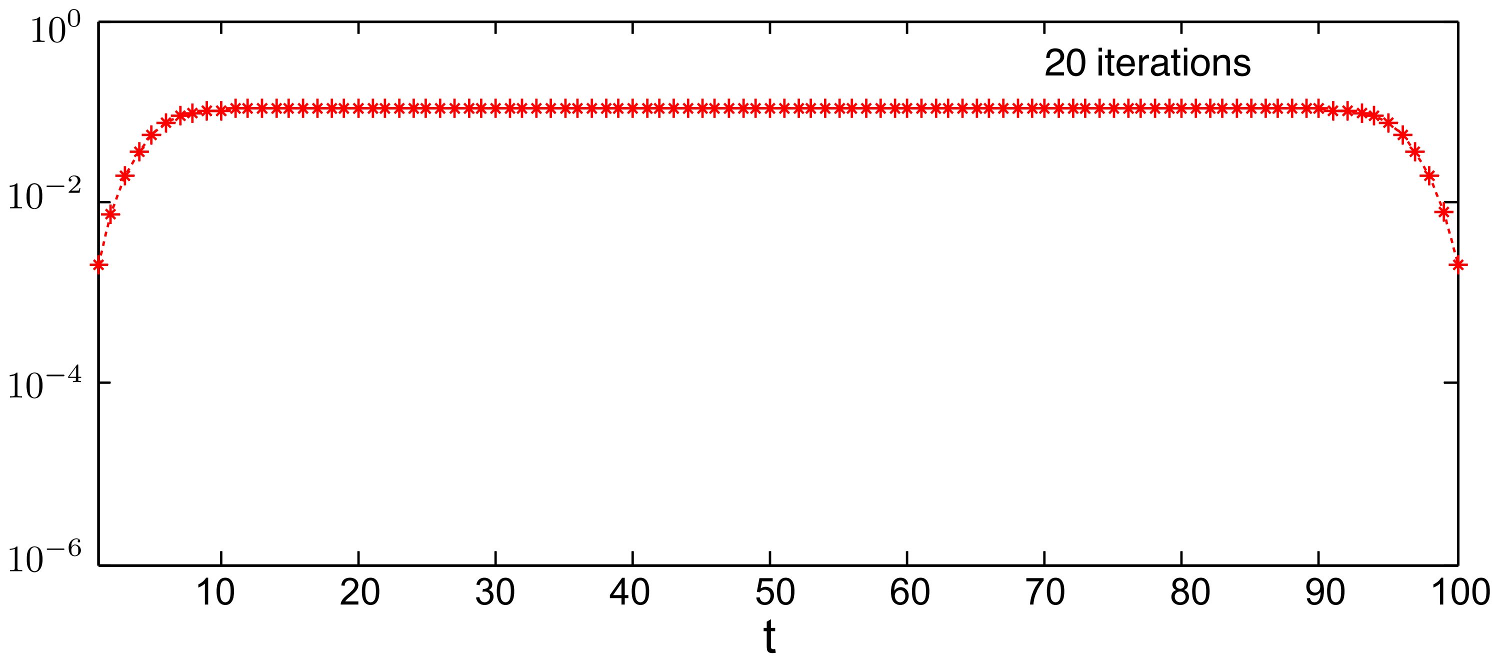
Evolution of message probabilities $(L = 100)$:



Thresholds of SC-LDPC Codes

- Variable nodes all have the **same degree** as the block code.
- Check nodes with **lower degrees** (at the ends) improve the BP decoder.

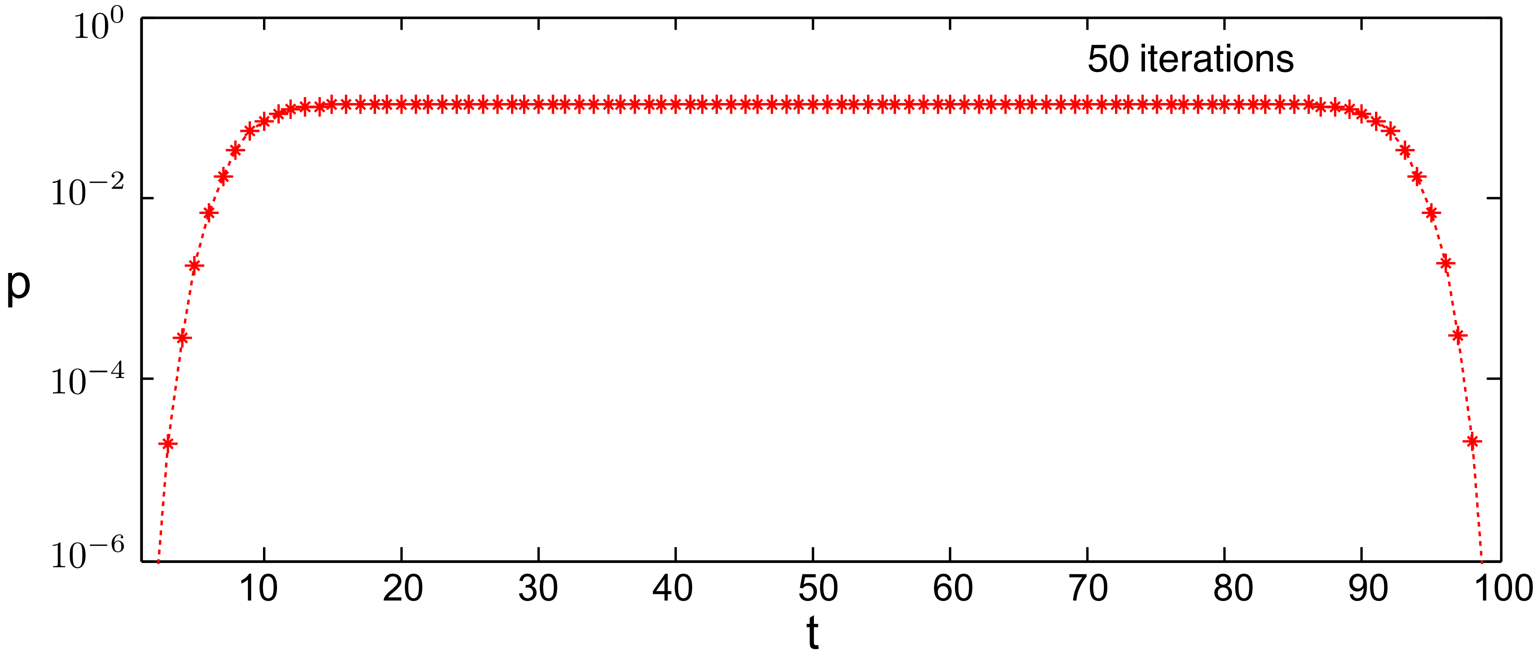
Evolution of message probabilities $(L = 100)$:



Thresholds of SC-LDPC Codes

- Variable nodes all have the **same degree** as the block code.
- Check nodes with **lower degrees** (at the ends) improve the BP decoder.

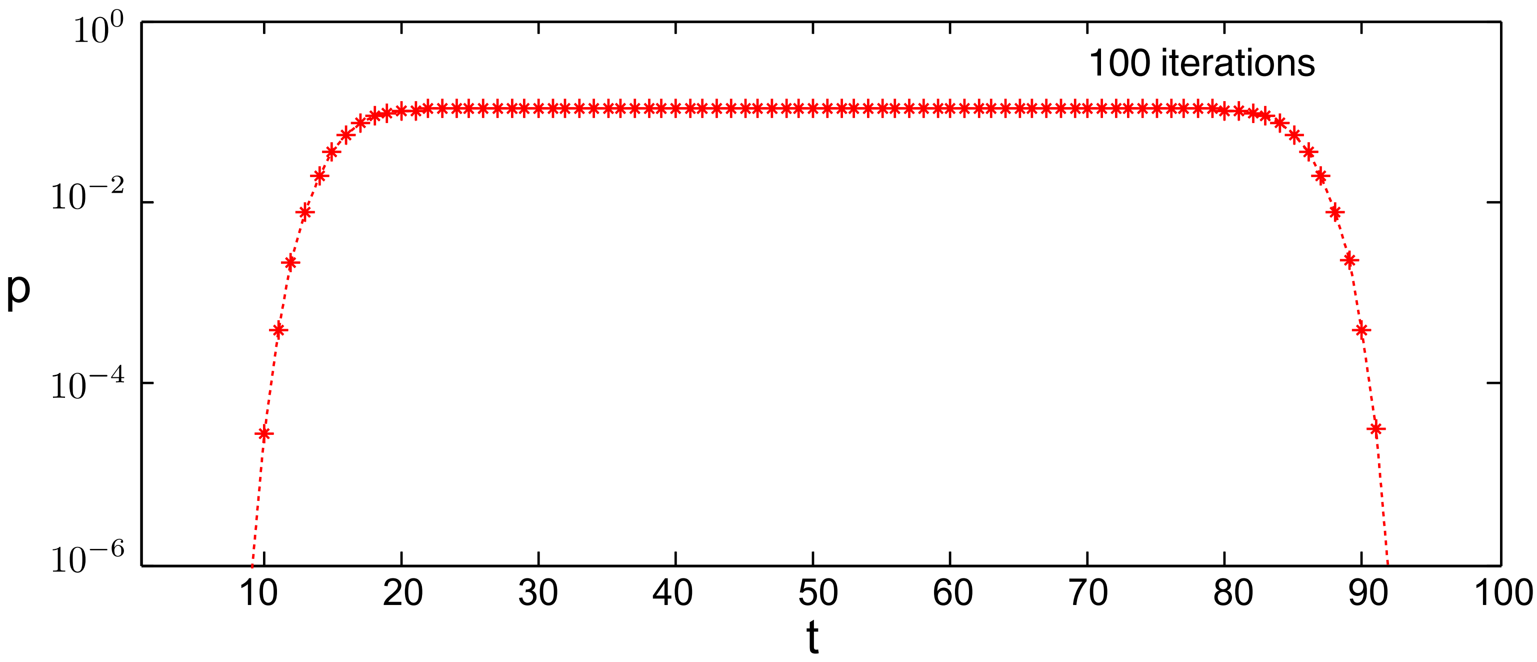
Evolution of message probabilities $(L = 100)$:



Thresholds of SC-LDPC Codes

- Variable nodes all have the **same degree** as the block code.
- Check nodes with **lower degrees** (at the ends) improve the BP decoder.

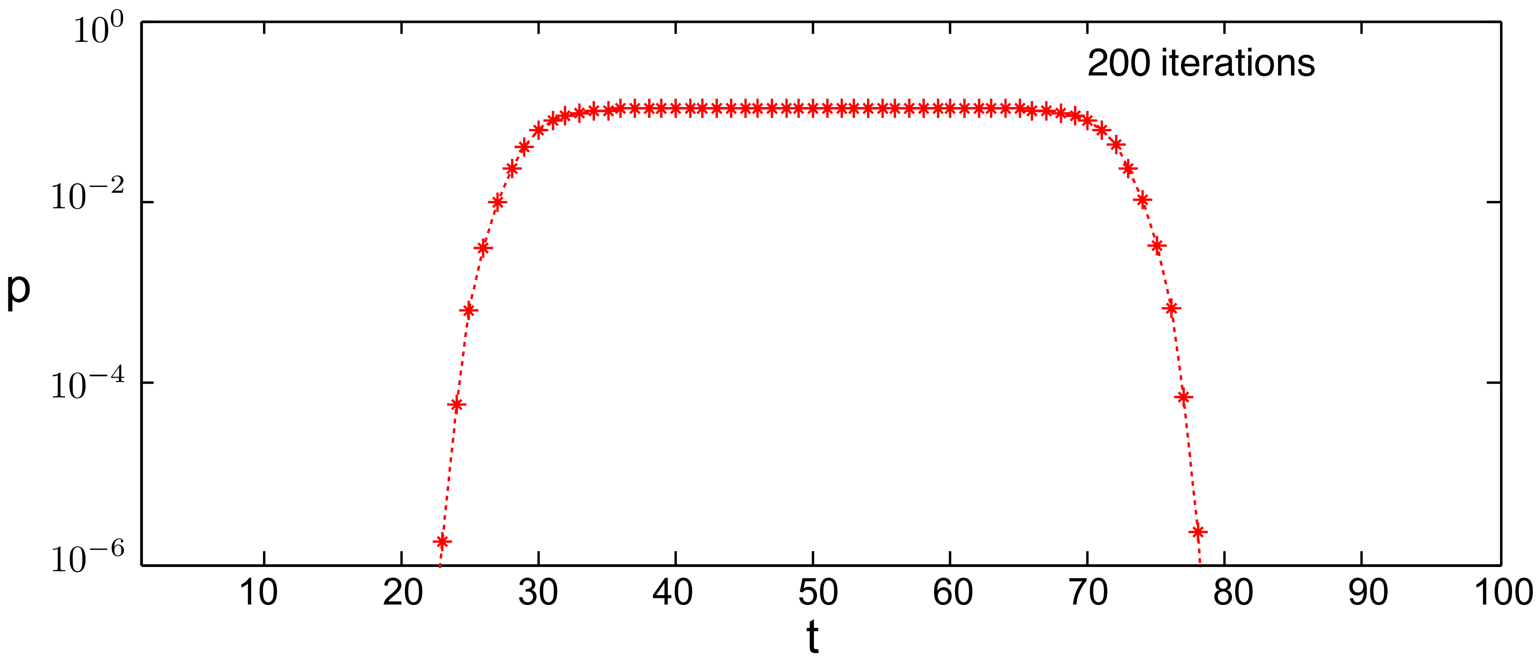
Evolution of message probabilities $(L = 100)$:



Thresholds of SC-LDPC Codes

- Variable nodes all have the **same degree** as the block code.
- Check nodes with **lower degrees** (at the ends) improve the BP decoder.

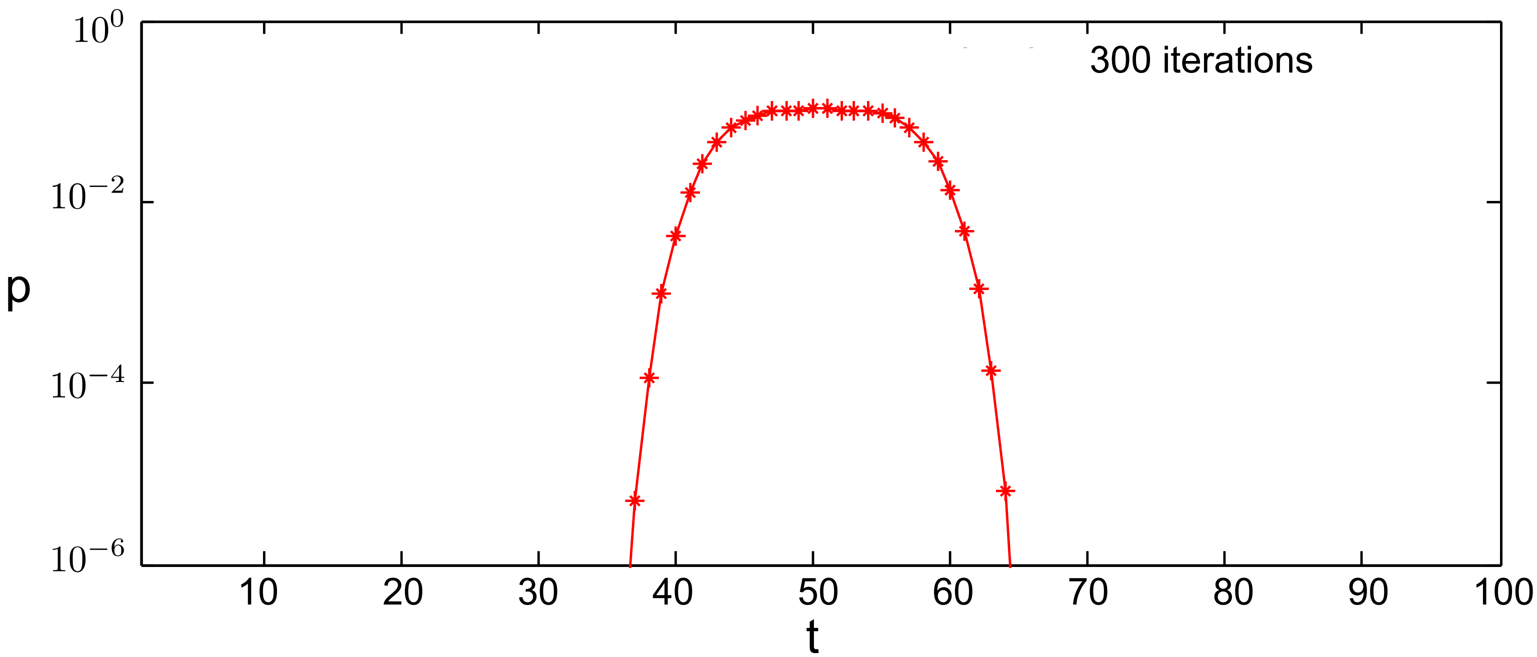
Evolution of message probabilities $(L = 100)$:



Thresholds of SC-LDPC Codes

- Variable nodes all have the **same degree** as the block code.
- Check nodes with **lower degrees** (at the ends) improve the BP decoder.

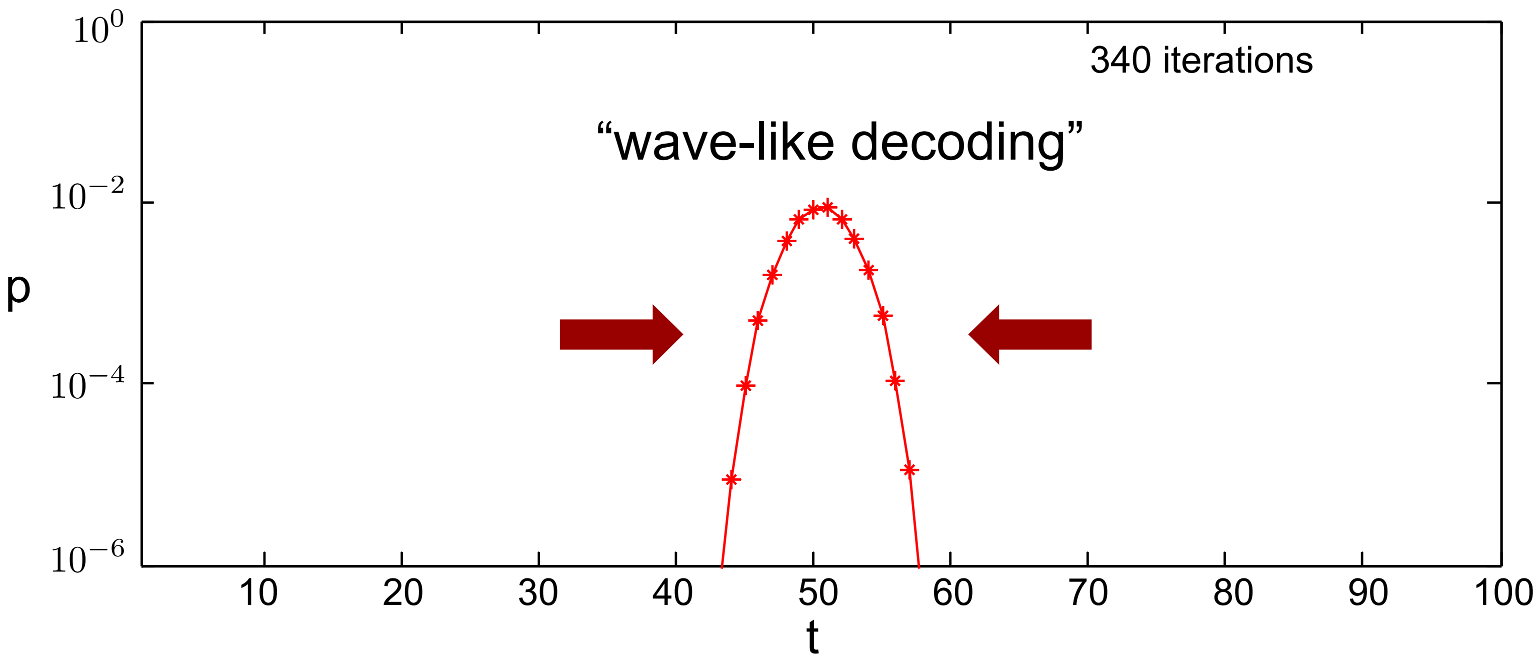
Evolution of message probabilities $(L = 100)$:



Thresholds of SC-LDPC Codes

- Variable nodes all have the **same degree** as the block code.
- Check nodes with **lower degrees** (at the ends) improve the BP decoder.

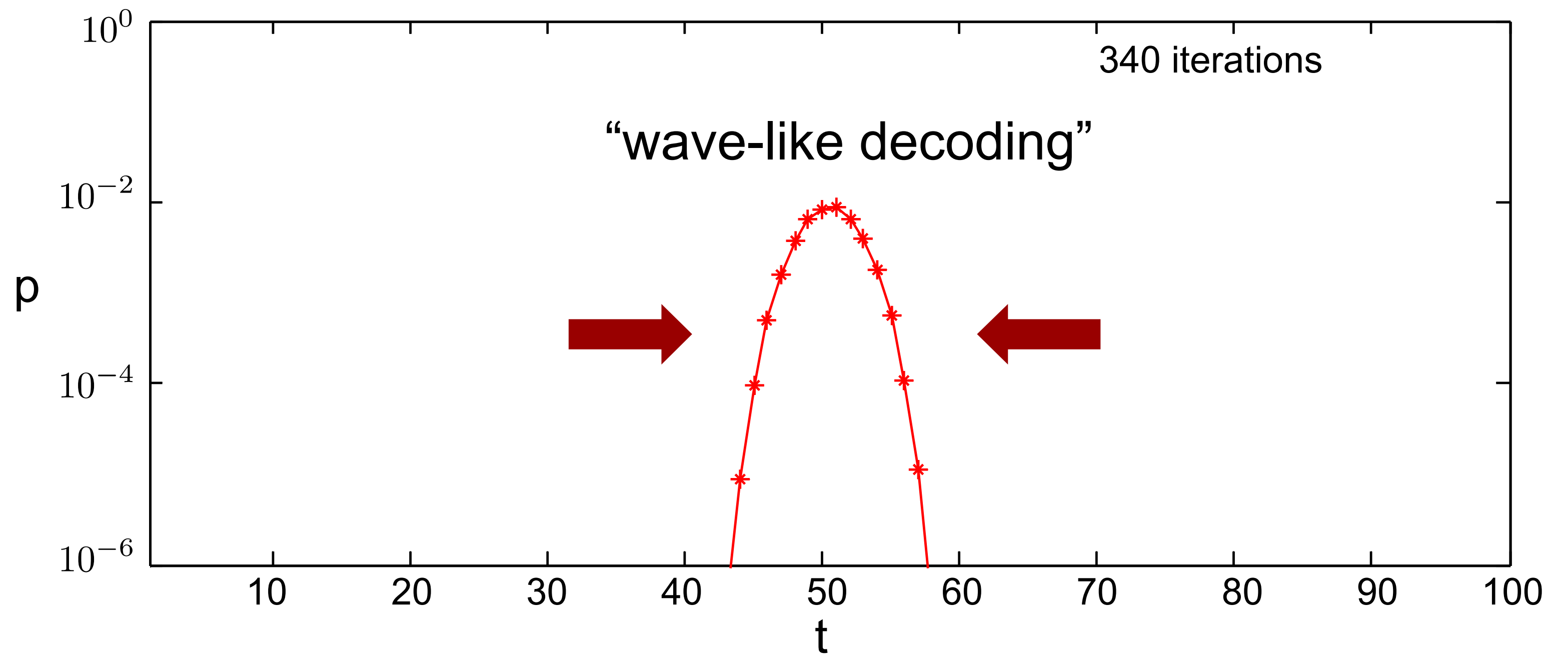
Evolution of message probabilities $(L = 100)$:



Thresholds of SC-LDPC Codes

- Variable nodes all have the **same degree** as the block code.
- Check nodes with **lower degrees** (at the ends) improve the BP decoder.

Evolution of message probabilities $(L = 100)$:

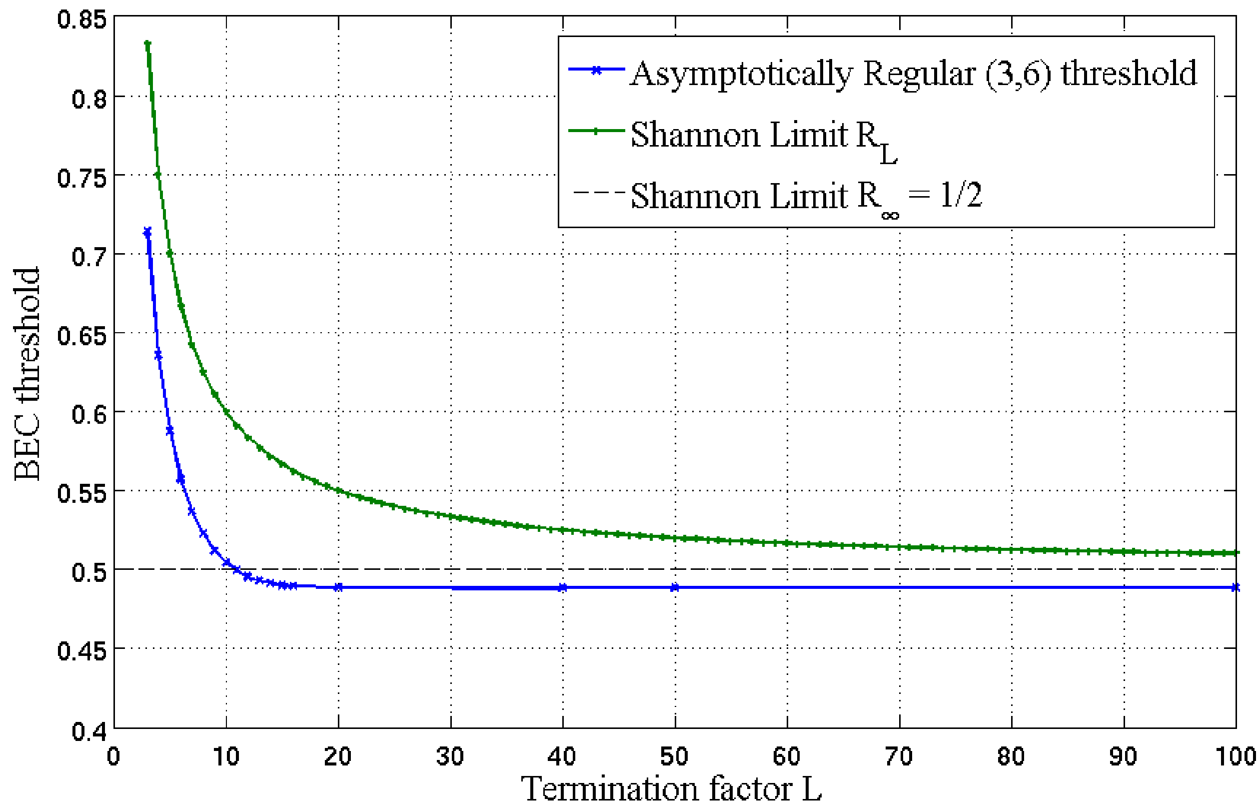


- Note: the **fraction** of **lower degree** nodes tends to zero as $L \to \infty$, i.e., the codes are **asymptotically regular**.

Thresholds of SC-LDPC Codes

- **Density evolution** can be applied to the protograph-based ensembles with $M \to \infty$ [Sridharan et al. '04]:

Example: BEC



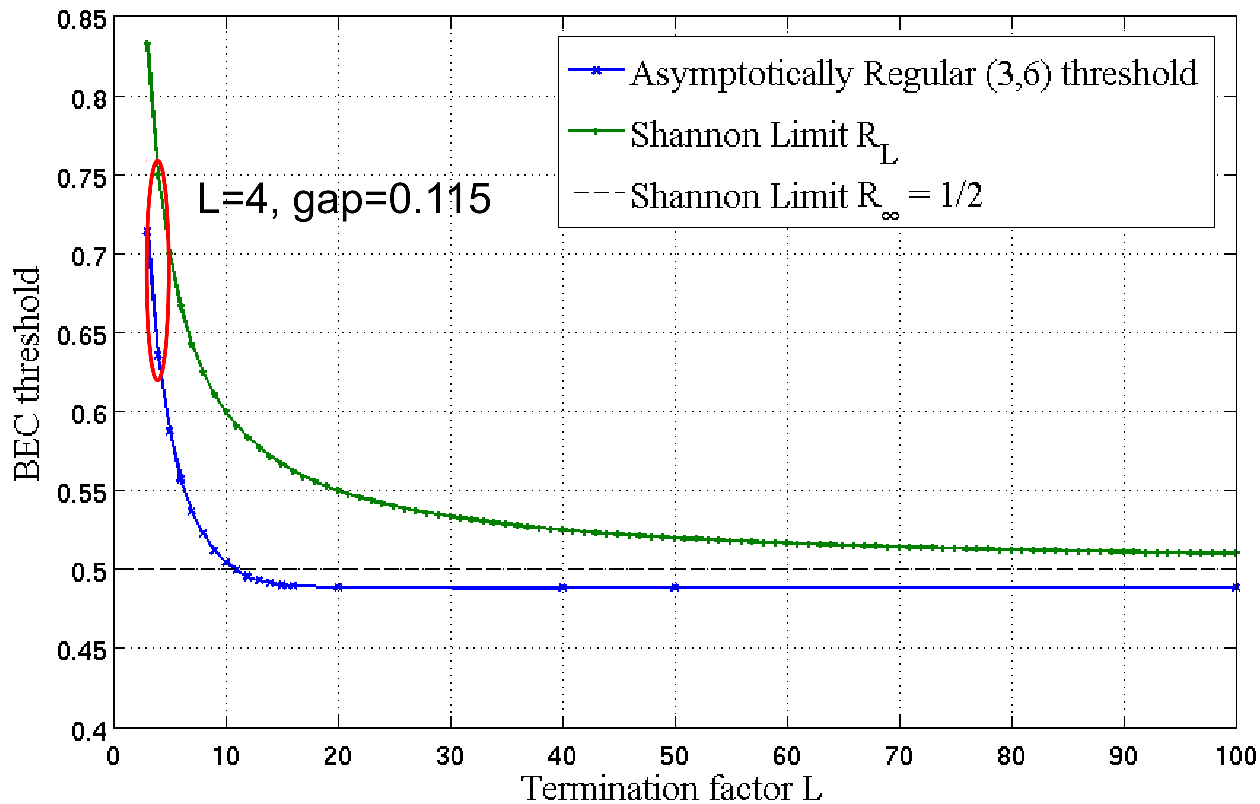
Thresholds of SC-LDPC Codes

- **Density evolution** can be applied to the protograph-based ensembles with $M \to \infty$ [Sridharan et al. '04]:

Example: BEC

$L = 4$, $R = 1/4$

$\varepsilon^* = 0.635$, $\varepsilon_{\mathrm{Sh}} = 0.75$



Thresholds of SC-LDPC Codes

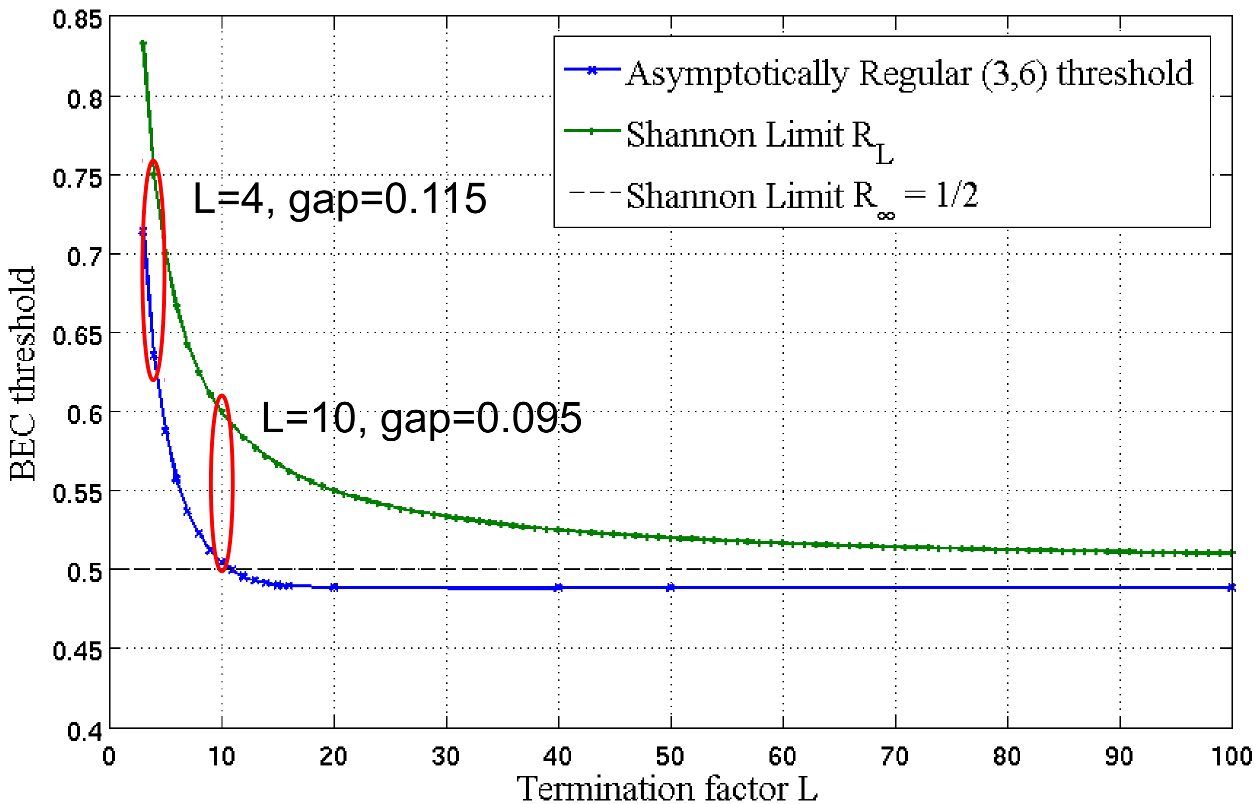
UNIVERSITY OF NOTRE DAME

- **Density evolution** can be applied to the protograph-based ensembles with $M \to \infty$ [Sridharan et al. '04]:

Example: BEC

$L = 4$, $R = 1/4$
$\varepsilon^* = 0.635$, $\varepsilon_{\text{Sh}} = 0.75$

$L = 10$, $R = 2/5$
$\varepsilon^* = 0.505$, $\varepsilon_{\text{Sh}} = 0.6$



Thresholds of SC-LDPC Codes

UNIVERSITY OF NOTRE DAME

- **Density evolution** can be applied to the protograph-based ensembles with $M \to \infty$ [Sridharan et al. '04]:

Example: BEC

$L = 4$, $R = 1/4$
$\varepsilon^* = 0.635$, $\varepsilon_{\mathrm{Sh}} = 0.75$

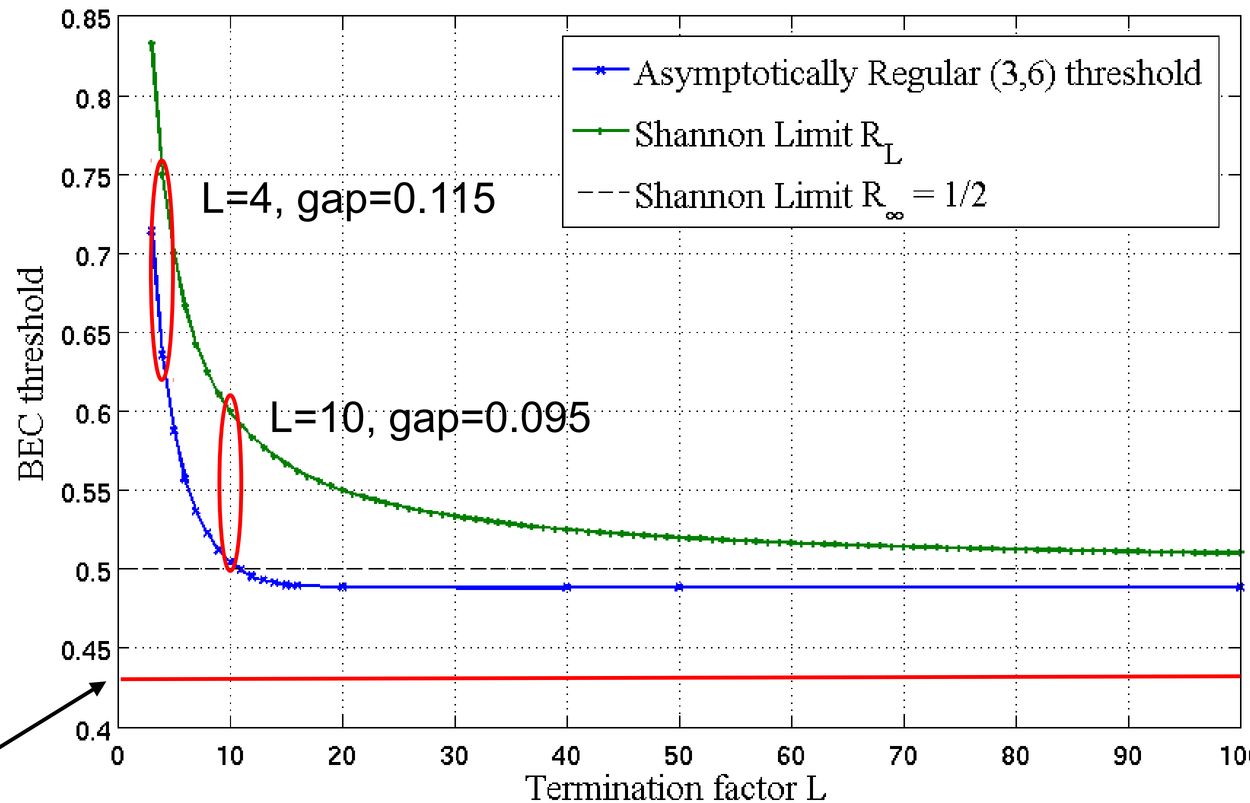
$L = 10$, $R = 2/5$
$\varepsilon^* = 0.505$, $\varepsilon_{\mathrm{Sh}} = 0.6$

⋮

$L \to \infty$, $R \to 1/2$
$\varepsilon^* = 0.488$, $\varepsilon_{\mathrm{Sh}} = 0.5$

(3,6)-regular block code:

$\varepsilon^* = 0.429$



Thresholds of SC-LDPC Codes

UNIVERSITY OF NOTRE DAME

Iterative decoding thresholds (protograph-based ensembles)

BEC

(J,K)	ϵ^*_{SC}	$\epsilon^*_{\mathrm{blk}}$
(3,6)	0.488	0.429
(4,8)	0.497	0.383
(5,10)	0.499	0.341

AWGN

(J,K)	E_b/N_o SC	E_b/N_o blk
(3,6)	0.46 dB	1.11 dB
(4,8)	0.26 dB	1.61 dB
(5,10)	0.21 dB	2.04 dB

- We observe a **significant improvement** in the thresholds of SC-LDPC codes compared to the associated LDPC block codes (LDPC-BCs) due to the lower degree check nodes at the ends of the graph and the wave-like decoding.

[LSCZ10] M. Lentmaier, A. Sridharan, D. J. Costello, Jr., and K.Sh. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes," *IEEE Trans. Inf. Theory*, 56:10, Oct. 2010.

Thresholds of SC-LDPC Codes

UNIVERSITY OF NOTRE DAME

Iterative decoding thresholds (protograph-based ensembles)

BEC

(J, K)	ϵ^*_{SC}	$\epsilon^*_{\mathrm{blk}}$
(3,6)	0.488	0.429
(4,8)	0.497	0.383
(5,10)	0.499	0.341

AWGN

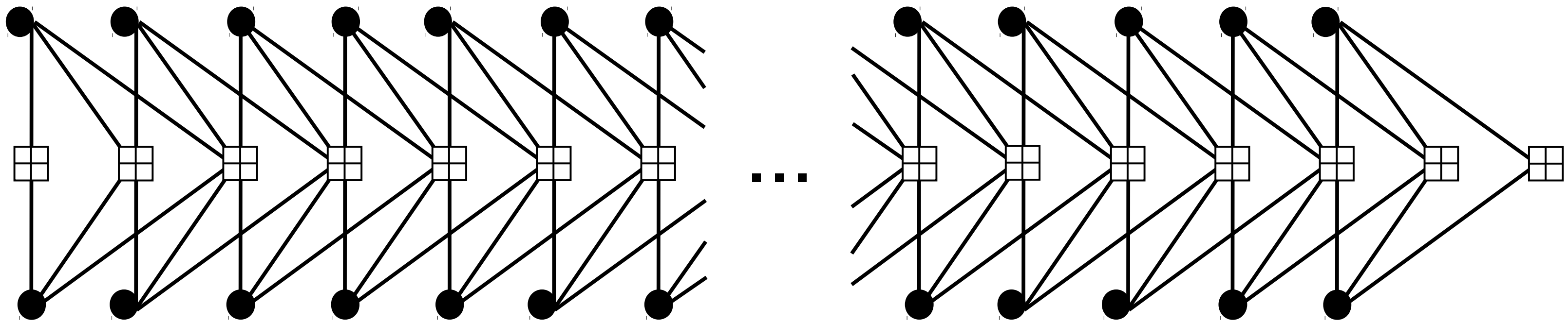
(J, K)	$E_b/N_{o\,\mathrm{SC}}$	$E_b/N_{o\,\mathrm{blk}}$
(3,6)	0.46 dB	1.11 dB
(4,8)	0.26 dB	1.61 dB
(5,10)	0.21 dB	2.04 dB

- We observe a **significant improvement** in the thresholds of SC-LDPC codes compared to the associated LDPC block codes (LDPC-BCs) due to the lower degree check nodes at the ends of the graph and the wave-like decoding.
- In contrast to LDPC-BCs, the iterative decoding thresholds of SC-LDPC codes **improve** as the graph density increases.

[LSCZ10] M. Lentmaier, A. Sridharan, D. J. Costello, Jr., and K.Sh. Zigangirov, “Iterative decoding threshold analysis for LDPC convolutional codes,” *IEEE Trans. Inf. Theory*, 56:10, Oct. 2010.

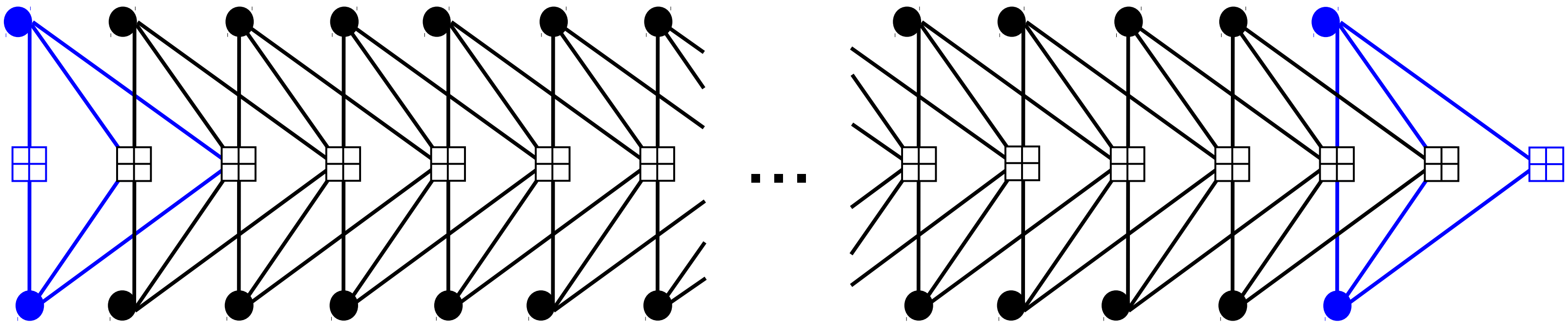
Why are SC-LDPC Codes Better?

- When symbols are perfectly known (BEC), all adjacent edges can be removed from the Tanner graph.



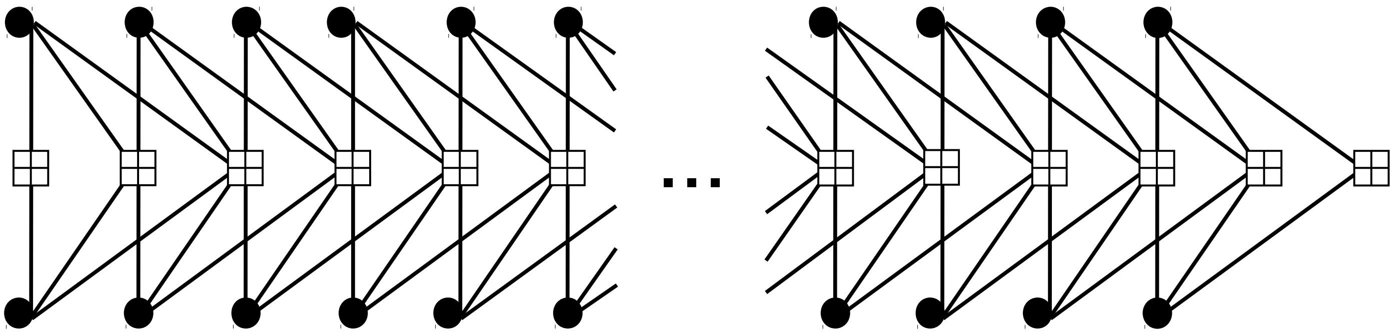
Why are SC-LDPC Codes Better?

- When symbols are perfectly known (BEC), all adjacent edges can be removed from the Tanner graph.



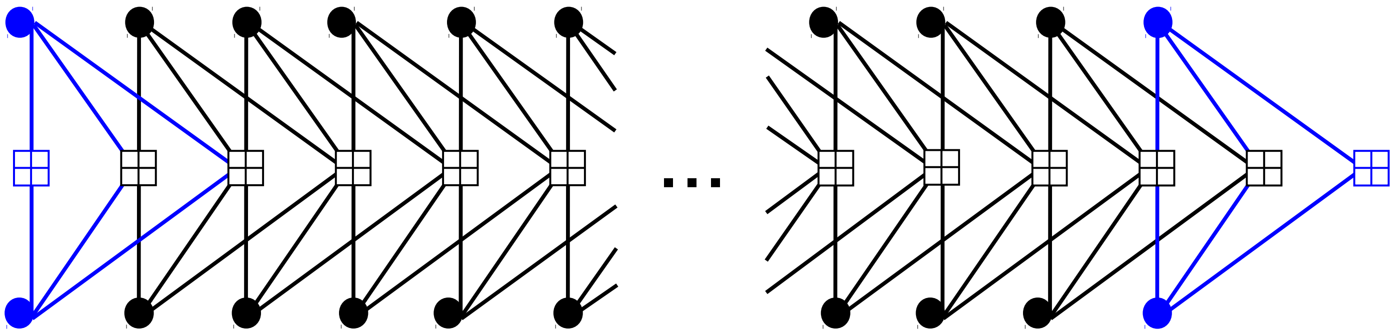
Why are SC-LDPC Codes Better?

- When symbols are perfectly known (BEC), all adjacent edges can be removed from the Tanner graph.



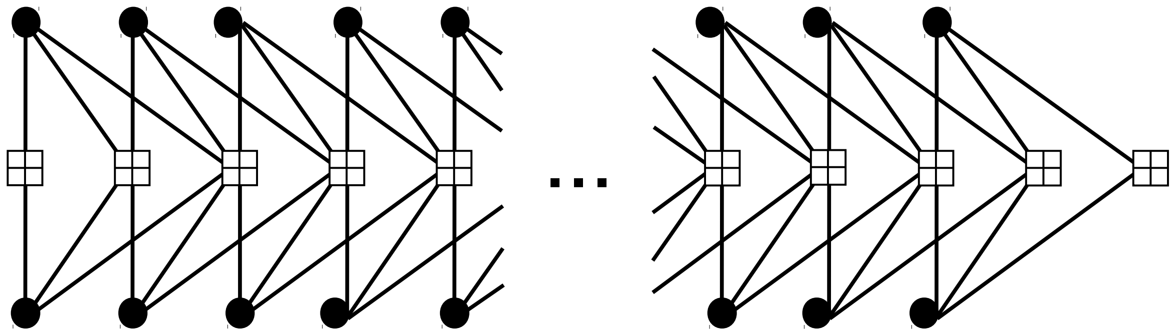
Why are SC-LDPC Codes Better?

- When symbols are perfectly known (BEC), all adjacent edges can be removed from the Tanner graph.



Why are SC-LDPC Codes Better?

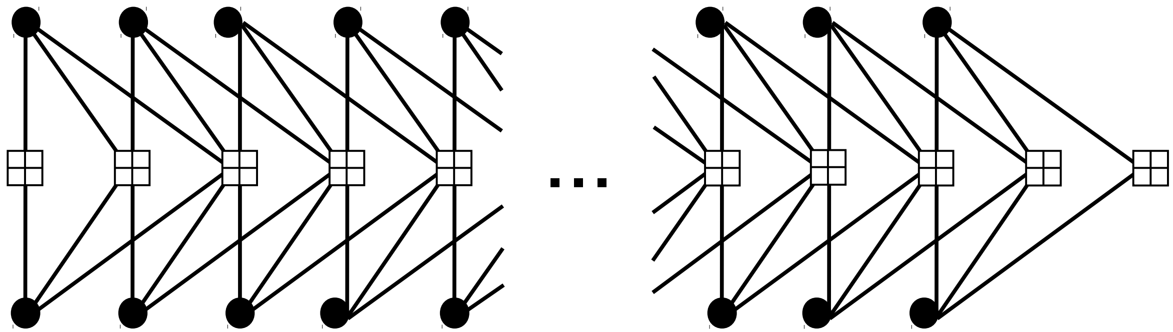
- When symbols are perfectly known (BEC), all adjacent edges can be removed from the Tanner graph.



Why are SC-LDPC Codes Better?

UNIVERSITY OF NOTRE DAME

- When symbols are perfectly known (BEC), all adjacent edges can be removed from the Tanner graph.

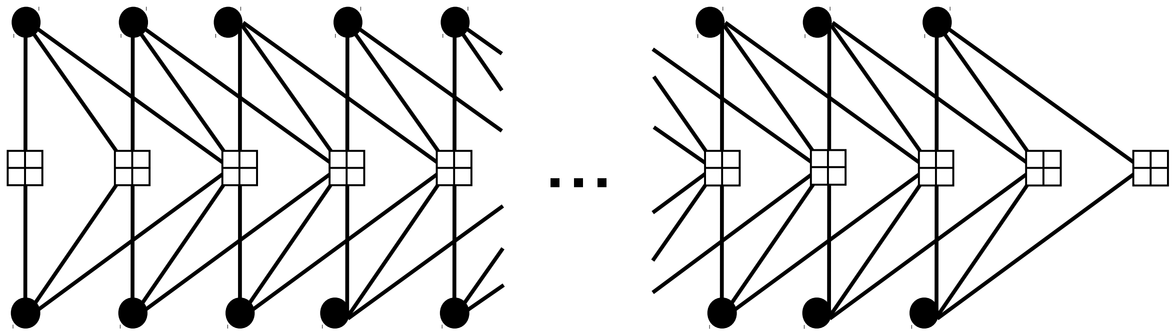


- The **threshold saturates** (converges) to a fixed value numerically indistinguishable from the **maximum a posteriori** (MAP) threshold of the (J, K)-regular LDPC-BC ensemble as $L \to \infty$ [LSCZ10].

[LSCZ10] M. Lentmaier, A. Sridharan, D. J. Costello, Jr., and K.Sh. Zigangirov, “Iterative decoding threshold analysis for LDPC convolutional codes,” *IEEE Trans. Inf. Theory*, 56:10, Oct. 2010.

Why are SC-LDPC Codes Better?

- When symbols are perfectly known (BEC), all adjacent edges can be removed from the Tanner graph.



- The **threshold saturates** (converges) to a fixed value numerically indistinguishable from the **maximum a posteriori** (MAP) threshold of the (J, K)-regular LDPC-BC ensemble as $L \to \infty$ [LSCZ10].
- For a more random-like ensemble, this has been proven analytically, first for the BEC [KRU11], then for all BMS channels [KRU13].

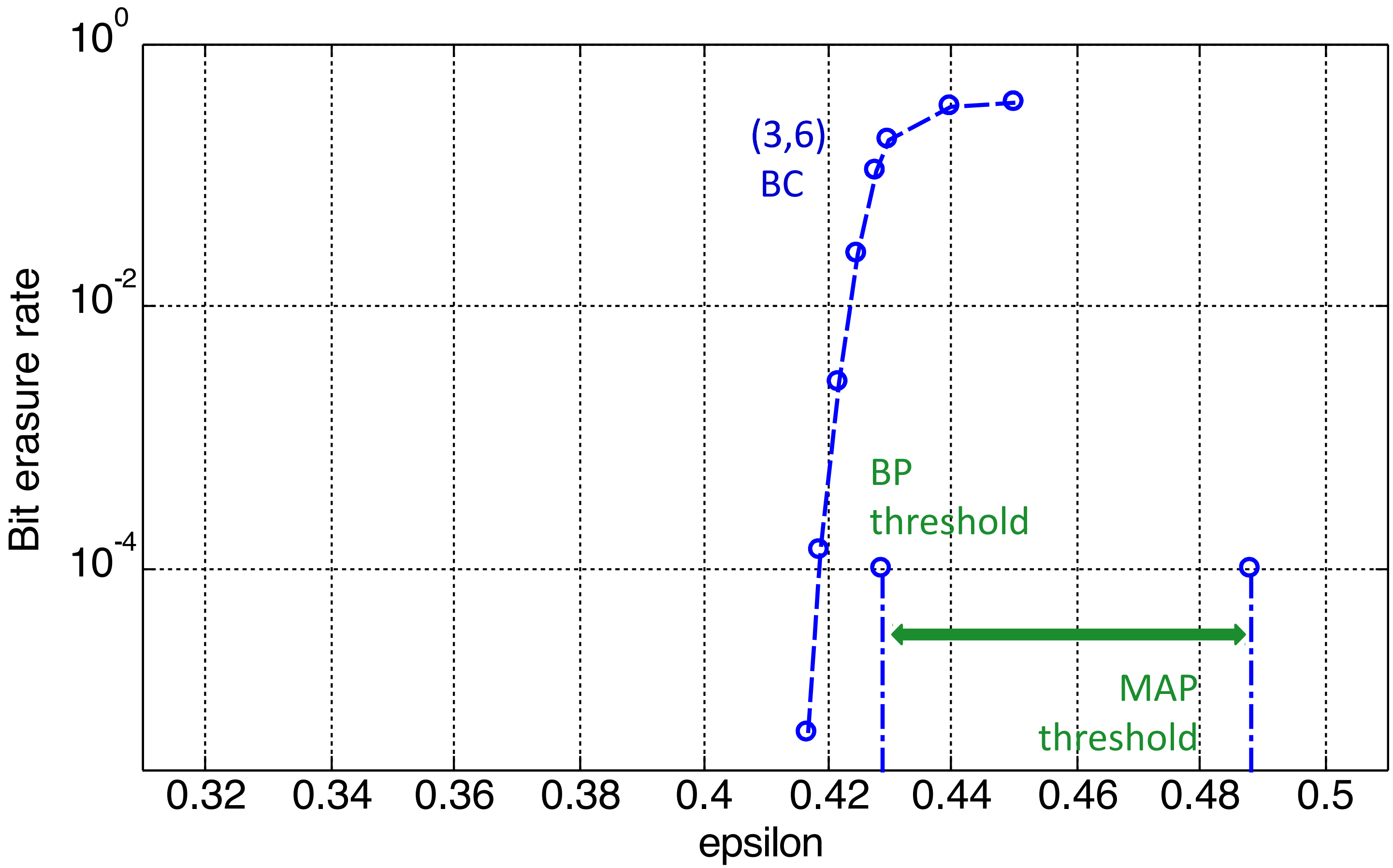
[LSCZ10] M. Lentmaier, A. Sridharan, D. J. Costello, Jr., and K.Sh. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes," *IEEE Trans. Inf. Theory*, 56:10, Oct. 2010.

[KRU11] S. Kudekar, T. J. Richardson and R. Urbanke, "Threshold saturation via spatial coupling: why convolutional LDPC ensembles perform so well over the BEC", *IEEE Trans. on Inf. Theory*, 57:2, 2011

[KRU13] S. Kudekar, T. J. Richardson and R. Urbanke, "Spatially coupled ensembles universally achieve capacity under belief propagation", *IEEE Trans. on Inf. Theory*, 59:12, 2013.

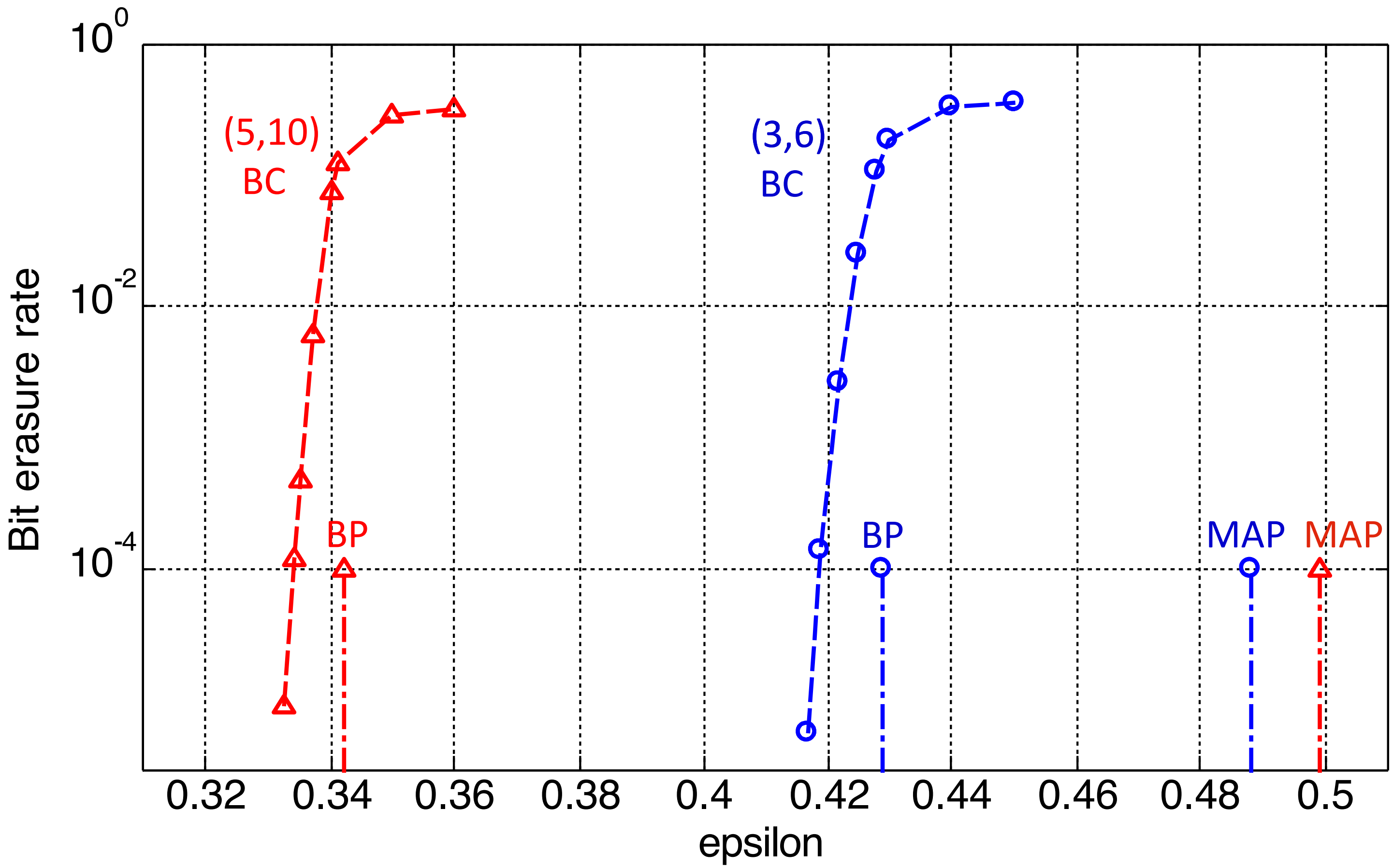
Threshold Saturation (BEC)

BP = iterative (suboptimal) decoding threshold
MAP = (optimal) maximum a posteriori threshold



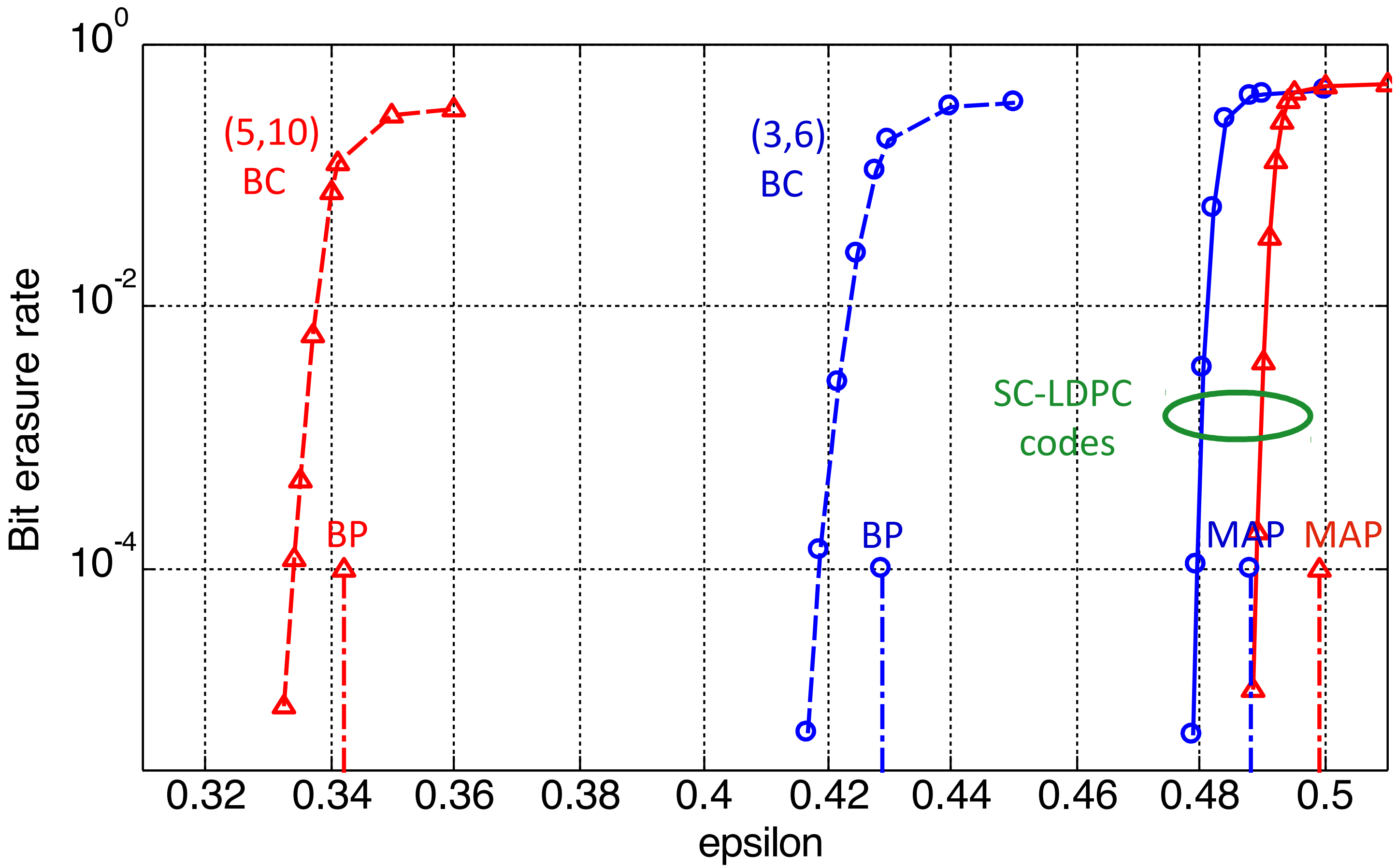
Threshold Saturation (BEC)

BP = iterative (suboptimal) decoding threshold
MAP = (optimal) maximum a posteriori threshold



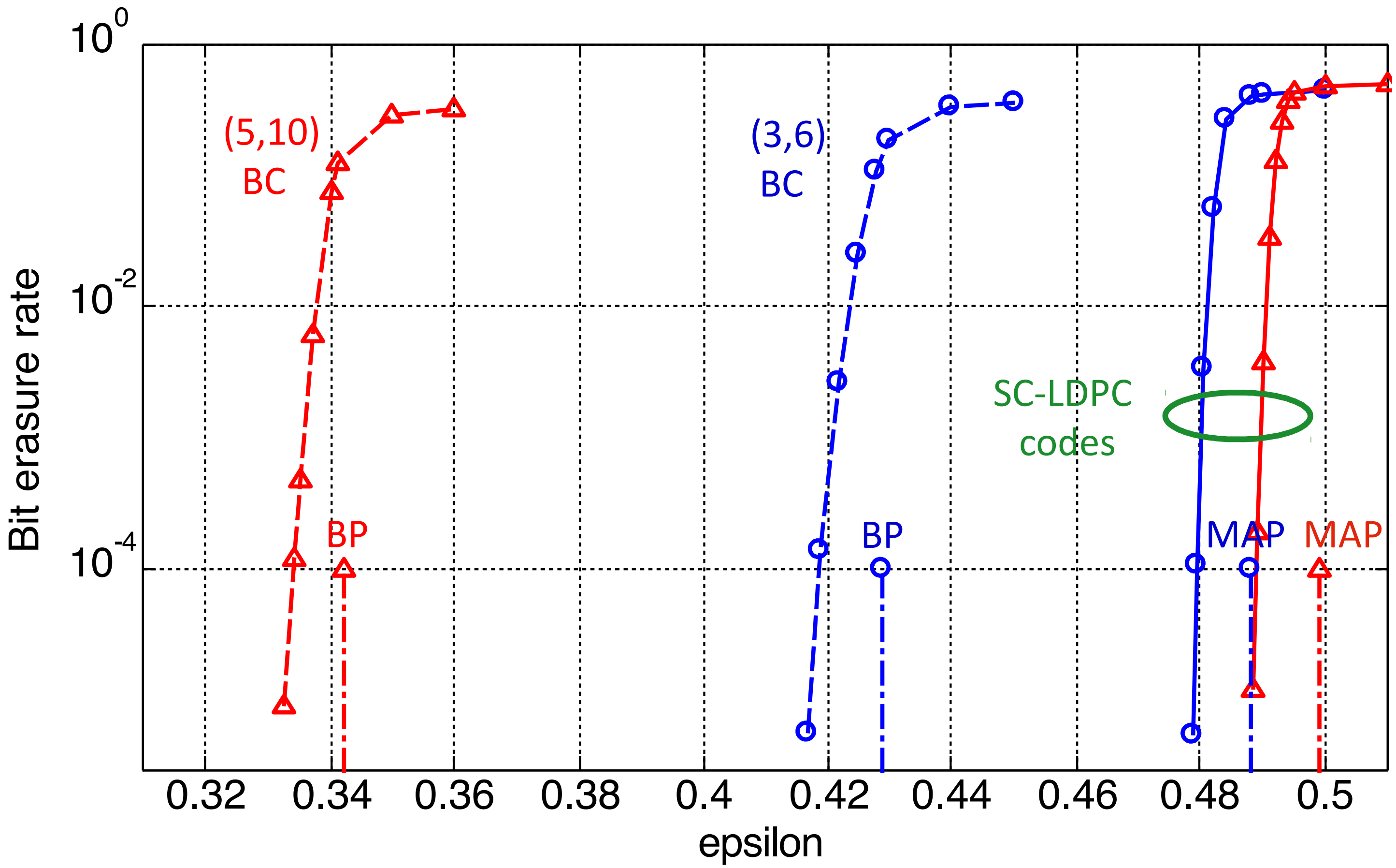
Threshold Saturation (BEC)

BP = iterative (suboptimal) decoding threshold
MAP = (optimal) maximum a posteriori threshold



Threshold Saturation (BEC)

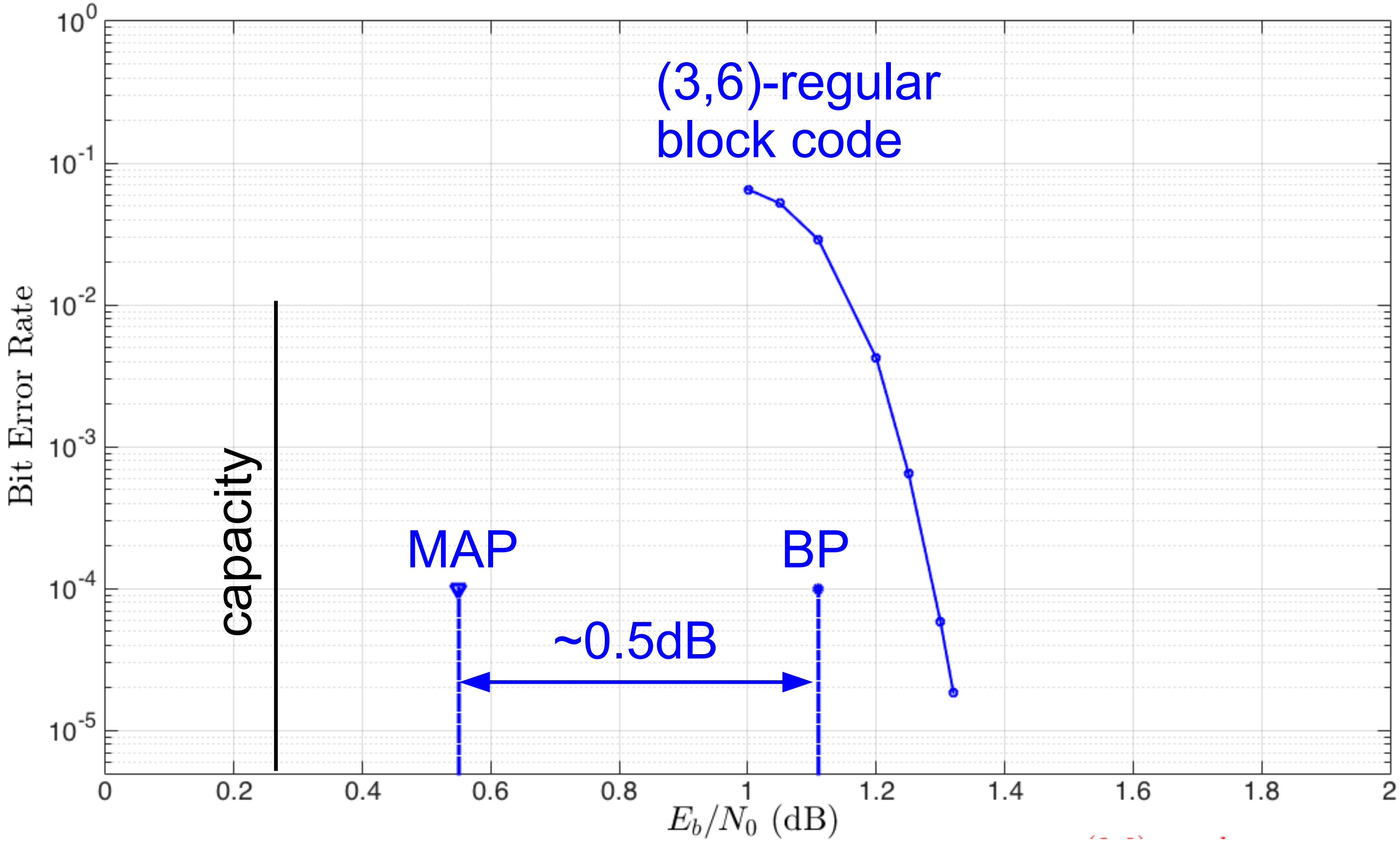
BP = iterative (suboptimal) decoding threshold
MAP = (optimal) maximum a posteriori threshold



➡ **optimal** decoding performance with a **suboptimal** iterative algorithm!

Threshold Saturation (AWGNC)

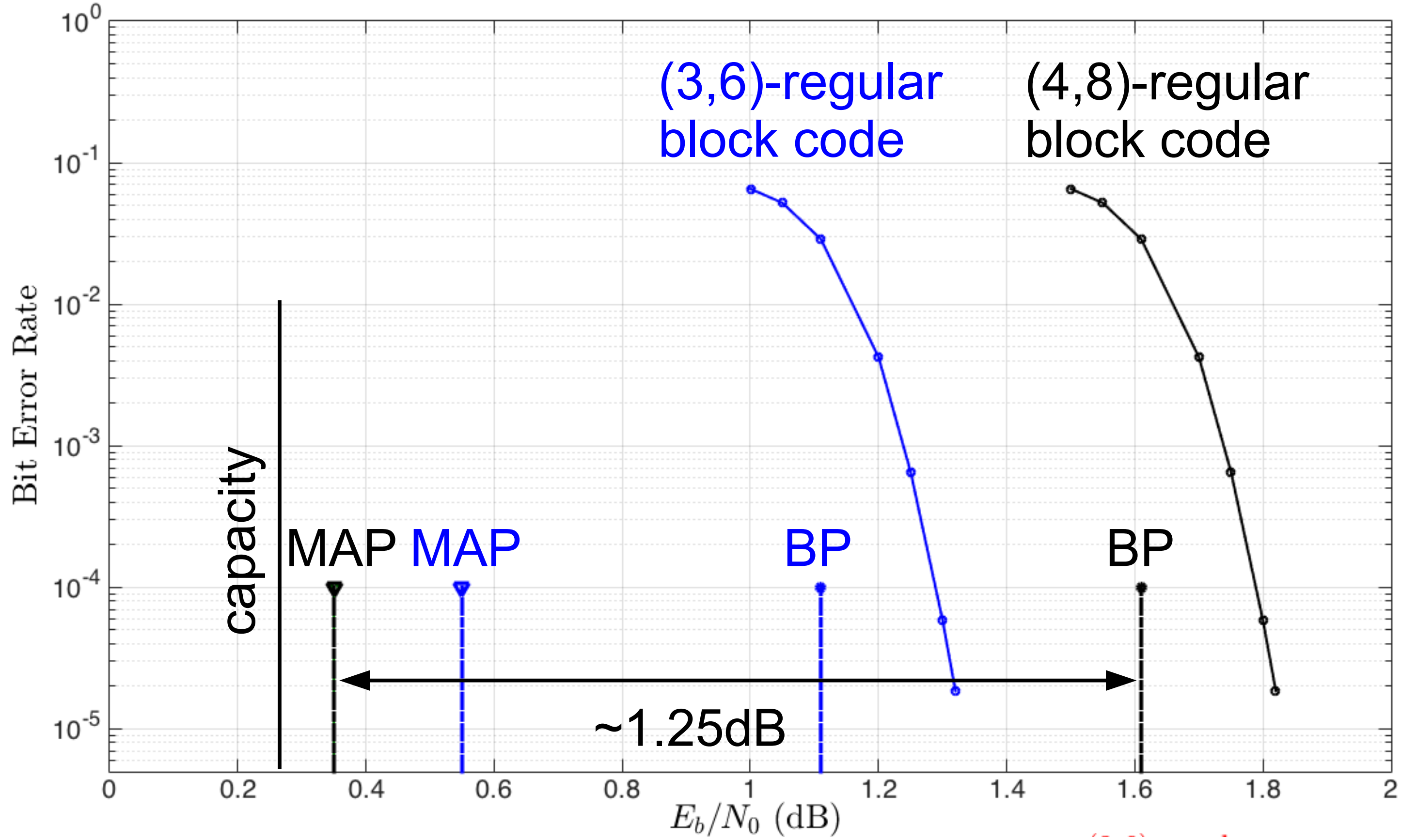
BP = iterative (suboptimal) decoding threshold
MAP = (optimal) maximum a posteriori threshold



Threshold Saturation (AWGNC)

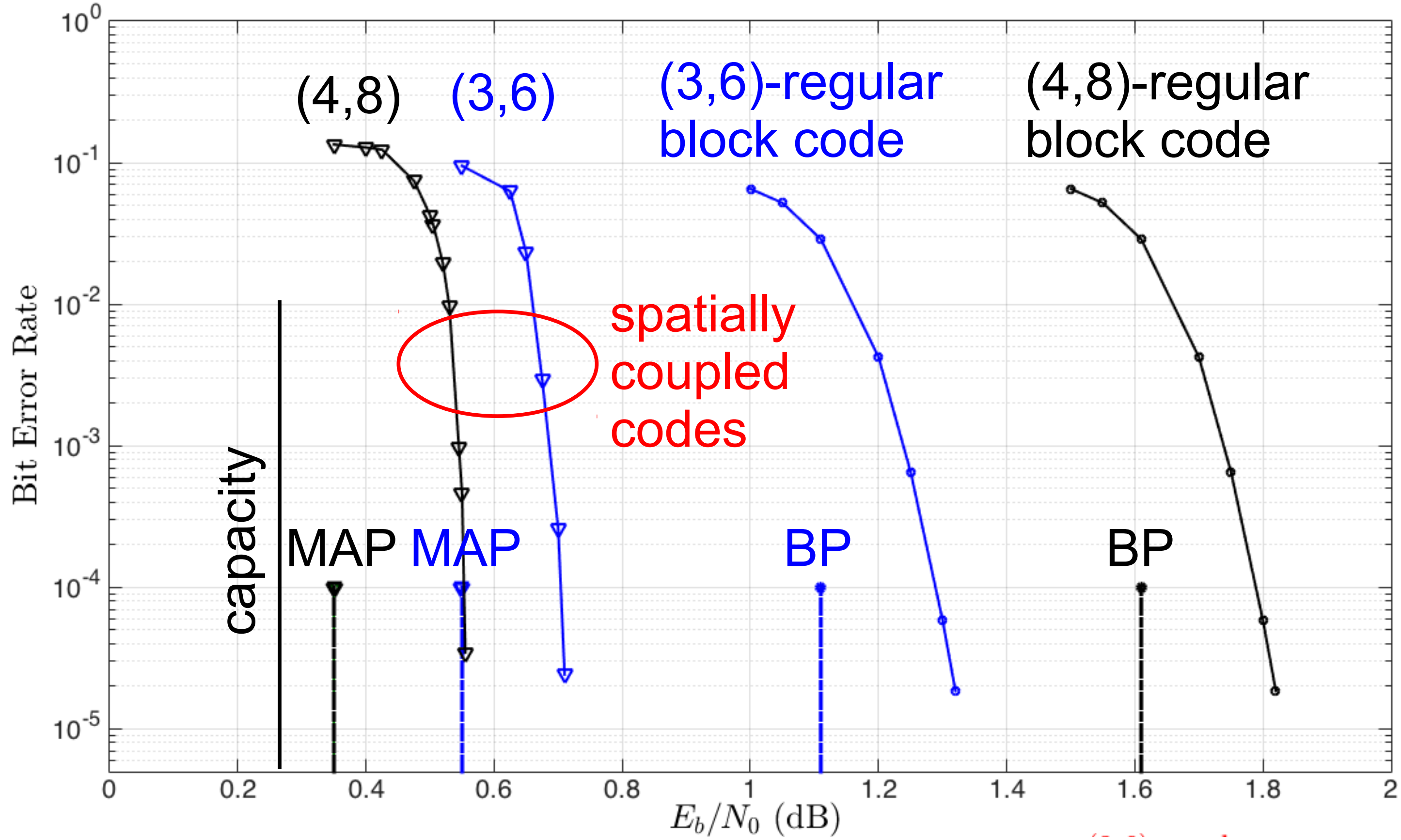
UNIVERSITY OF NOTRE DAME

BP = iterative (suboptimal) decoding threshold
MAP = (optimal) maximum a posteriori threshold



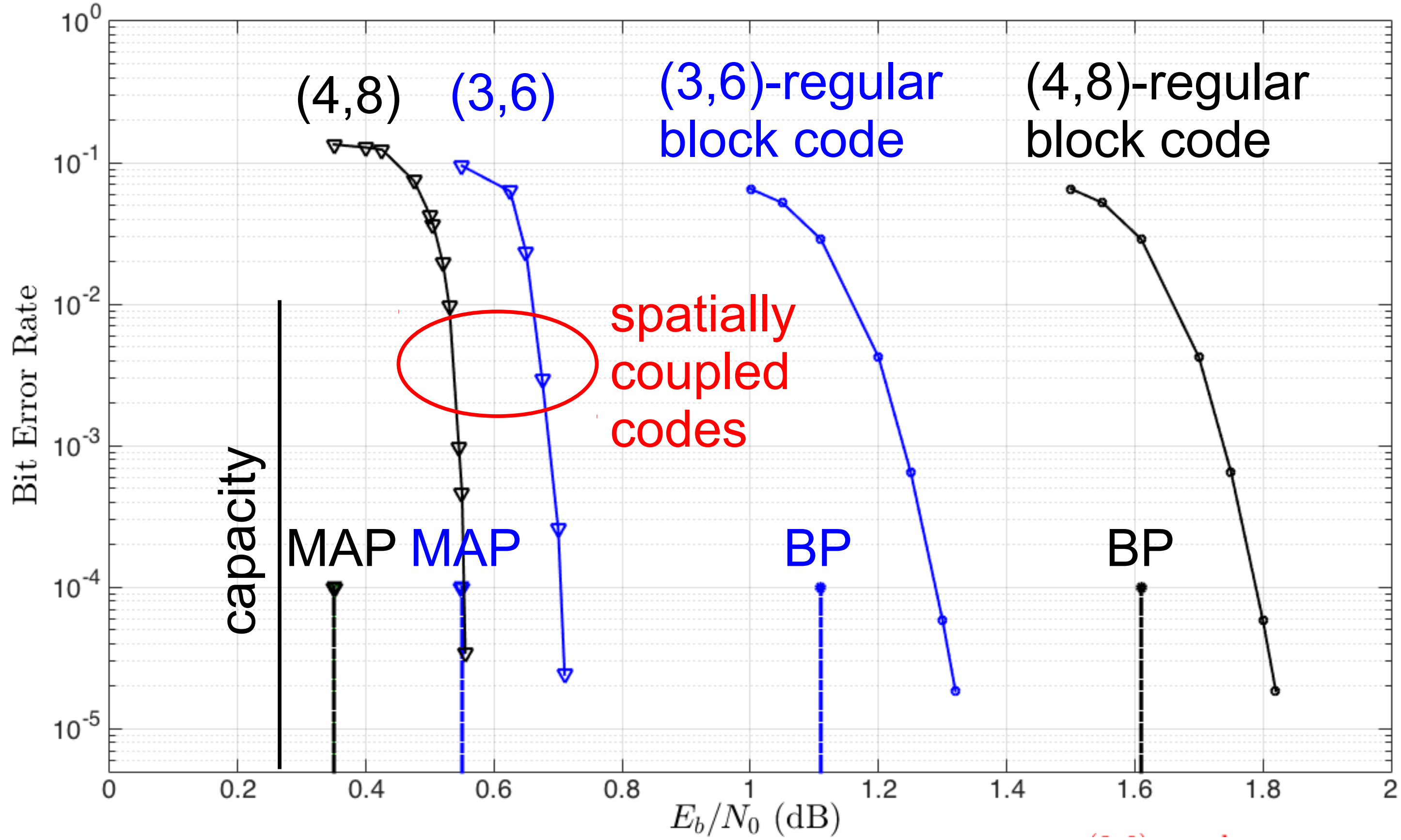
Threshold Saturation (AWGNC)

BP = iterative (suboptimal) decoding threshold
MAP = (optimal) maximum a posteriori threshold



Threshold Saturation (AWGNC)

BP = iterative (suboptimal) decoding threshold
MAP = (optimal) maximum a posteriori threshold

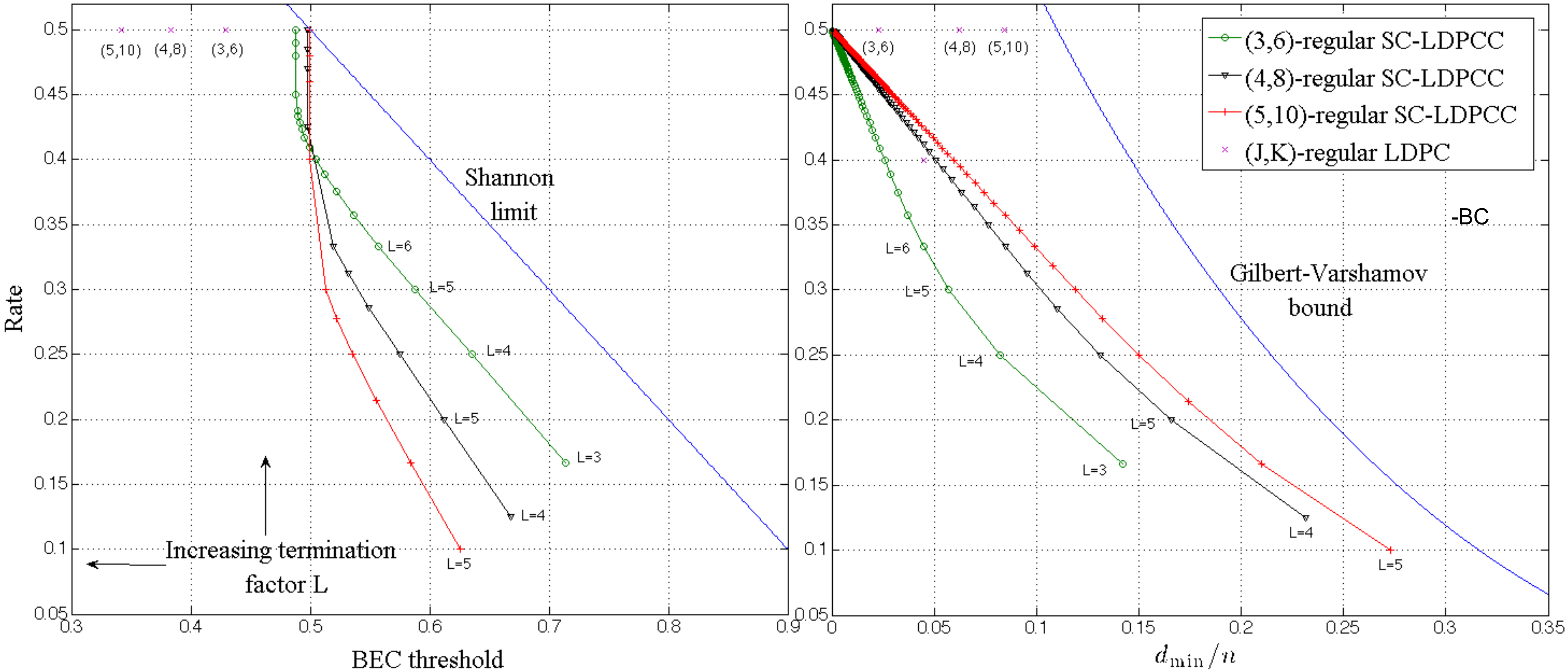


➡ **optimal** decoding performance with a **suboptimal** iterative algorithm!

BEC Thresholds vs Distance Growth

UNIVERSITY OF NOTRE DAME

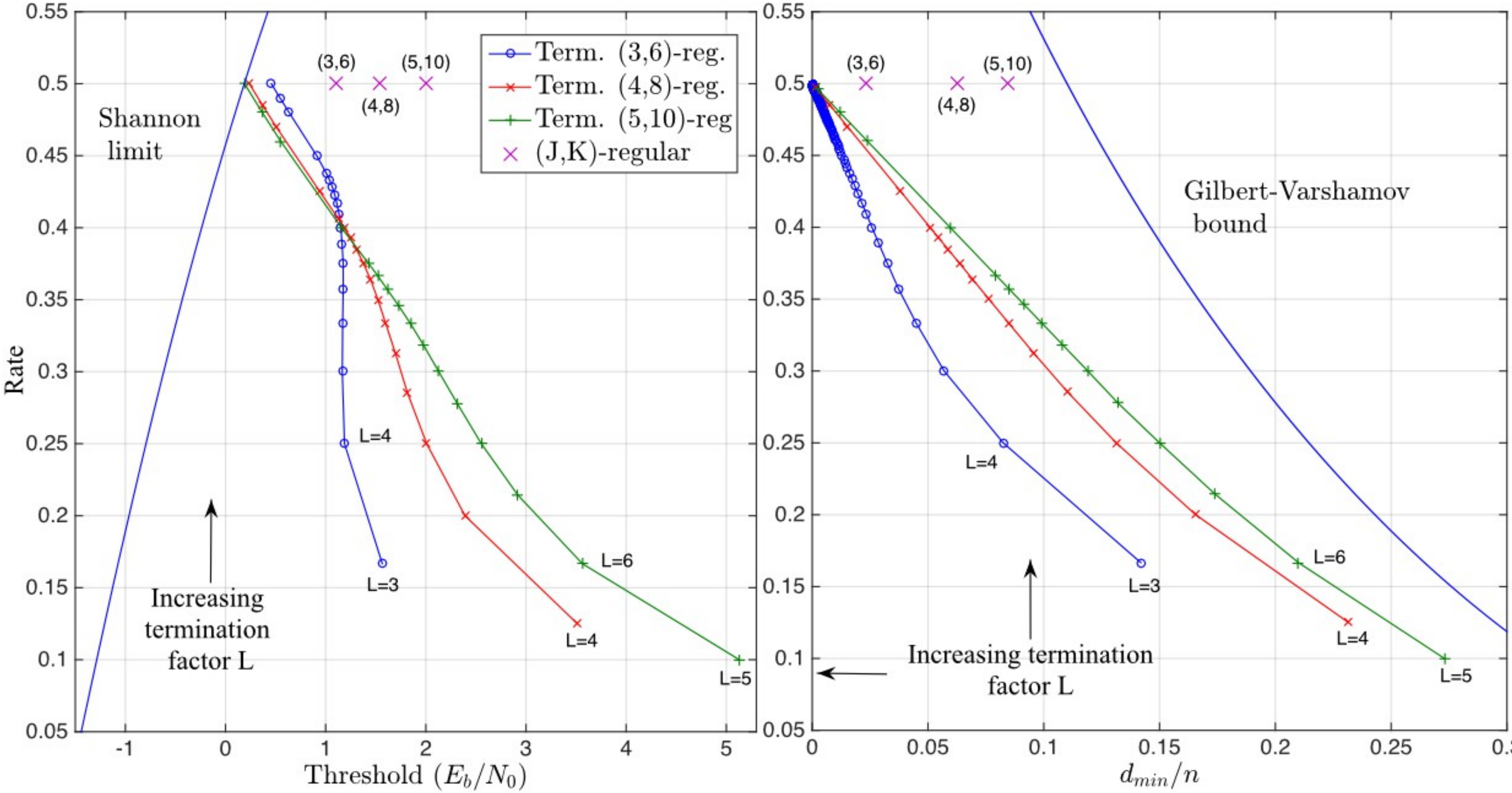
- By increasing J and K, we obtain **capacity achieving** (J,K)-regular SC-LDPC code ensembles with linear minimum distance growth.



- (J,K)-regular SC-LDPC codes combine the best features of irregular and regular LDPC-BCs, i.e., capacity approaching thresholds and linear distance growth.

AWGNC Thresholds vs. Distance Growth

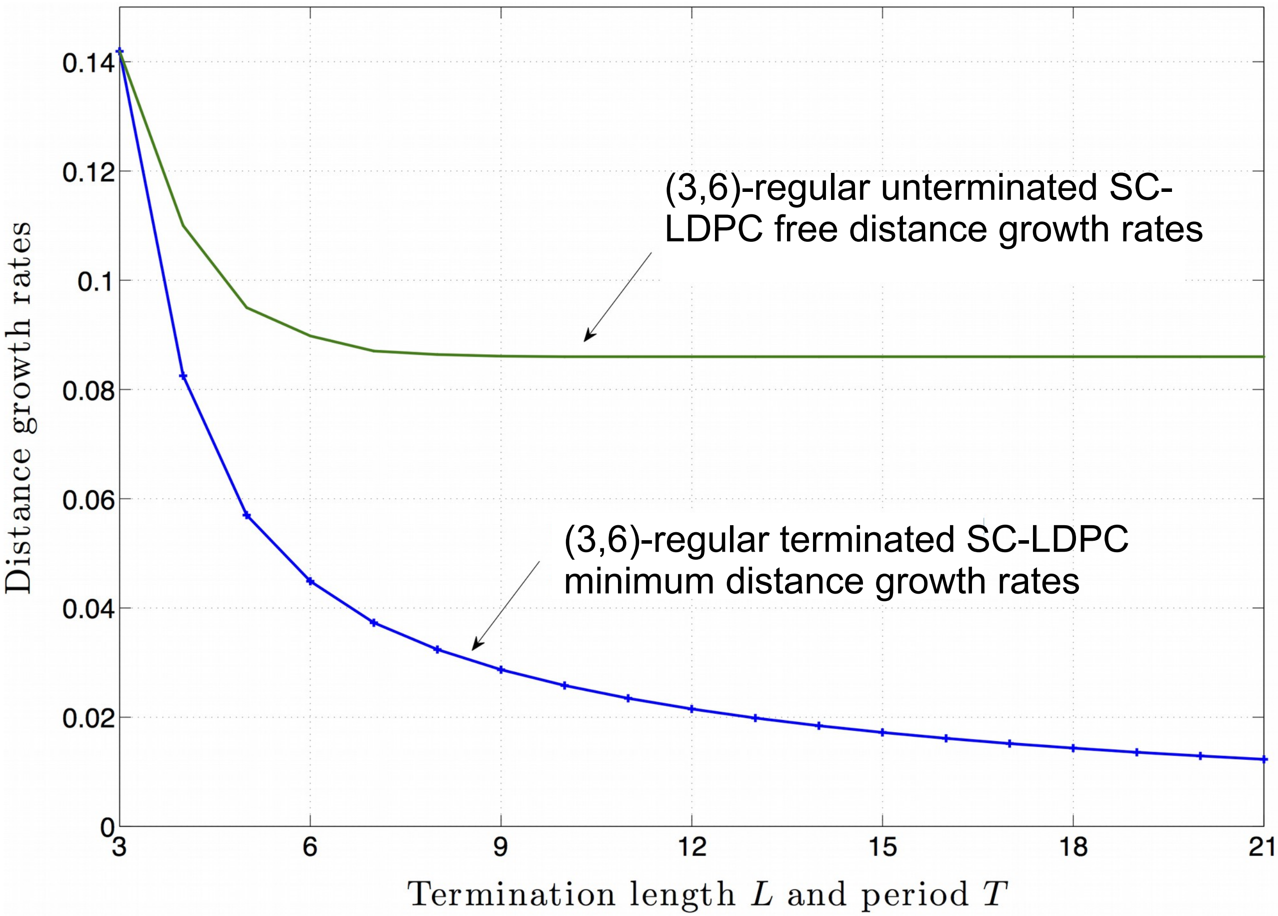
- Similar results are obtained for the AWGNC



[MLC10] D. G. M. Mitchell, M. Lentmaier and D. J. Costello, Jr., “AWGN Channel Analysis of Terminated LDPC Convolutional Codes”, *Proc. Information Theory and Applications Workshop*, San Diego, Feb. 2011.

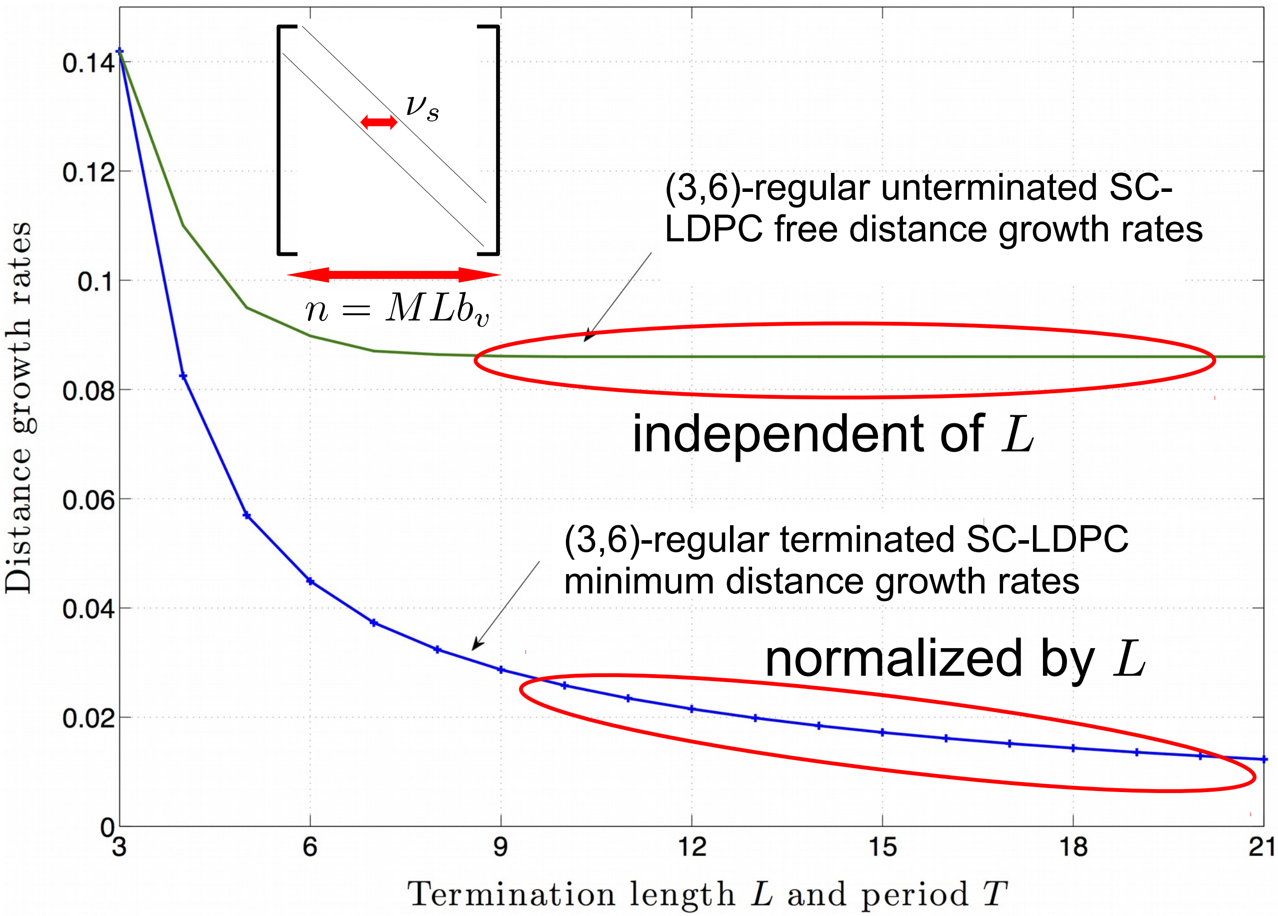
Distance Measures for SC-LDPC Codes

- As $L \to \infty$ the minimum distance growth rates of **terminated** SC-LDPC code ensembles tend to zero. However, the free distance growth rates of the **unterminated** ensembles remain constant.



Distance Measures for SC-LDPC Codes

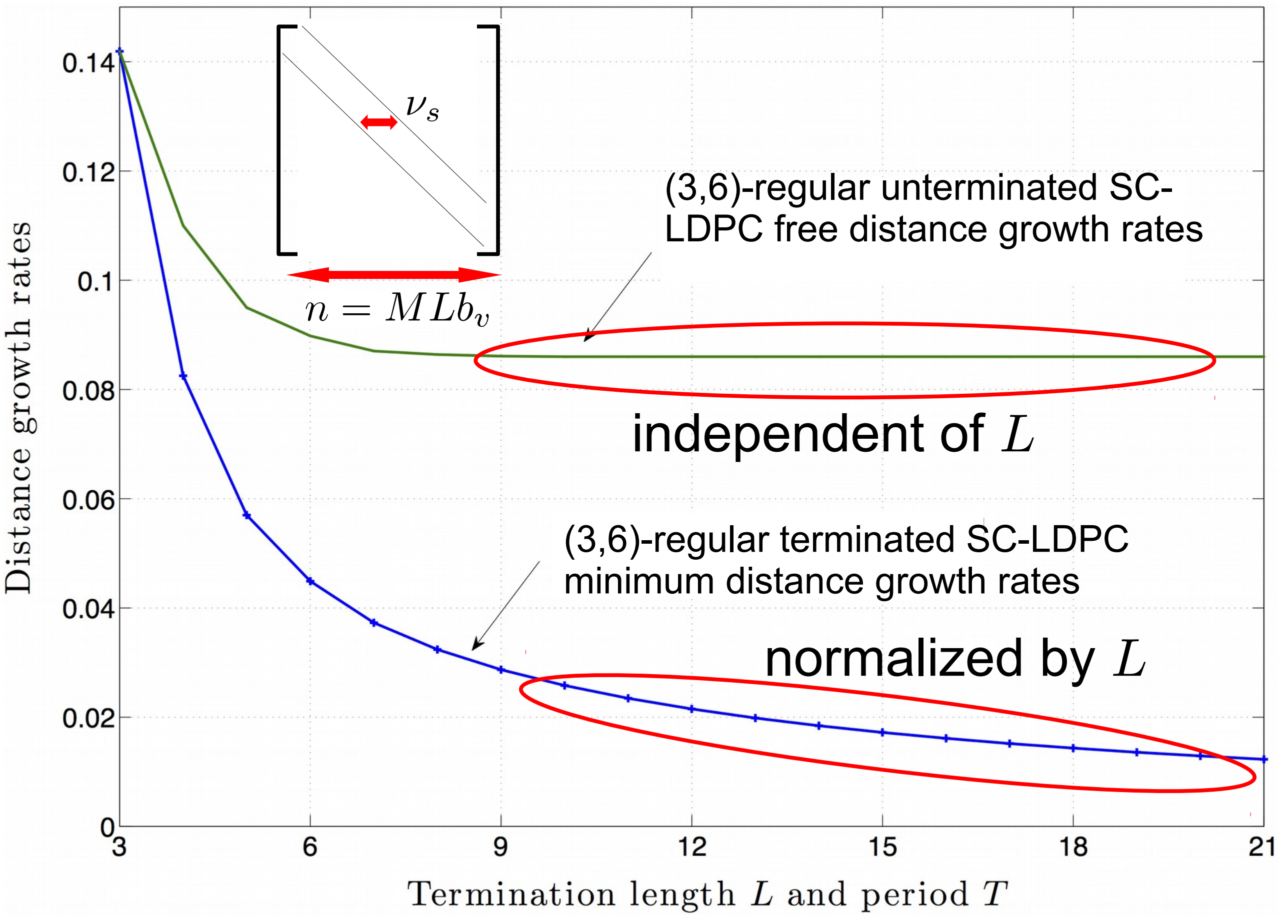
- As $L \to \infty$ the minimum distance growth rates of **terminated** SC-LDPC code ensembles tend to zero. However, the free distance growth rates of the **unterminated** ensembles remain constant.



- For large L, the strength of unterminated ensembles **scales with the constraint length** $\nu_s = M(m_s + 1)b_v$ and is **independent** of L.

Distance Measures for SC-LDPC Codes

- As $L \to \infty$ the minimum distance growth rates of **terminated** SC-LDPC code ensembles tend to zero. However, the free distance growth rates of the **unterminated** ensembles remain constant.



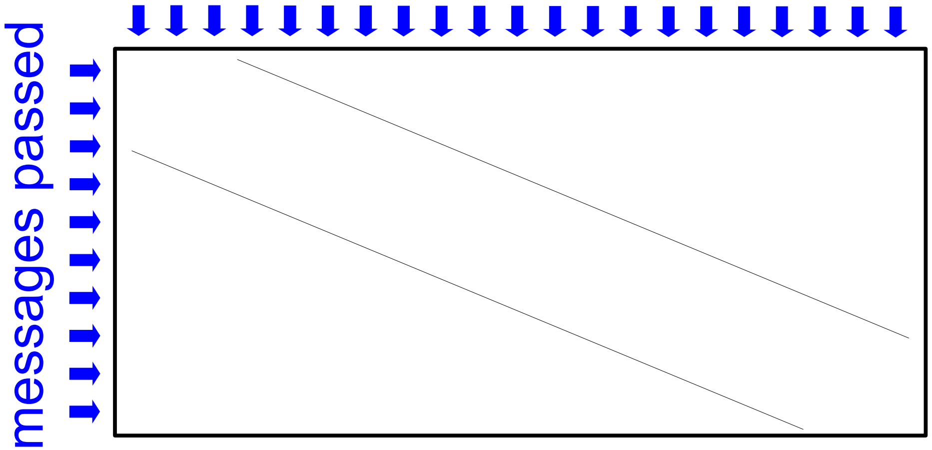
- For large L, the strength of unterminated ensembles **scales with the constraint length** $\nu_s = M(m_s+1)b_v$ and is **independent** of L.
- An appropriate distance measure for 'convolutional-like' terminated ensembles should be independent of L.

Outline

- **LDPC Block Codes**
  - ➡ Parity-check matrix and Tanner graph representations, minimum distance bounds, iterative decoding thresholds, protograph-based constructions
- **Spatially Coupled LDPC Codes**
  - ➡ Protograph representation, edge-spreading construction, termination
  - ➡ Iterative decoding thresholds, threshold saturation, minimum distance
- **Practical Considerations**
  - ➡ Window decoding; performance, latency, and complexity comparisons to LDPC block codes; rate-compatibility; implementation aspects

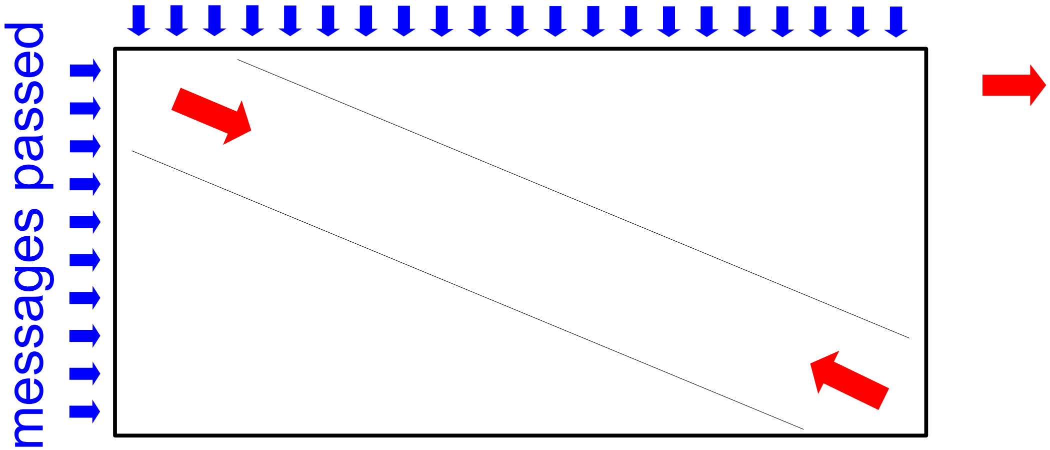
Block Decoding of SC-LDPC Codes

- SC-LDPC codes can be decoded with standard iterative decoding schedules.



Block Decoding of SC-LDPC Codes

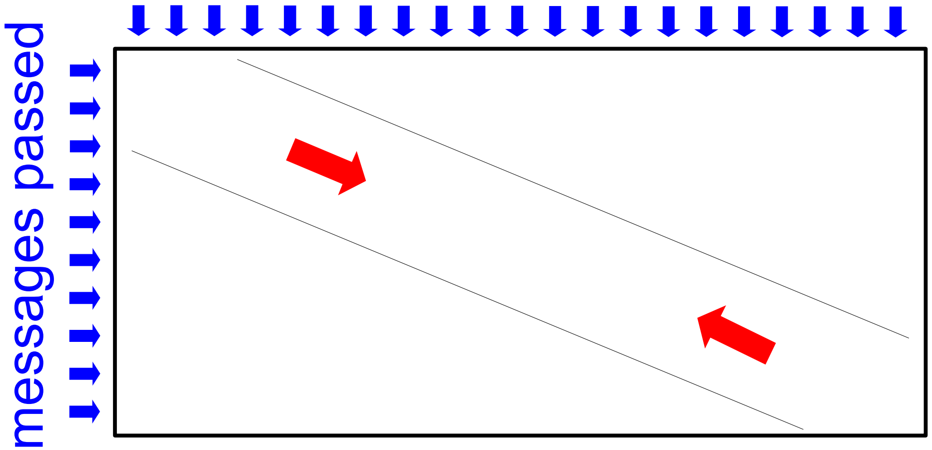
- SC-LDPC codes can be decoded with standard iterative decoding schedules.



Reliable messages from the ends propagate through the graph toward the center as iterations proceed.

Block Decoding of SC-LDPC Codes

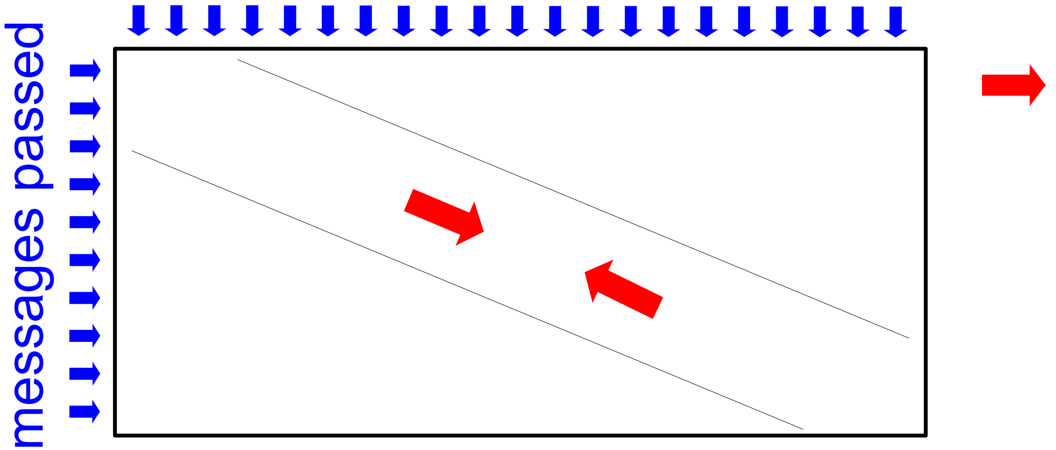
- SC-LDPC codes can be decoded with standard iterative decoding schedules.



Reliable messages from the ends propagate through the graph toward the center as iterations proceed.

Block Decoding of SC-LDPC Codes

- SC-LDPC codes can be decoded with standard iterative decoding schedules.

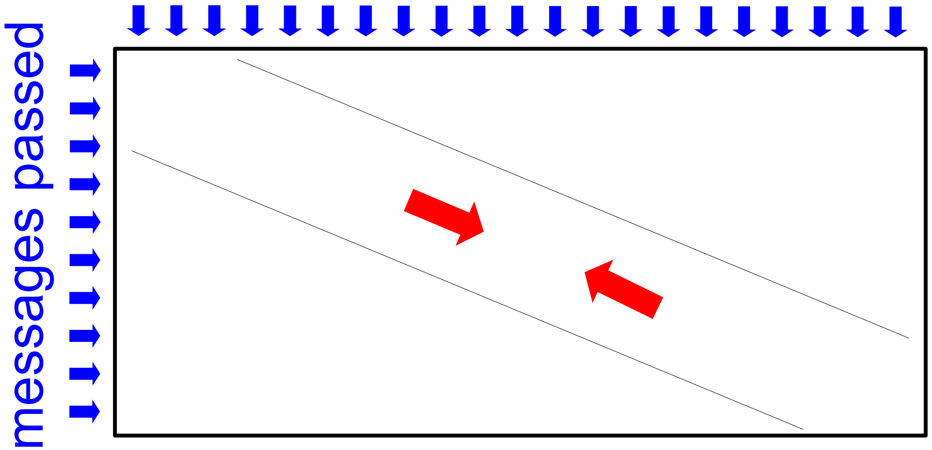


Reliable messages from the ends propagate through the graph toward the center as iterations proceed.

Block Decoding of SC-LDPC Codes

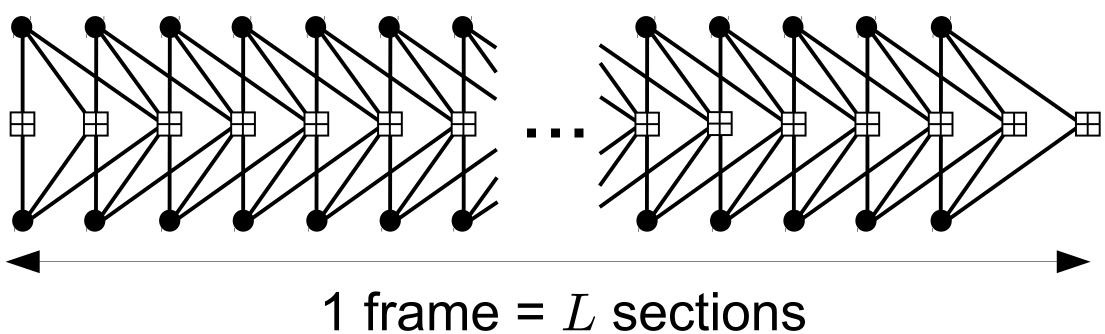
UNIVERSITY OF NOTRE DAME

- SC-LDPC codes can be decoded with standard iterative decoding schedules.



➡ Reliable messages from the ends propagate through the graph toward the center as iterations proceed.

- The **frame error rate (FER)** of a terminated graph can be analyzed

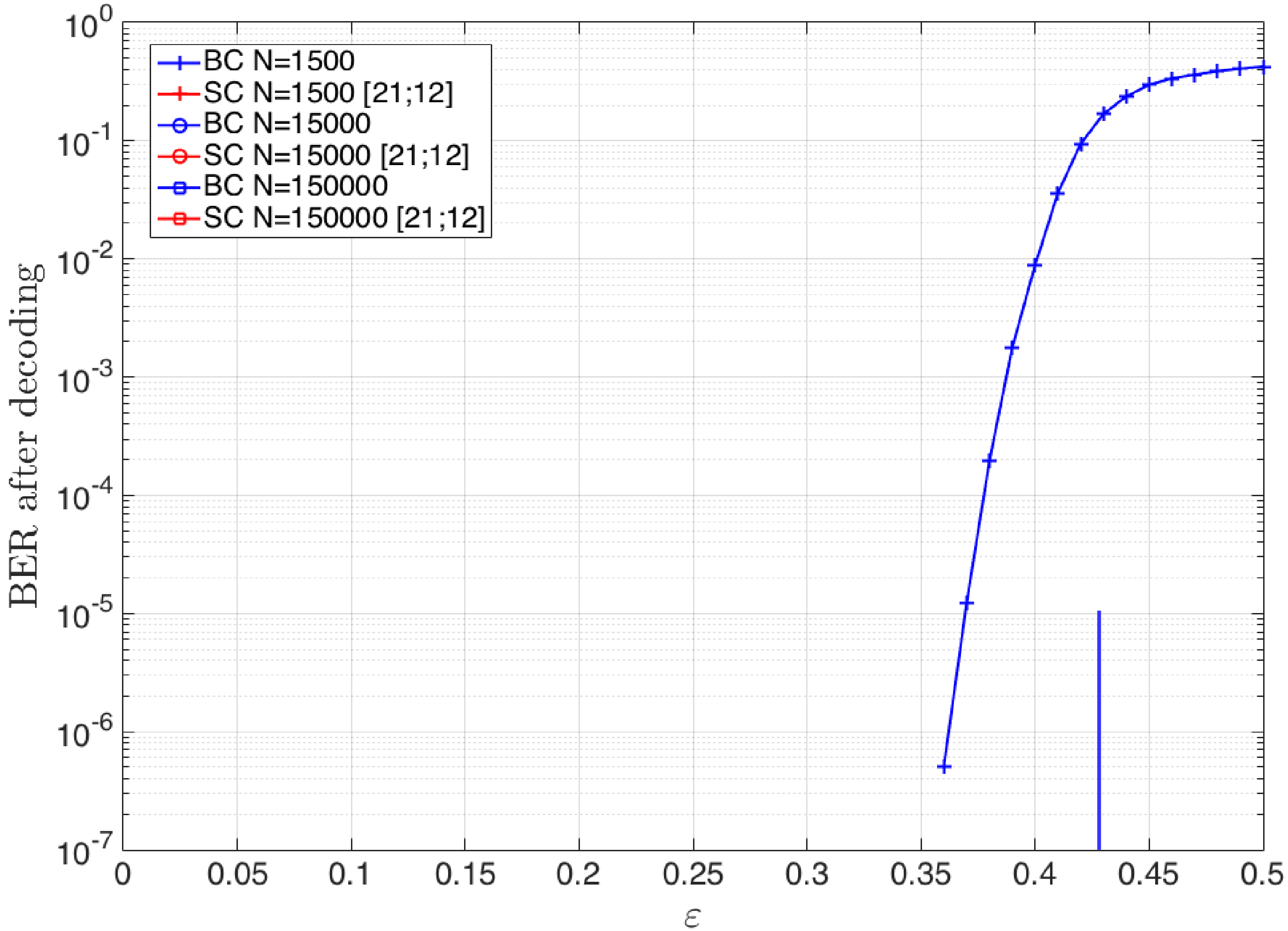


➡ The **FER depends on L** (FER $\underset{L\to\infty}{\longrightarrow} 1$)

Block Decoding Performance

UNIVERSITY OF NOTRE DAME

- Consider LDPC-BCs and SC-LDPC codes with increasing frame length N

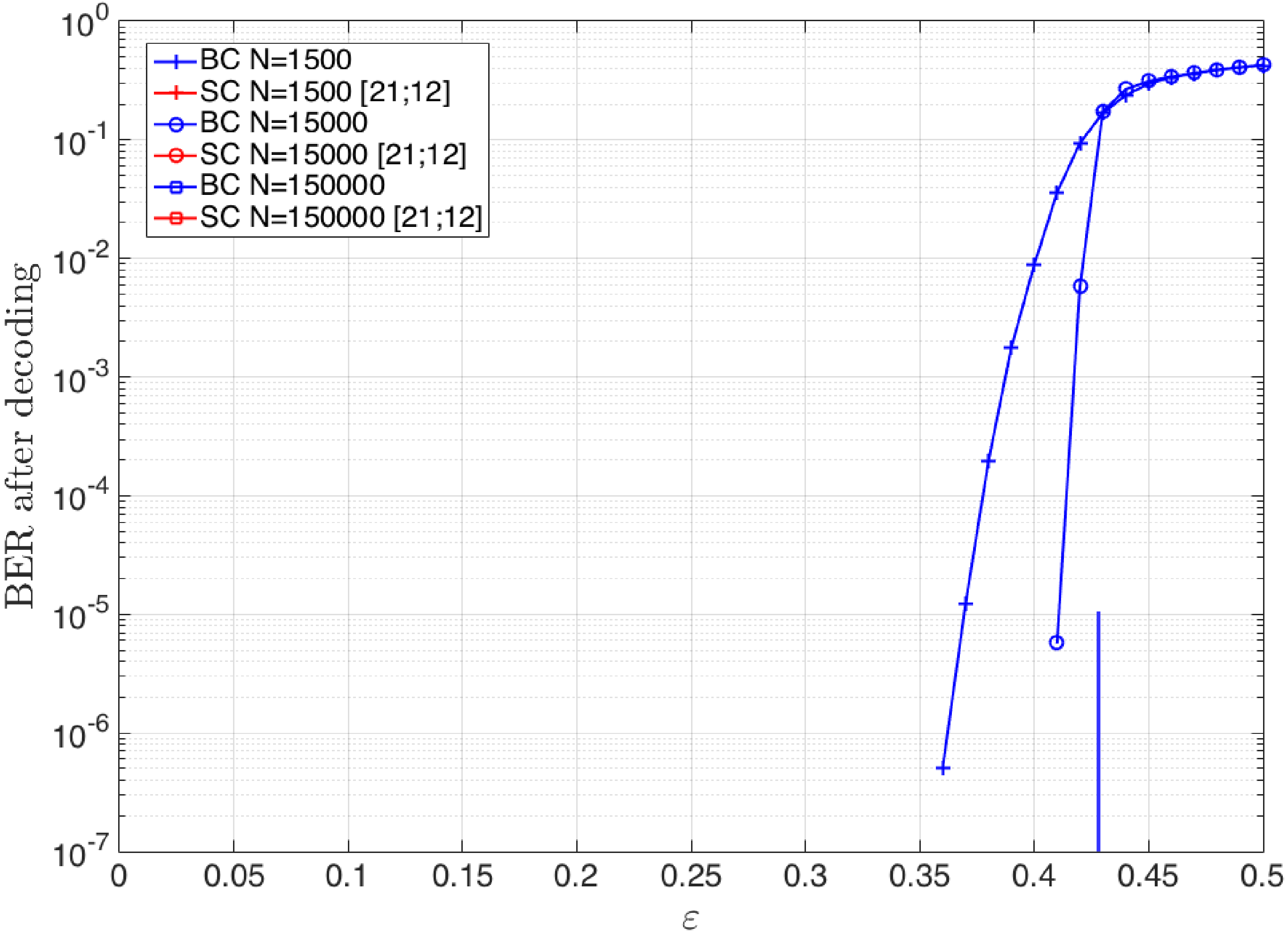


- As N increases the LDPC-BC performance approaches

$$\varepsilon^* = 0.429$$

Block Decoding Performance

- Consider LDPC-BCs and SC-LDPC codes with increasing frame length N

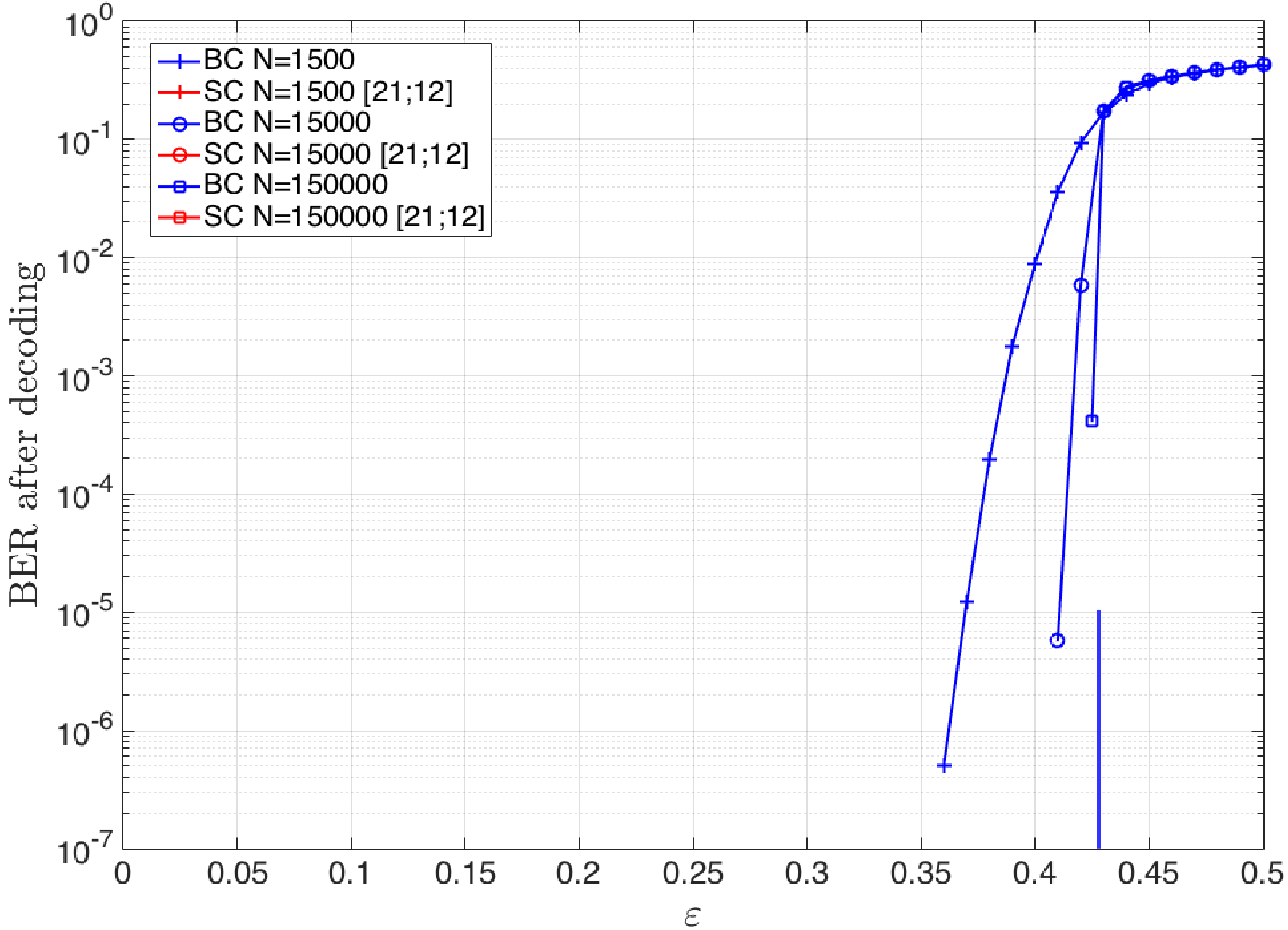


- As N increases the LDPC-BC performance approaches

$$\varepsilon^* = 0.429$$

Block Decoding Performance

- Consider LDPC-BCs and SC-LDPC codes with increasing frame length N

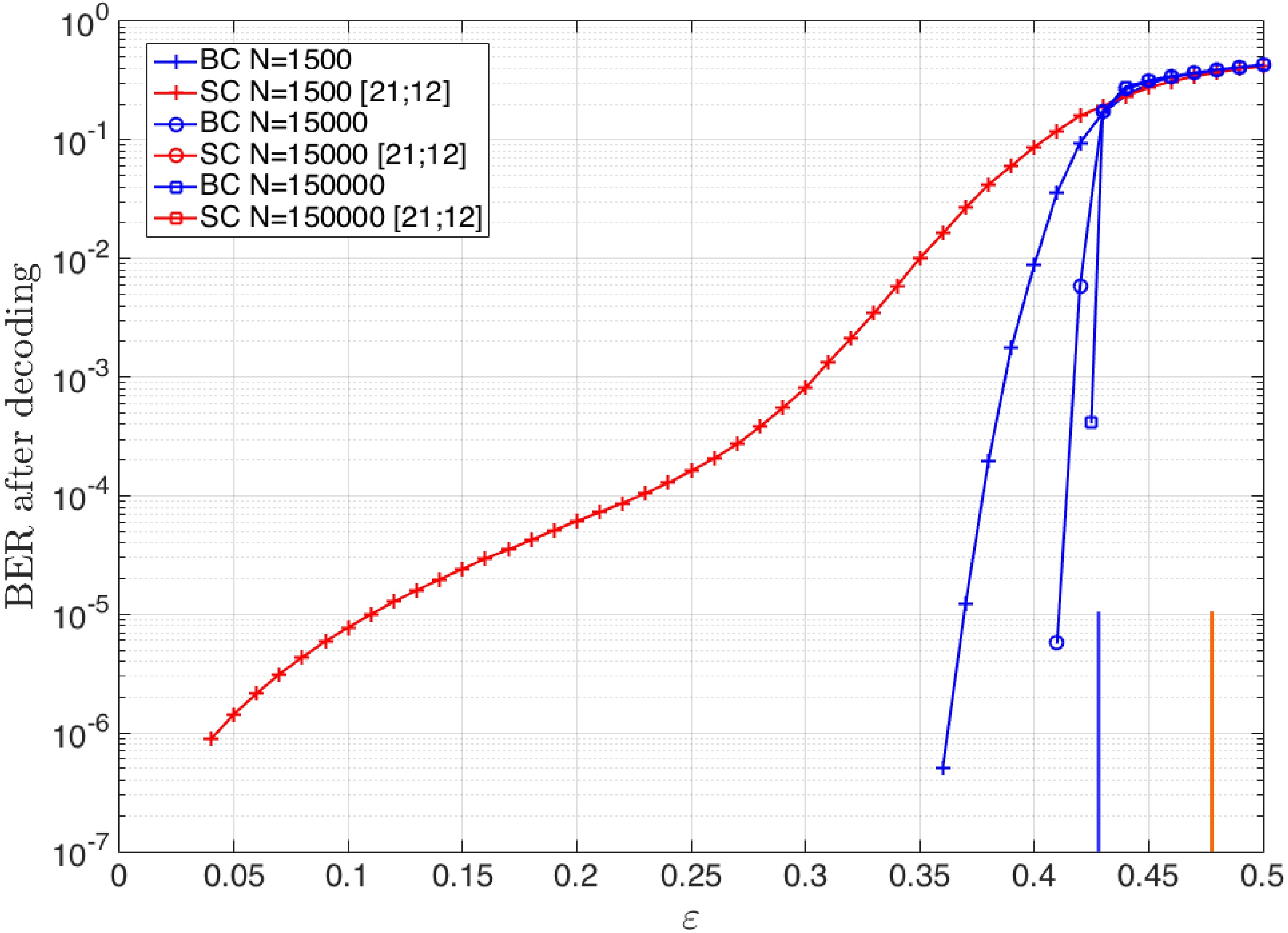


- As N increases the LDPC-BC performance approaches

$$\varepsilon^* = 0.429$$

Block Decoding Performance

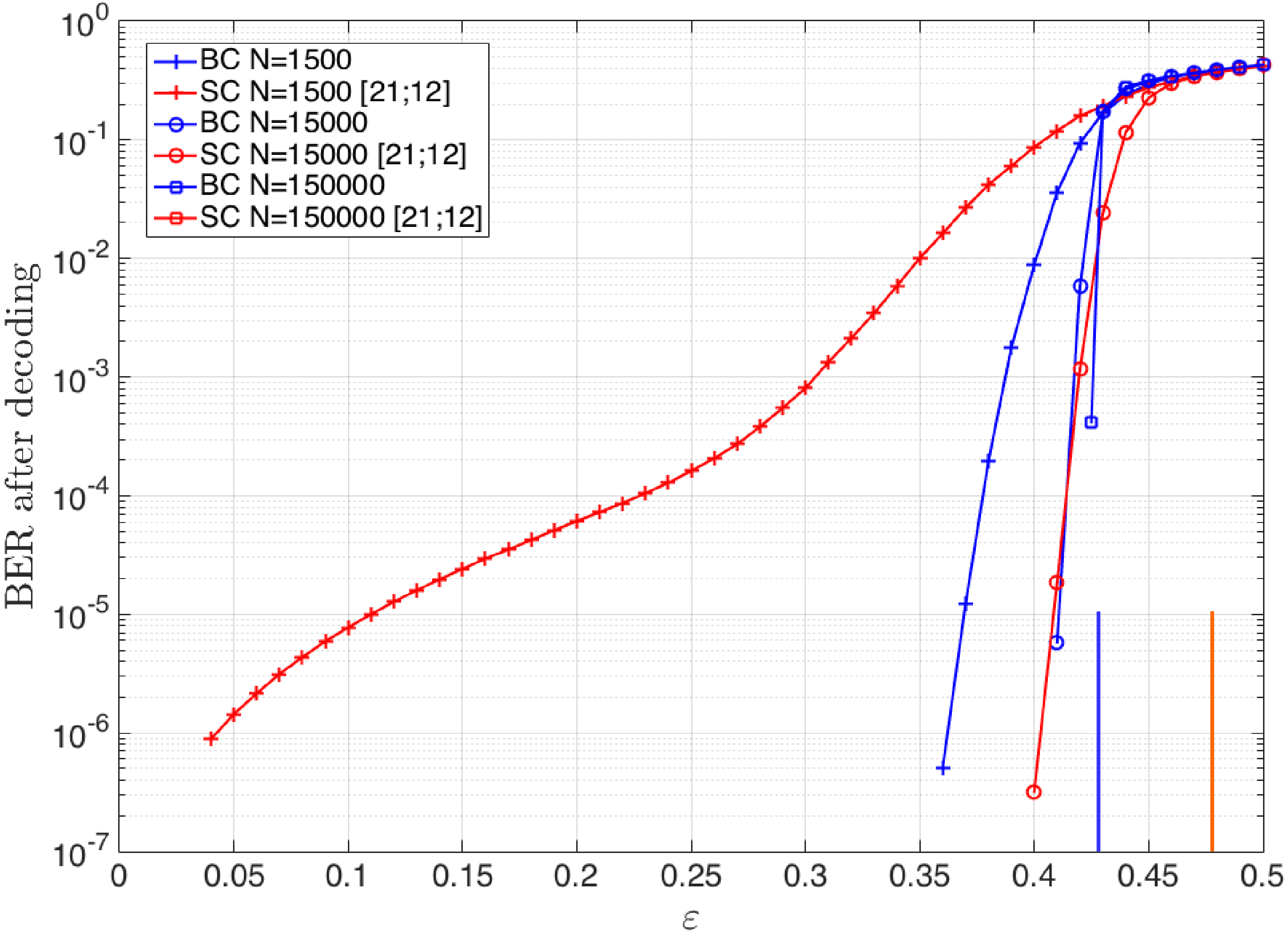
- Consider LDPC-BCs and SC-LDPC codes with increasing frame length N



- As N increases the LDPC-BC performance approaches $\varepsilon^* = 0.429$
- As N increases the SC-LDPC code performance approaches $\varepsilon^* = 0.488$ outperforming the LDPC-BC for a sufficiently large N

Block Decoding Performance

- Consider LDPC-BCs and SC-LDPC codes with increasing frame length N



- As N increases the LDPC-BC performance approaches

$$\varepsilon^* = 0.429$$

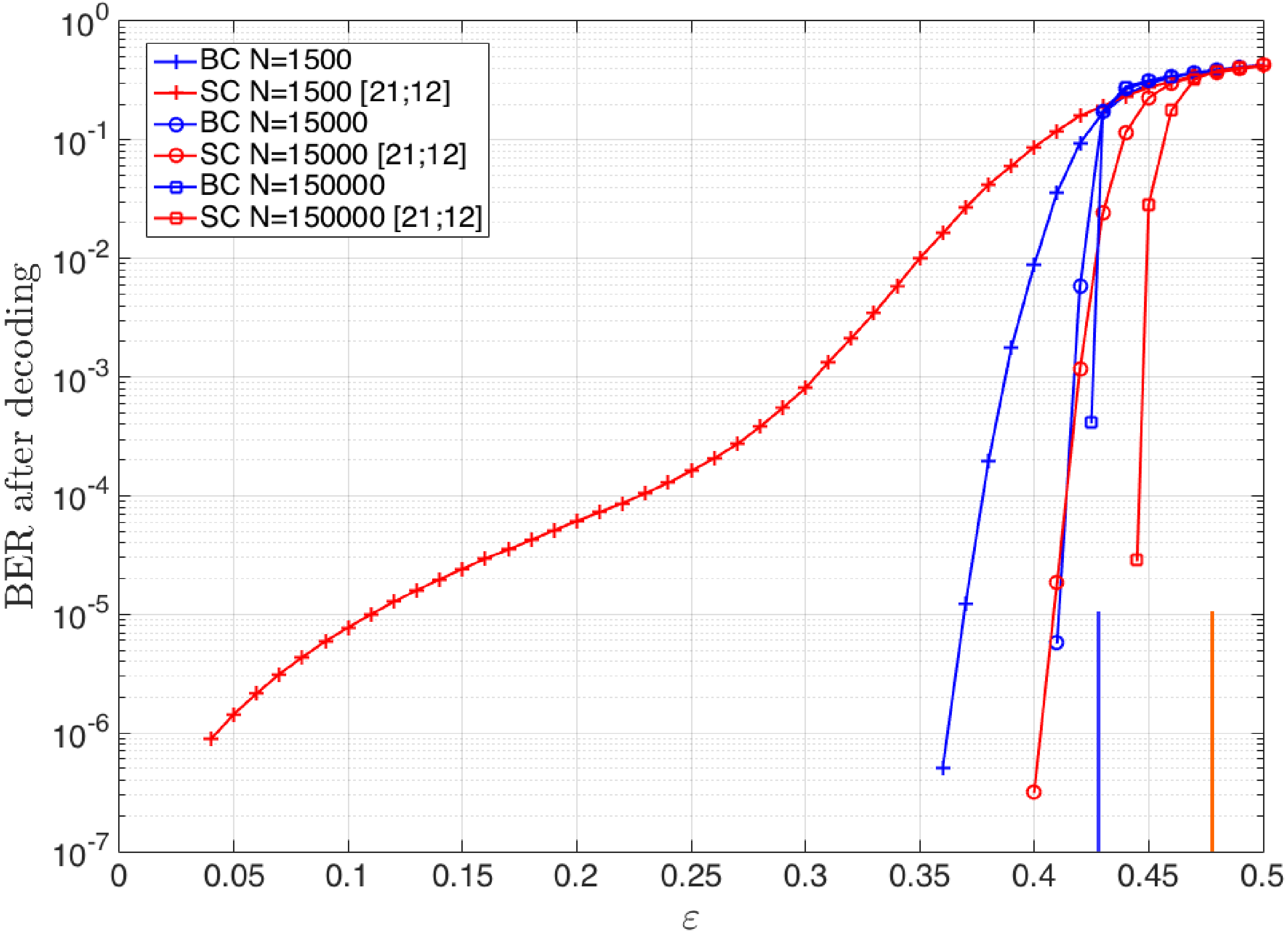
- As N increases the SC-LDPC code performance approaches

$$\varepsilon^* = 0.488$$

outperforming the LDPC-BC for a sufficiently large N

Block Decoding Performance

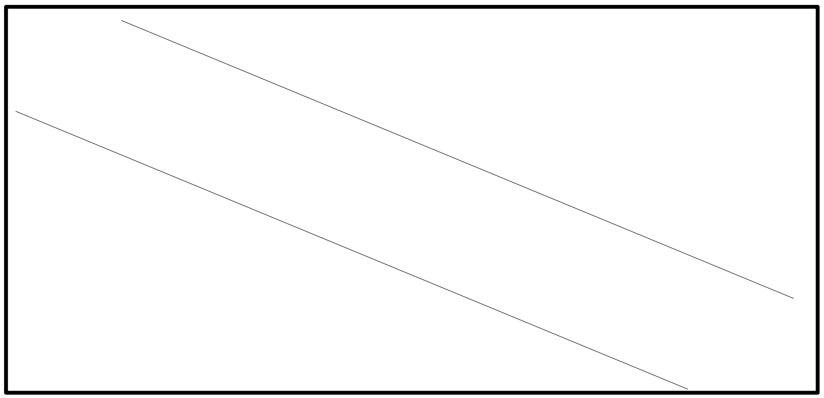
- Consider LDPC-BCs and SC-LDPC codes with increasing frame length N



- As N increases the LDPC-BC performance approaches $\varepsilon^* = 0.429$
- As N increases the SC-LDPC code performance approaches $\varepsilon^* = 0.488$ outperforming the LDPC-BC for a sufficiently large N

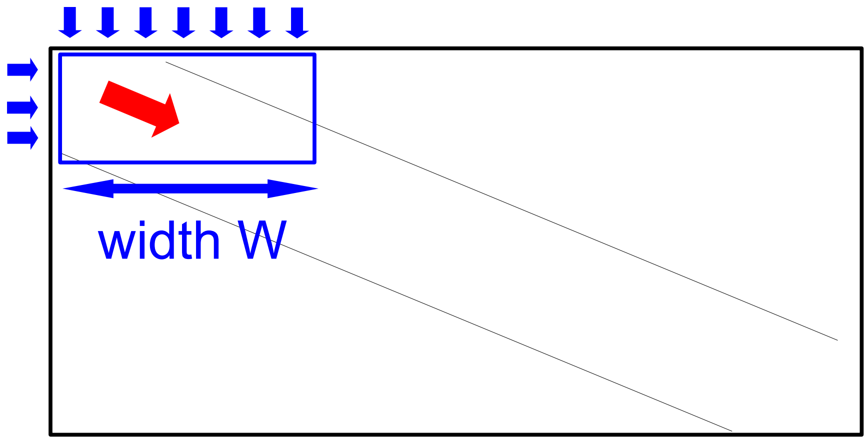
Window Decoding of SC-LDPC Codes

- The **highly localized (convolutional) structure** is well-suited for efficient decoding schedules that reduce memory and latency requirements.



Window Decoding of SC-LDPC Codes

- The **highly localized (convolutional) structure** is well-suited for efficient decoding schedules that reduce memory and latency requirements.

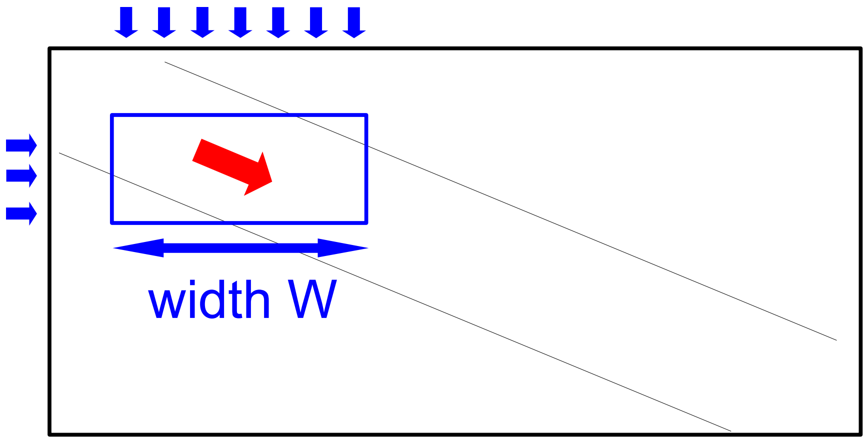


➡ **Sliding window decoding** (WD) updates nodes only within a localized window and then the window shifts across the graph [Lentmaier et al '10, Iyengar et al '12].

Window Decoding of SC-LDPC Codes

UNIVERSITY OF NOTRE DAME

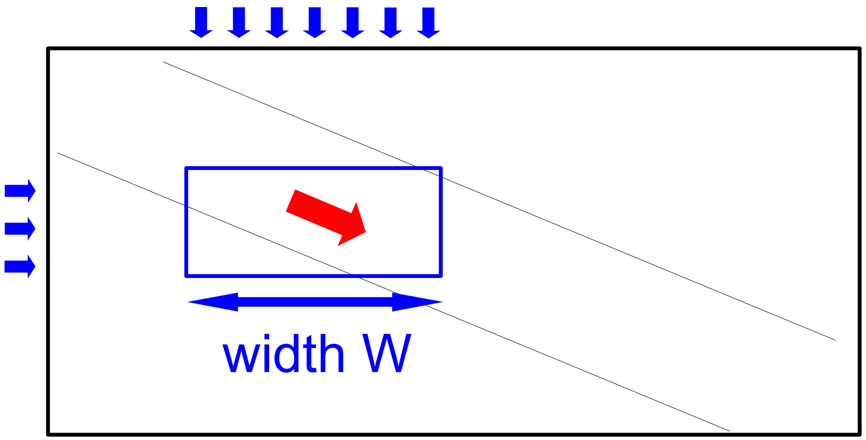
- The **highly localized (convolutional) structure** is well-suited for efficient decoding schedules that reduce memory and latency requirements.



Sliding window decoding (WD) updates nodes only within a localized window and then the window shifts across the graph [Lentmaier et al '10, Iyengar et al '12].

Window Decoding of SC-LDPC Codes

- The **highly localized (convolutional) structure** is well-suited for efficient decoding schedules that reduce memory and latency requirements.

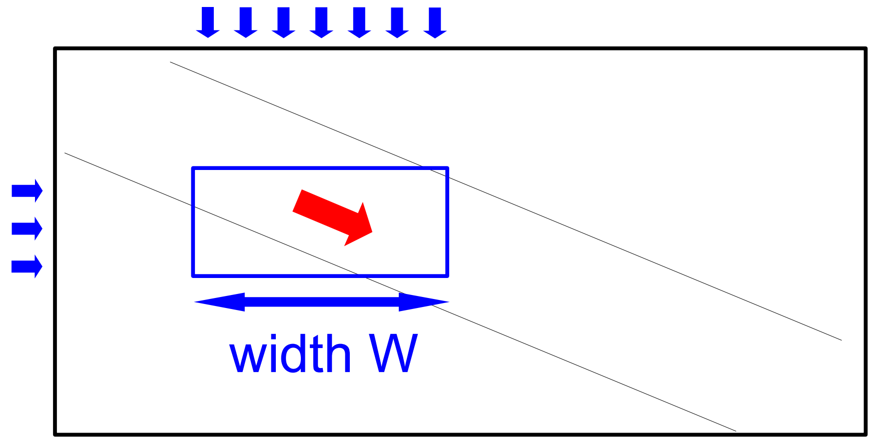


➡ **Sliding window decoding** (WD) updates nodes only within a localized window and then the window shifts across the graph [Lentmaier et al '10, Iyengar et al '12].

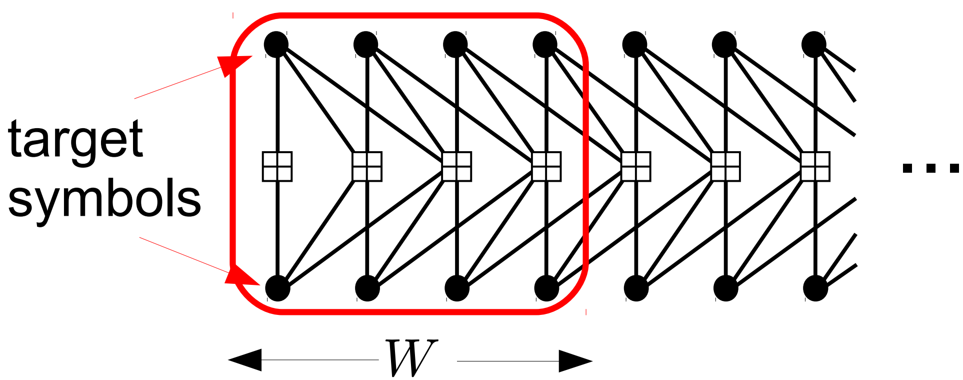
Window Decoding of SC-LDPC Codes

UNIVERSITY OF NOTRE DAME

- The **highly localized (convolutional) structure** is well-suited for efficient decoding schedules that reduce memory and latency requirements.



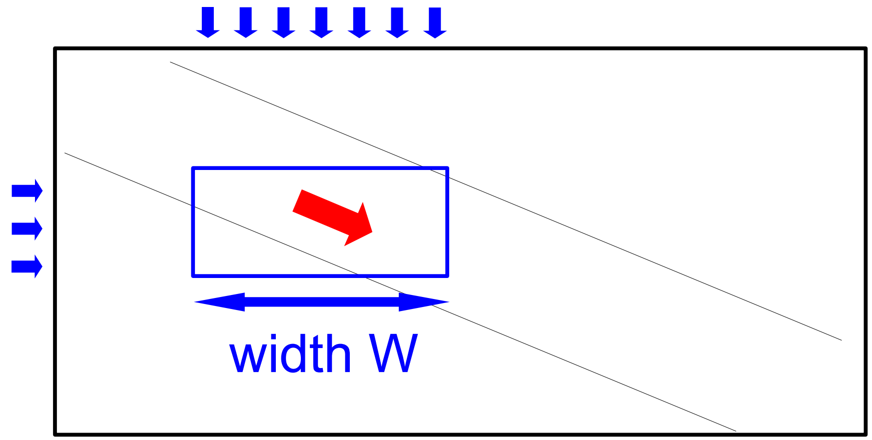
➡ **Sliding window decoding** (WD) updates nodes only within a localized window and then the window shifts across the graph [Lentmaier et al '10, Iyengar et al '12].



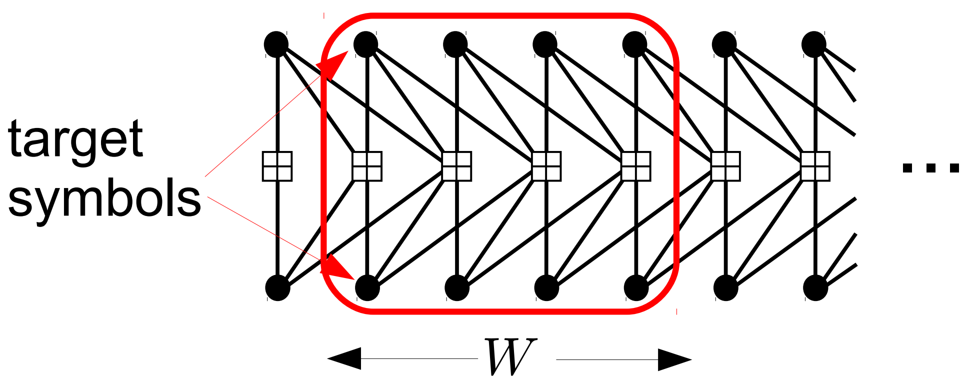
➡ One block of cM **target symbols** is decoded in each window position

Window Decoding of SC-LDPC Codes

- The **highly localized (convolutional) structure** is well-suited for efficient decoding schedules that reduce memory and latency requirements.



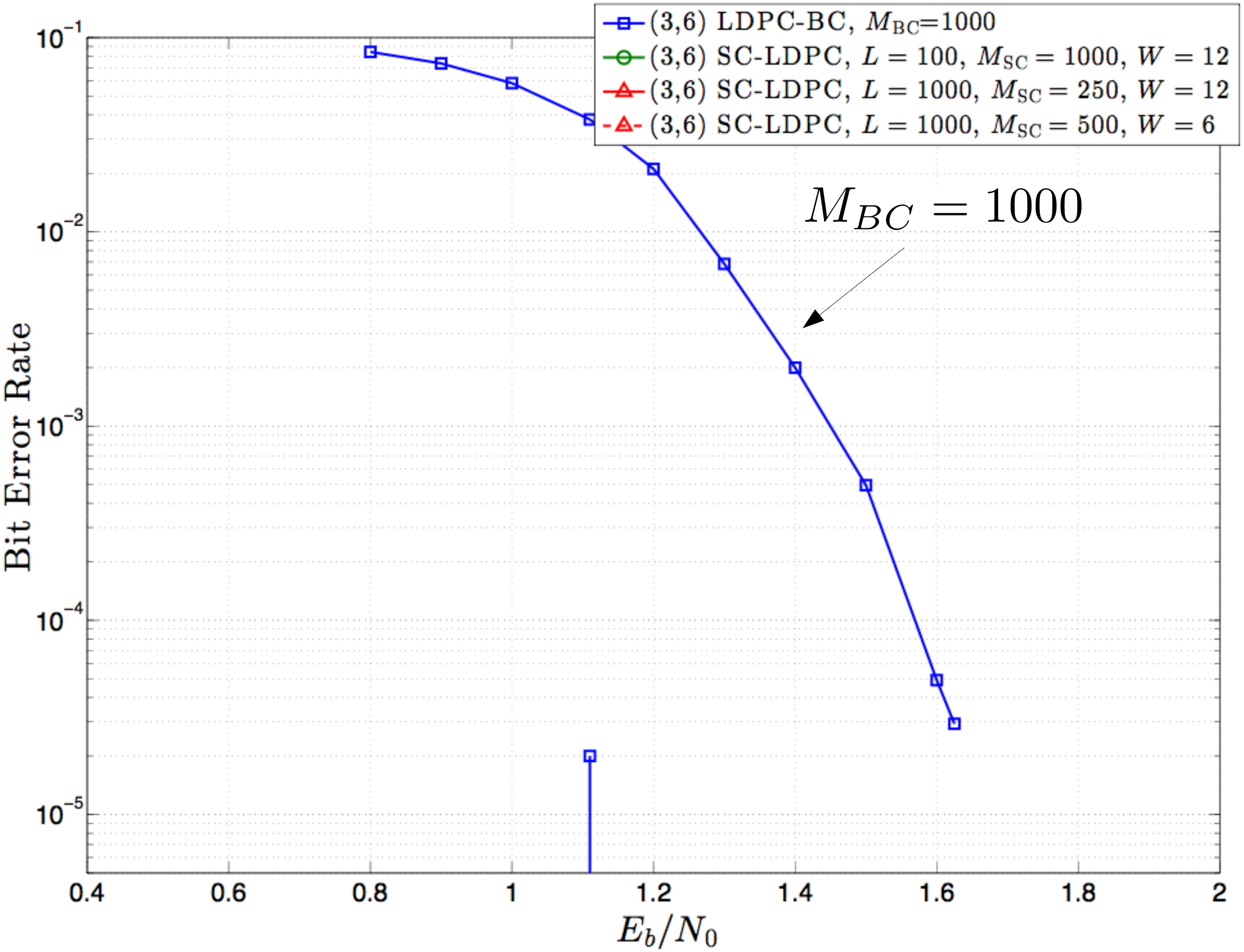
➡ **Sliding window decoding** (WD) updates nodes only within a localized window and then the window shifts across the graph [Lentmaier et al '10, Iyengar et al '12].



➡ One block of cM **target symbols** is decoded in each window position

➡ The window then shifts to the right

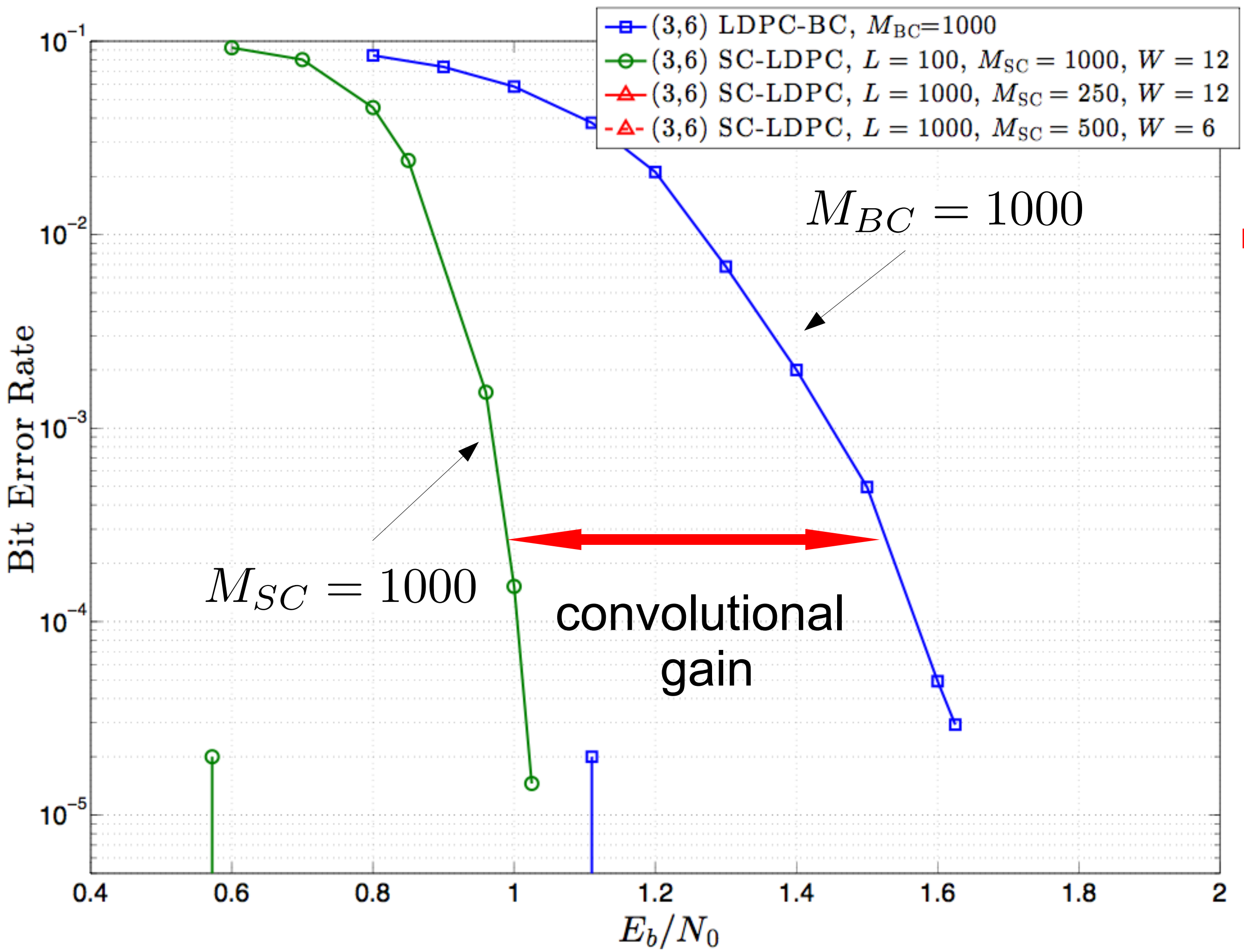
Window Decoding Performance



Latencies:
LDPC: $6M_{BC}$
SC-LDPC: $2M_{SC}W$

[LPF11] M. Lentmaier, M. M. Prenda, and G. Fettweis, "Efficient Message Passing Scheduling for Terminated LDPC Convolutional Codes", *Proc. IEEE ISIT*, St. Petersburg, Russia, July 2011.

Window Decoding Performance

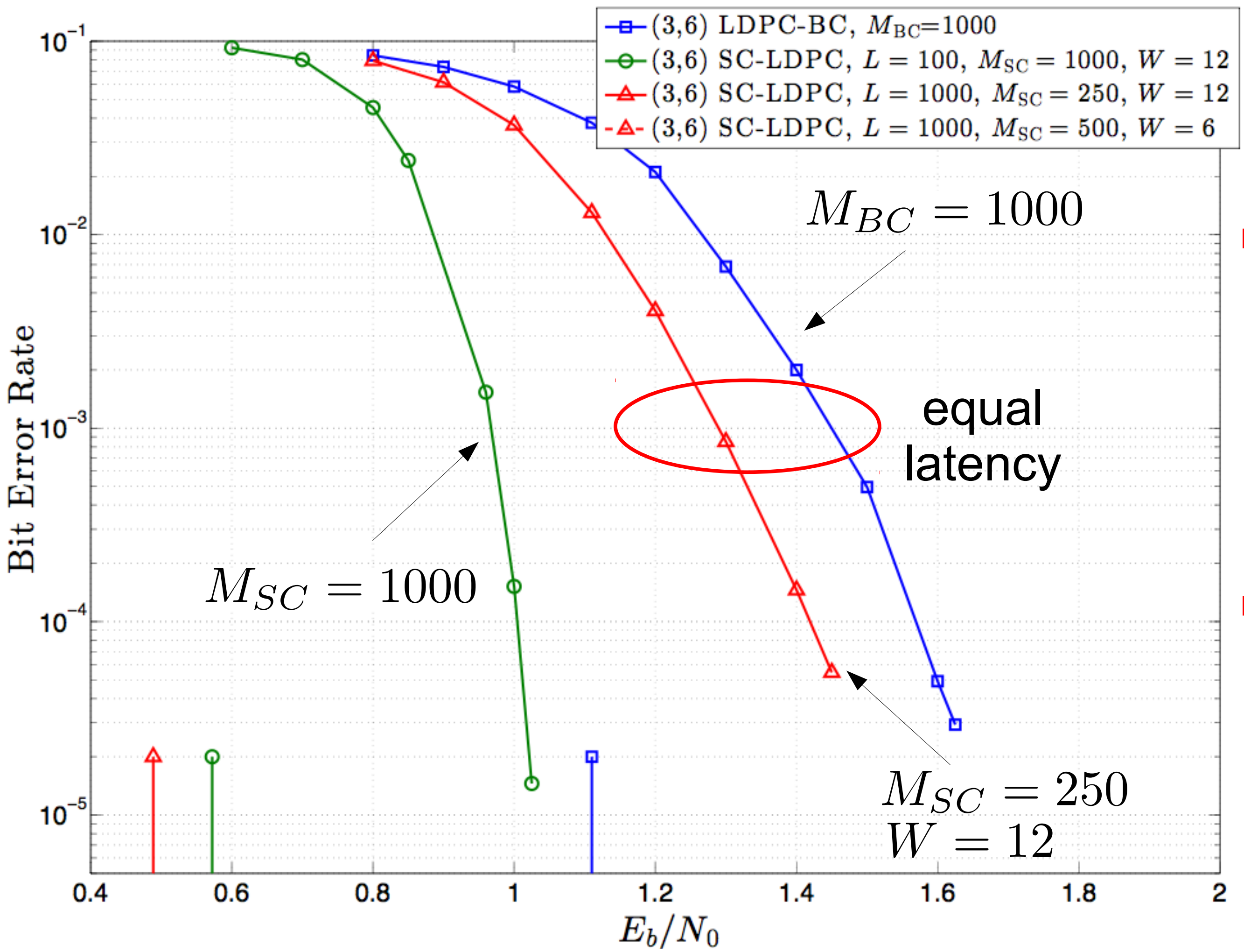


Latencies:
LDPC: $6M_{BC}$
SC-LDPC: $2M_{SC}W$

For **equal lifting factors**, SC-LDPC codes display a large **convolutional gain** at the cost of increased latency.

[LPF11] M. Lentmaier, M. M. Prenda, and G. Fettweis, "Efficient Message Passing Scheduling for Terminated LDPC Convolutional Codes", *Proc. IEEE ISIT*, St. Petersburg, Russia, July 2011.

Window Decoding Performance



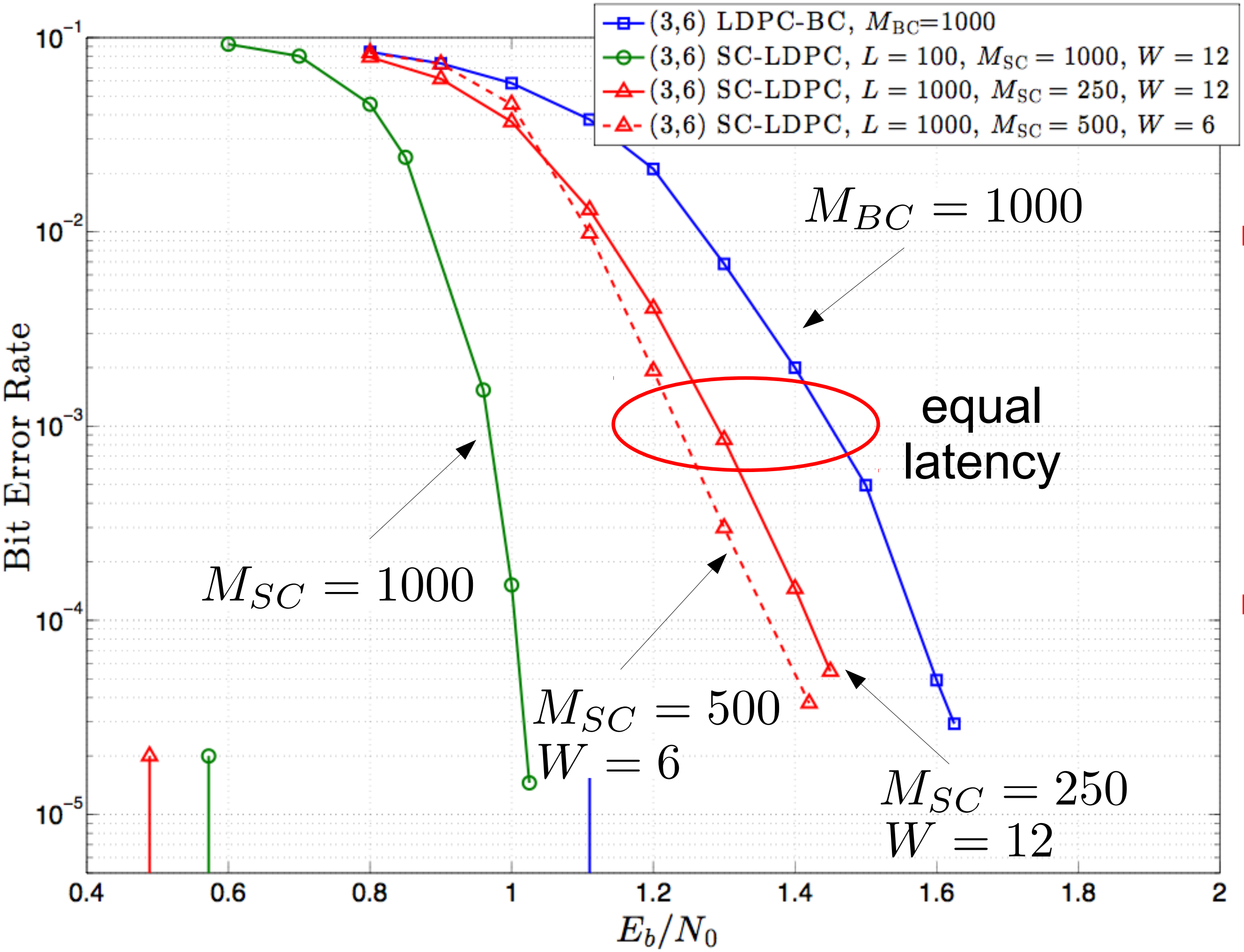
Latencies:
LDPC: $6M_{BC}$
SC-LDPC: $2M_{SC}W$

- For **equal lifting factors**, SC-LDPC codes display a large **convolutional gain** at the cost of increased latency.
- For **equal latency**, SC-LDPC codes still display a significant **performance gain**.

[LPF11] M. Lentmaier, M. M. Prenda, and G. Fettweis, “Efficient Message Passing Scheduling for Terminated LDPC Convolutional Codes”, *Proc. IEEE ISIT*, St. Petersburg, Russia, July 2011.

Window Decoding Performance

UNIVERSITY OF NOTRE DAME

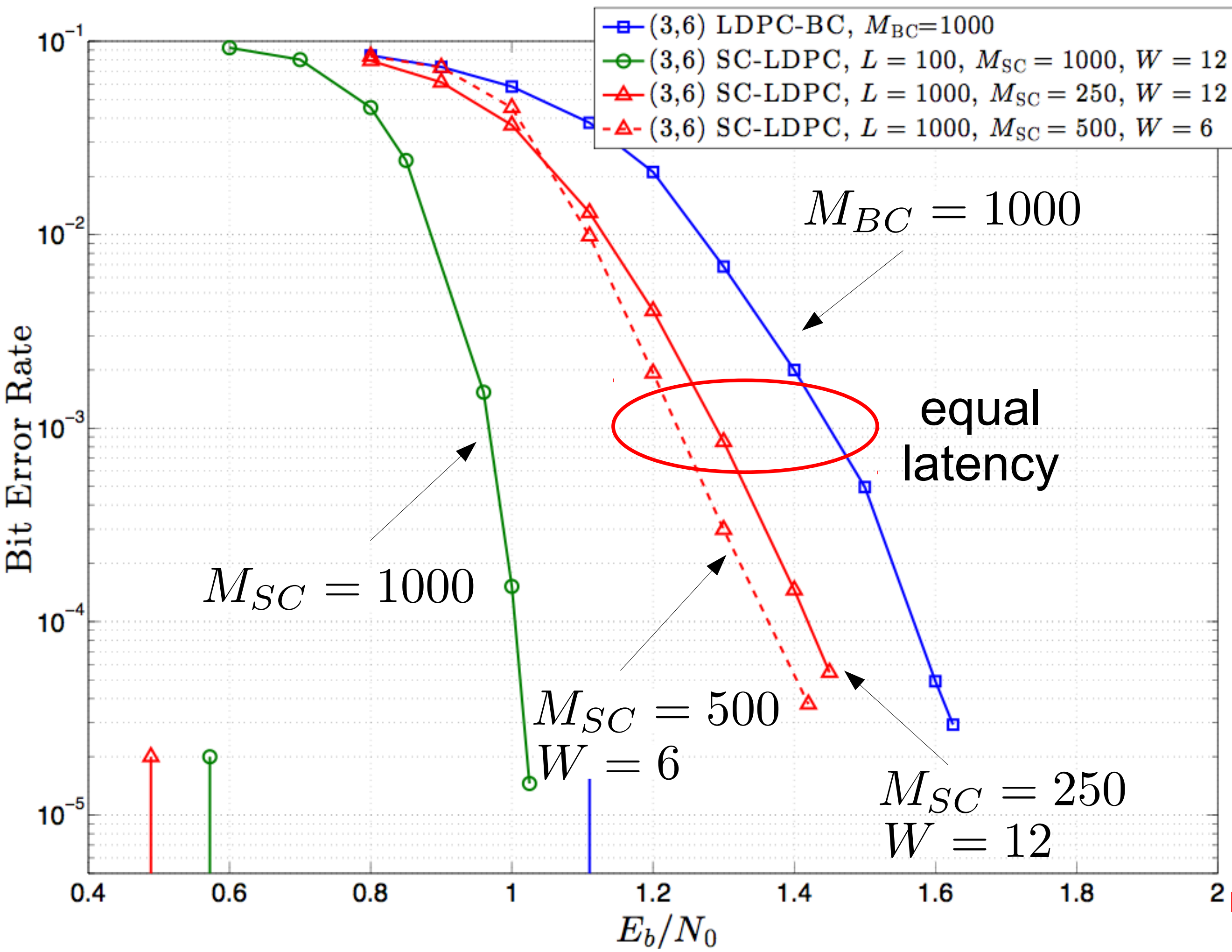


Latencies:
LDPC: $6M_{BC}$
SC-LDPC: $2M_{SC}W$

- For **equal lifting factors**, SC-LDPC codes display a large **convolutional gain** at the cost of increased latency.
- For **equal latency**, SC-LDPC codes still display a significant **performance gain**.

[LPF11] M. Lentmaier, M. M. Prenda, and G. Fettweis, "Efficient Message Passing Scheduling for Terminated LDPC Convolutional Codes", *Proc. IEEE ISIT*, St. Petersburg, Russia, July 2011.

Window Decoding Performance



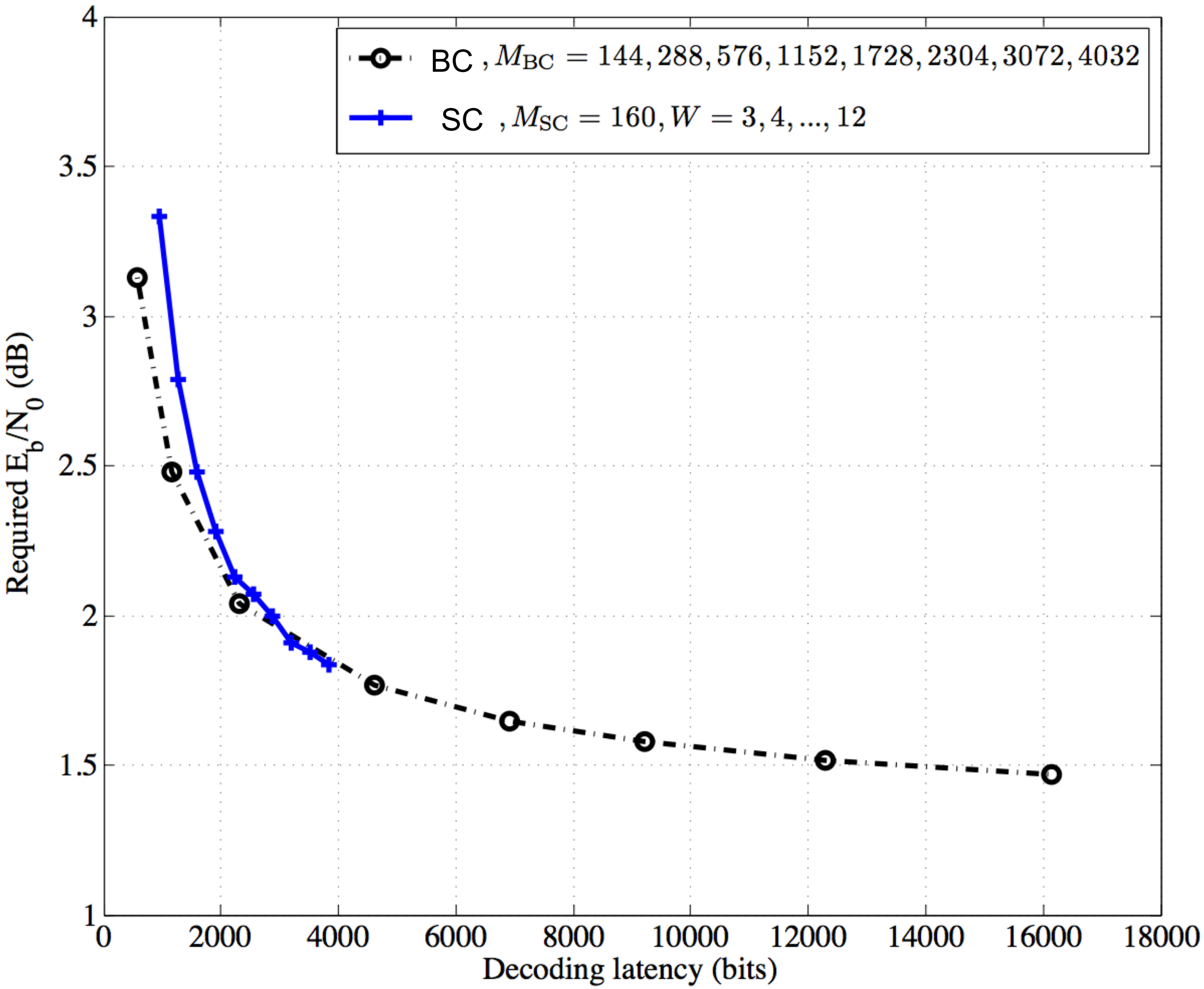
Latencies:
LDPC: $6M_{BC}$
SC-LDPC: $2M_{SC}W$

- For **equal lifting factors**, SC-LDPC codes display a large **convolutional gain** at the cost of increased latency.
- For **equal latency**, SC-LDPC codes still display a significant **performance gain**.
- Trade-off in M vs W

[LPF11] M. Lentmaier, M. M. Prenda, and G. Fettweis, "Efficient Message Passing Scheduling for Terminated LDPC Convolutional Codes", *Proc. IEEE ISIT*, St. Petersburg, Russia, July 2011.

Equal Latency Comparison for (3,6)-Regular LDPC Codes

- Required E_b/N_0 to achieve a BER of 10^{-5} as a function of latency:



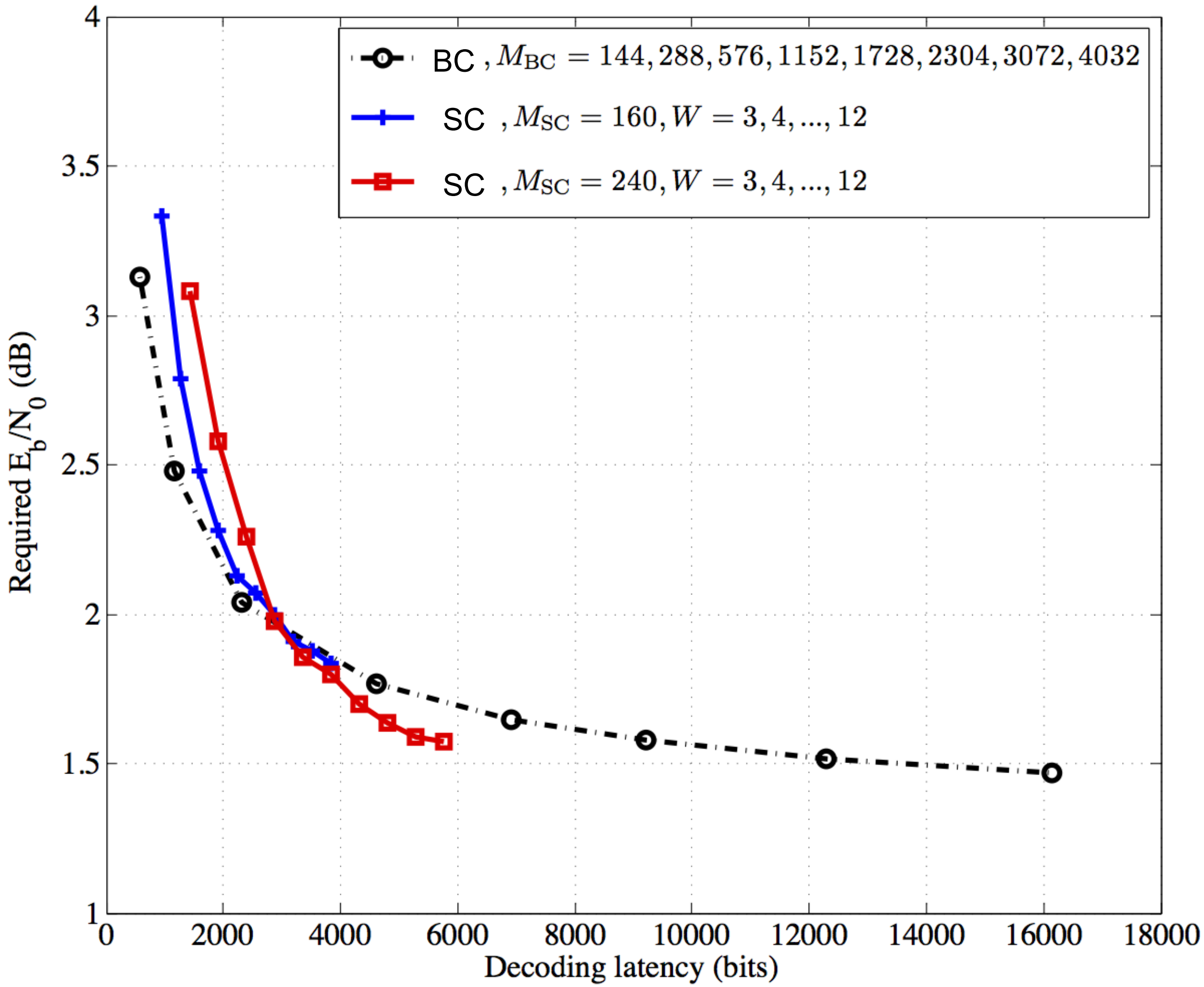
Latencies:
LDPC: $4M_{BC}$
SC-LDPC: $2M_{SC}W$

- ➡ decreases as W (and thus the latency) increases.
- ➡ does not decrease significantly beyond a certain W $(W \approx 10)$.

Equal Latency Comparison for (3,6)-Regular LDPC Codes

UNIVERSITY OF NOTRE DAME

- Required E_b/N_0 to achieve a BER of 10^{-5} as a function of latency:

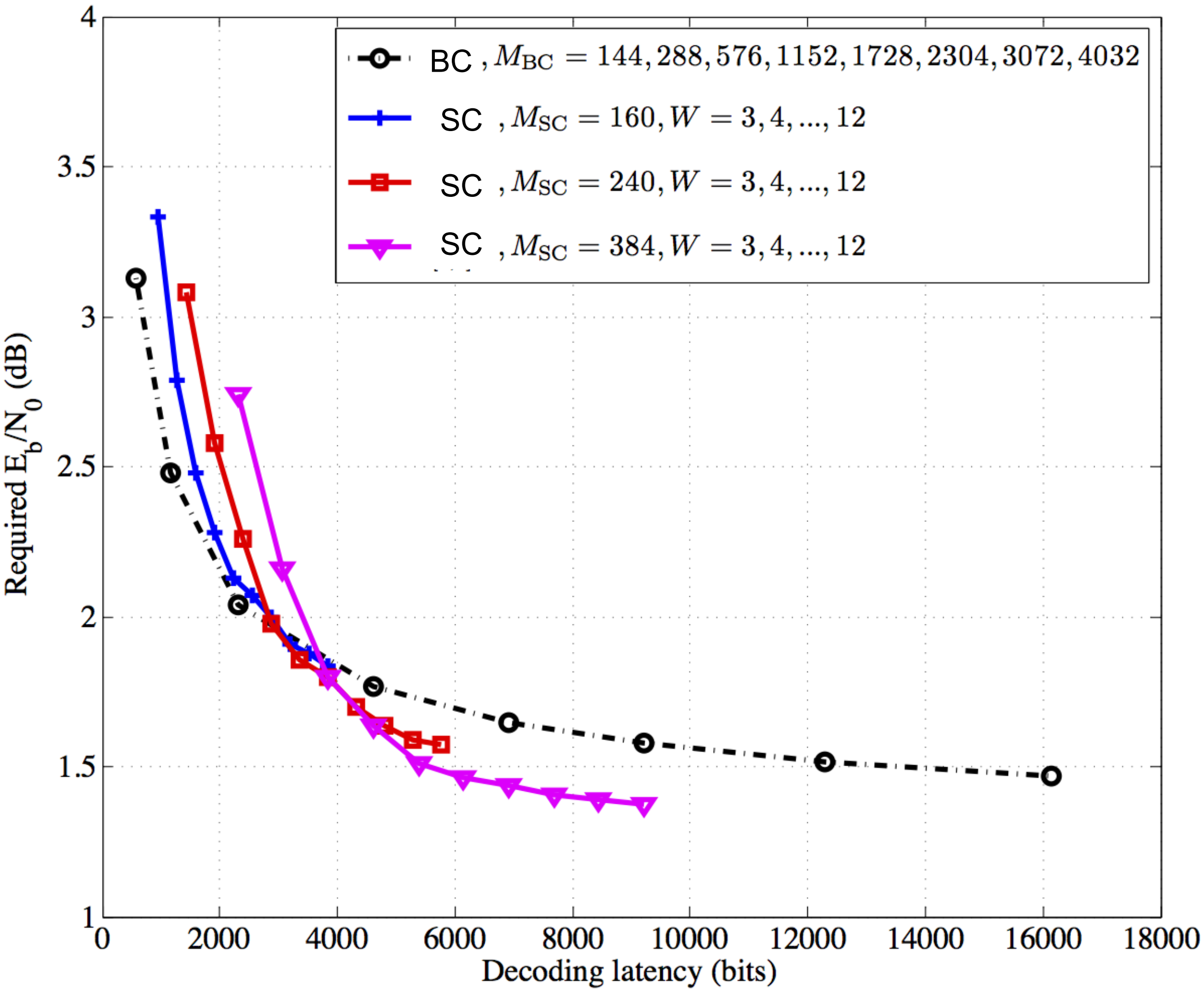


Latencies:
LDPC: $4M_{BC}$
SC-LDPC: $2M_{SC}W$

- decreases as W (and thus the latency) increases.
- does not decrease significantly beyond a certain W $(W \approx 10)$.

Equal Latency Comparison for (3,6)-Regular LDPC Codes

- Required E_b/N_0 to achieve a BER of 10^{-5} as a function of latency:



Latencies:
LDPC: $4M_{BC}$
SC-LDPC: $2M_{SC}W$

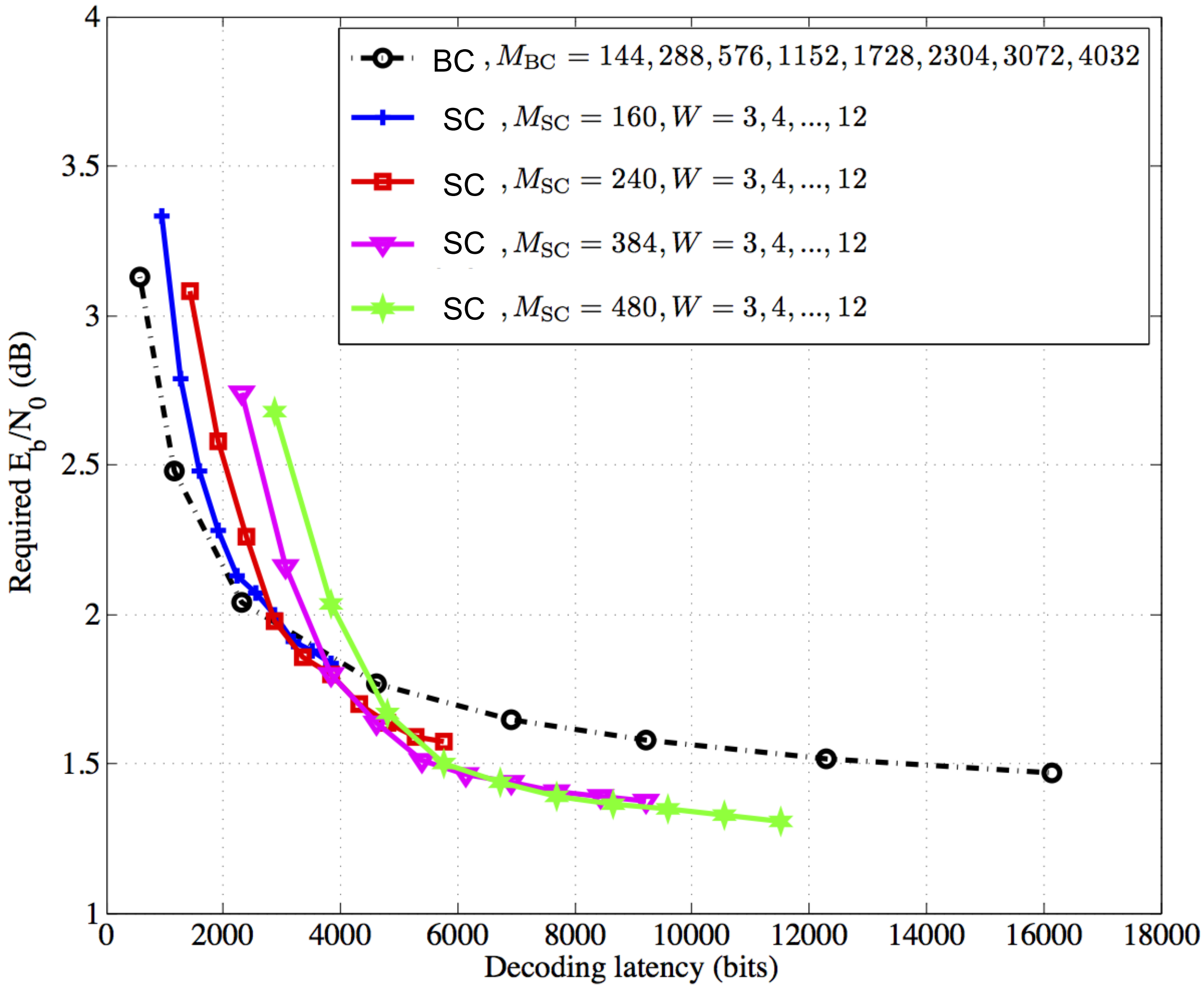
➡ decreases as W (and thus the latency) increases.

➡ does not decrease significantly beyond a certain W $(W \approx 10)$.

Equal Latency Comparison for (3,6)-Regular LDPC Codes

UNIVERSITY OF NOTRE DAME

- Required E_b/N_0 to achieve a BER of 10^{-5} as a function of latency:

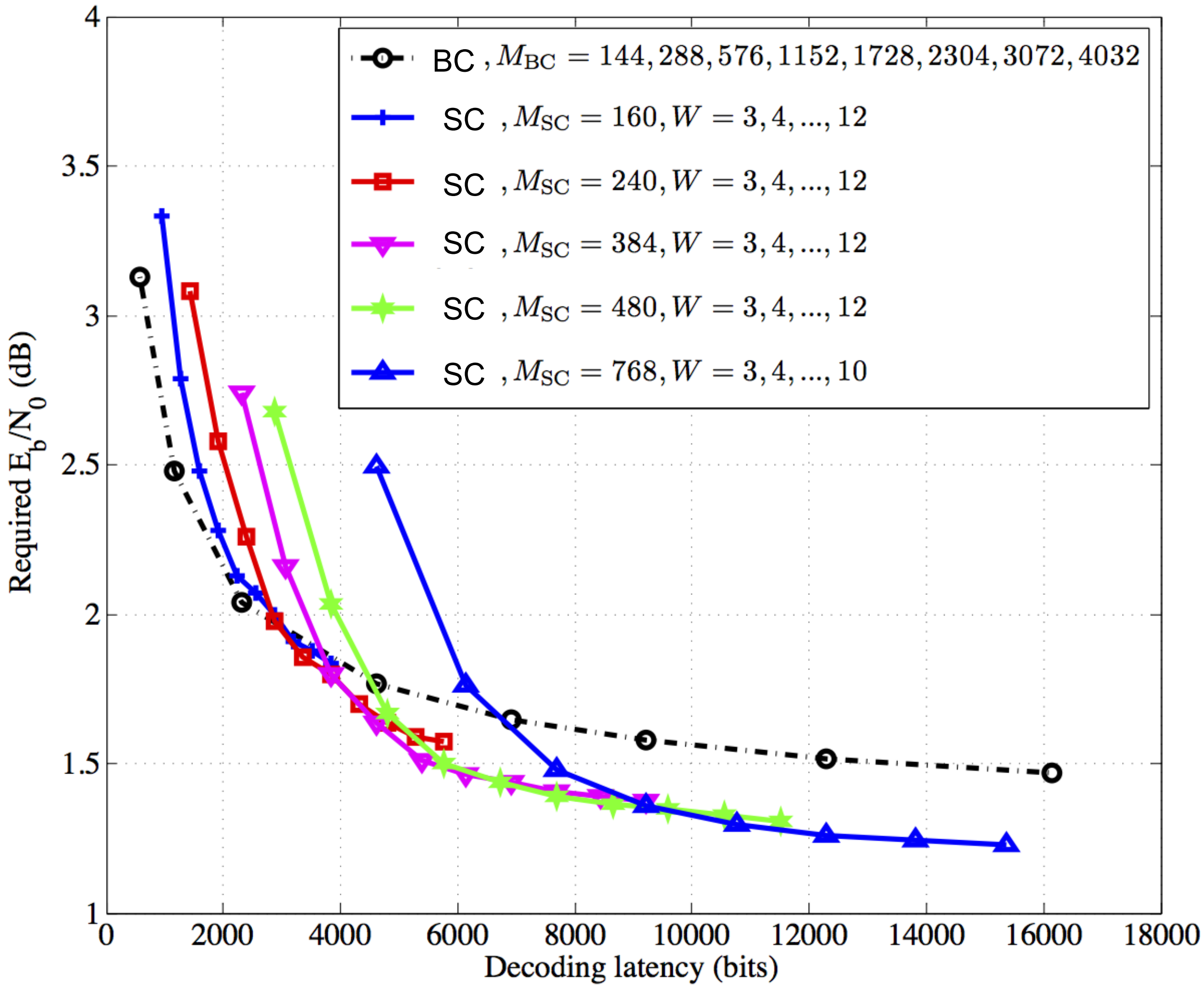


Latencies:
LDPC: $4M_{BC}$
SC-LDPC: $2M_{SC}W$

- decreases as W (and thus the latency) increases.
- does not decrease significantly beyond a certain W $(W \approx 10)$.

Equal Latency Comparison for (3,6)-Regular LDPC Codes

- Required E_b/N_0 to achieve a BER of 10^{-5} as a function of latency:

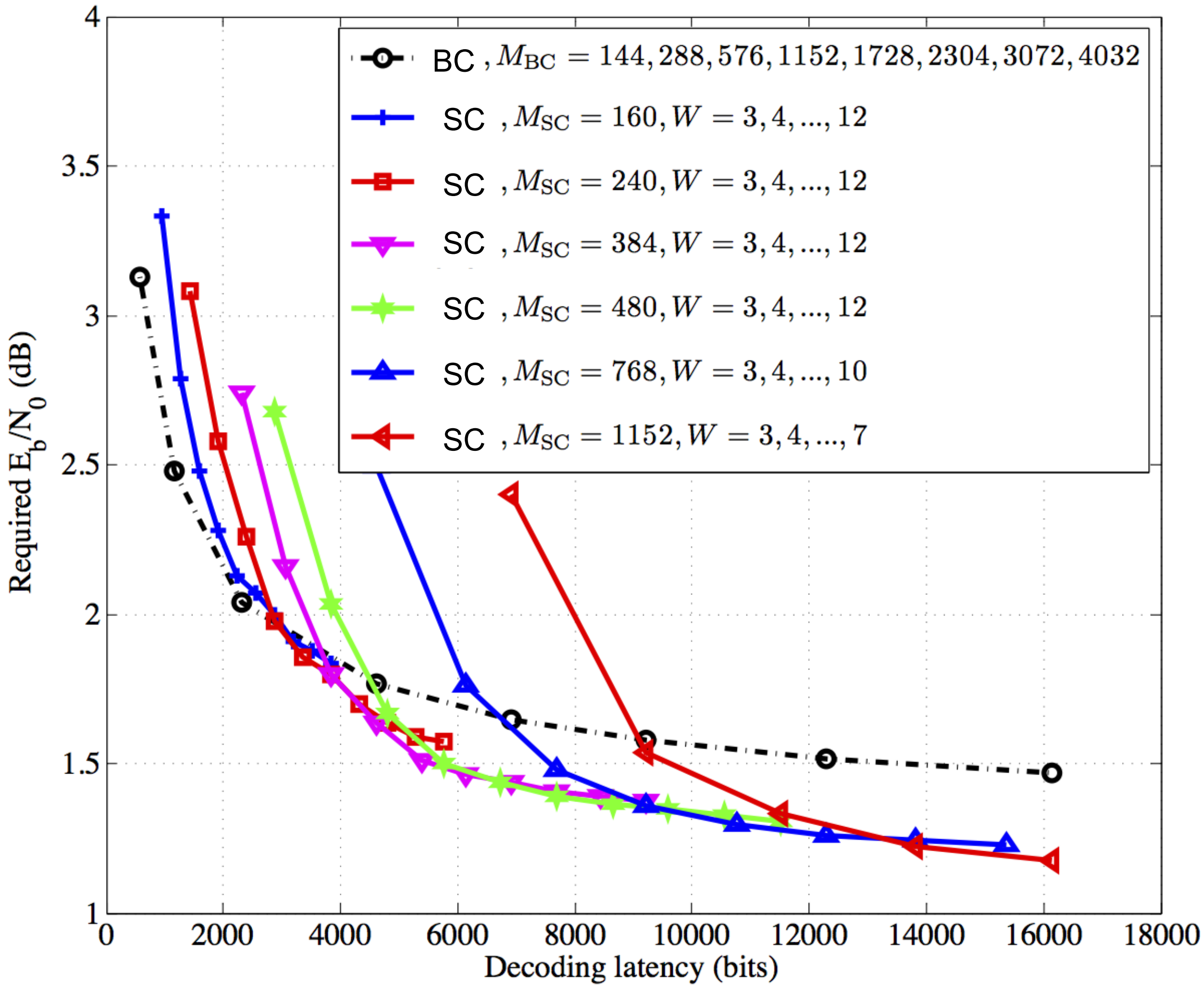


Latencies:
LDPC: $4M_{BC}$
SC-LDPC: $2M_{SC}W$

- ➡ decreases as W (and thus the latency) increases.
- ➡ does not decrease significantly beyond a certain W $(W \approx 10)$.

Equal Latency Comparison for (3,6)-Regular LDPC Codes

- Required E_b/N_0 to achieve a BER of 10^{-5} as a function of latency:



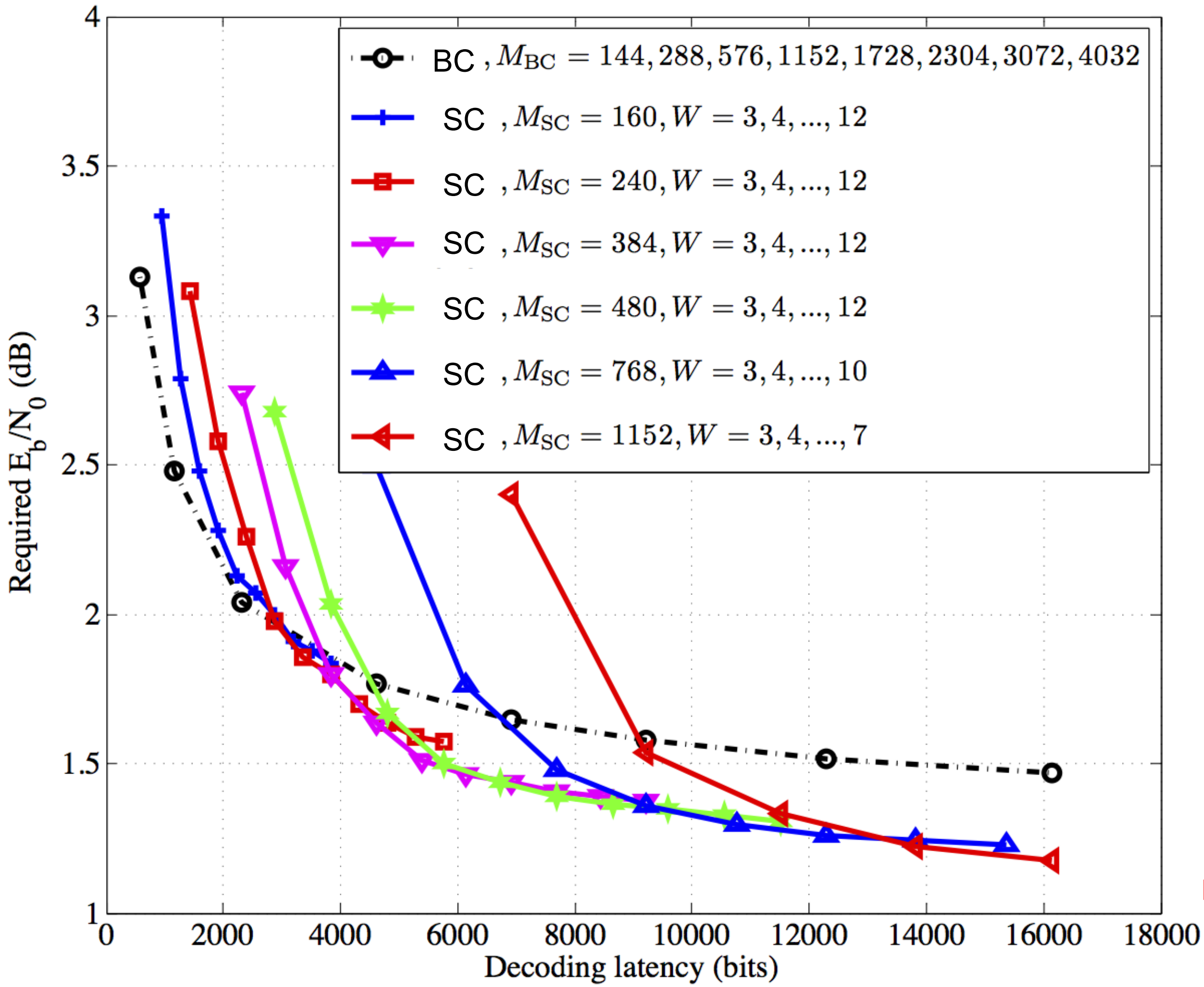
Latencies:
LDPC: $4M_{BC}$
SC-LDPC: $2M_{SC}W$

- decreases as W (and thus the latency) increases.
- does not decrease significantly beyond a certain W $(W \approx 10)$.

Equal Latency Comparison for (3,6)-Regular LDPC Codes

UNIVERSITY OF NOTRE DAME

- Required E_b/N_0 to achieve a BER of 10^{-5} as a function of latency:

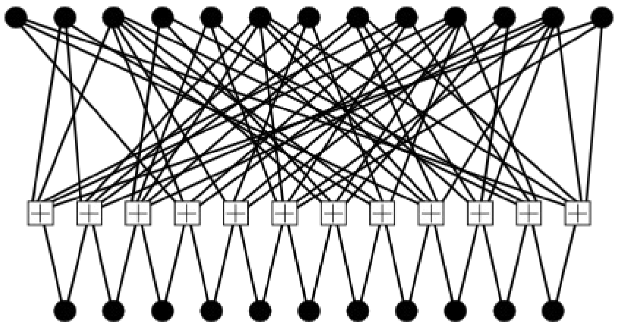


Latencies:
LDPC: $4M_{BC}$
SC-LDPC: $2M_{SC}W$

- ➡ decreases as W (and thus the latency) increases.
- ➡ does not decrease significantly beyond a certain W $(W \approx 10)$.
- When choosing parameters:
  - ➡ large M_{SC} improves code performance.
  - ➡ large W improves decoder performance.

Regular SC-LDPC Codes vs. Irregular LDPC-BCs

- Consider a comparison of a (3,6)-regular SC-LDPC code vs. an irregular-repeat-accumulate (IRA) LDPC-BC with optimized protograph taken from the WiMAX standard

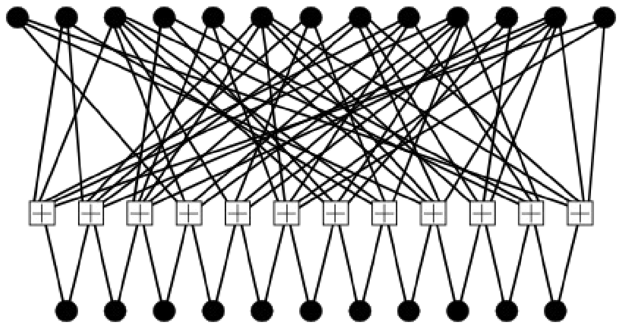


Ex: $M = 250$
$n = 24M = 6000$

Regular SC-LDPC Codes vs. Irregular LDPC-BCs

UNIVERSITY OF NOTRE DAME

- Consider a comparison of a (3,6)-regular SC-LDPC code vs. an irregular-repeat-accumulate (IRA) LDPC-BC with optimized protograph taken from the WiMAX standard



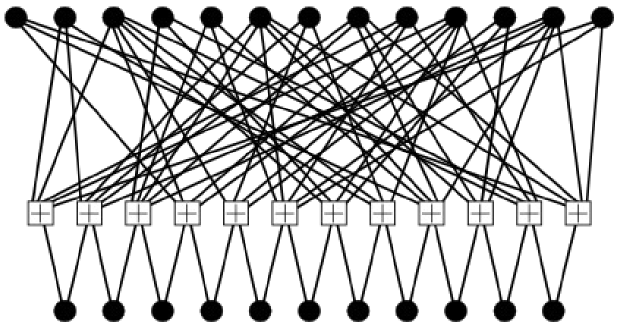
Ex: $M = 250$
$n = 24M = 6000$

- The IRA LDPC-BC ensemble has rate R=0.5, BEC threshold $\epsilon^* \approx 0.4489$, and AWGNC threshold $(E_b/N_0)^* \approx 0.8216$ dB.

Regular SC-LDPC Codes vs. Irregular LDPC-BCs

UNIVERSITY OF NOTRE DAME

- Consider a comparison of a (3,6)-regular SC-LDPC code vs. an irregular-repeat-accumulate (IRA) LDPC-BC with optimized protograph taken from the WiMAX standard



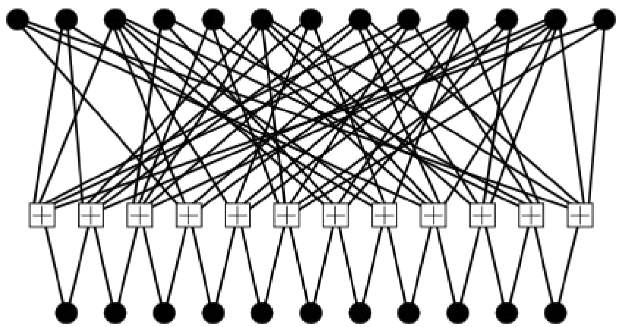
Ex: $M = 250$
$n = 24M = 6000$

- The IRA LDPC-BC ensemble has rate R=0.5, BEC threshold $\epsilon^* \approx 0.4489$, and AWGNC threshold $(E_b/N_0)^* \approx 0.8216$ dB.
- We compare this to a (3,6)-regular SC-LDPC code ensemble with L=50, R=0.49, and thresholds $\epsilon^* \approx 0.4881$ and $(E_b/N_0)^* \approx 0.4317$ dB.

Regular SC-LDPC Codes vs. Irregular LDPC-BCs

UNIVERSITY OF NOTRE DAME

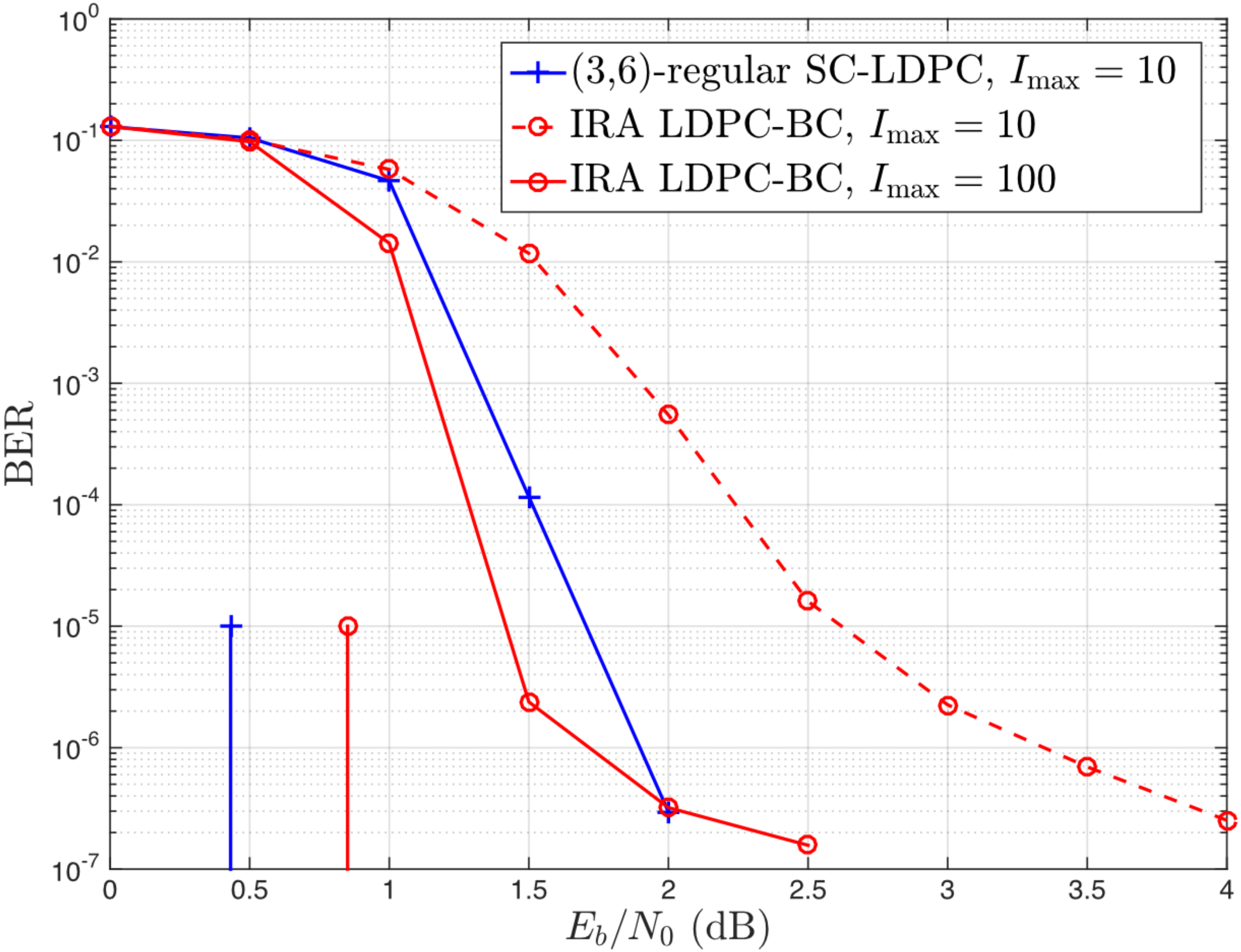
- Consider a comparison of a (3,6)-regular SC-LDPC code vs. an irregular-repeat-accumulate (IRA) LDPC-BC with optimized protograph taken from the WiMAX standard



Ex: $M = 250$
$n = 24M = 6000$

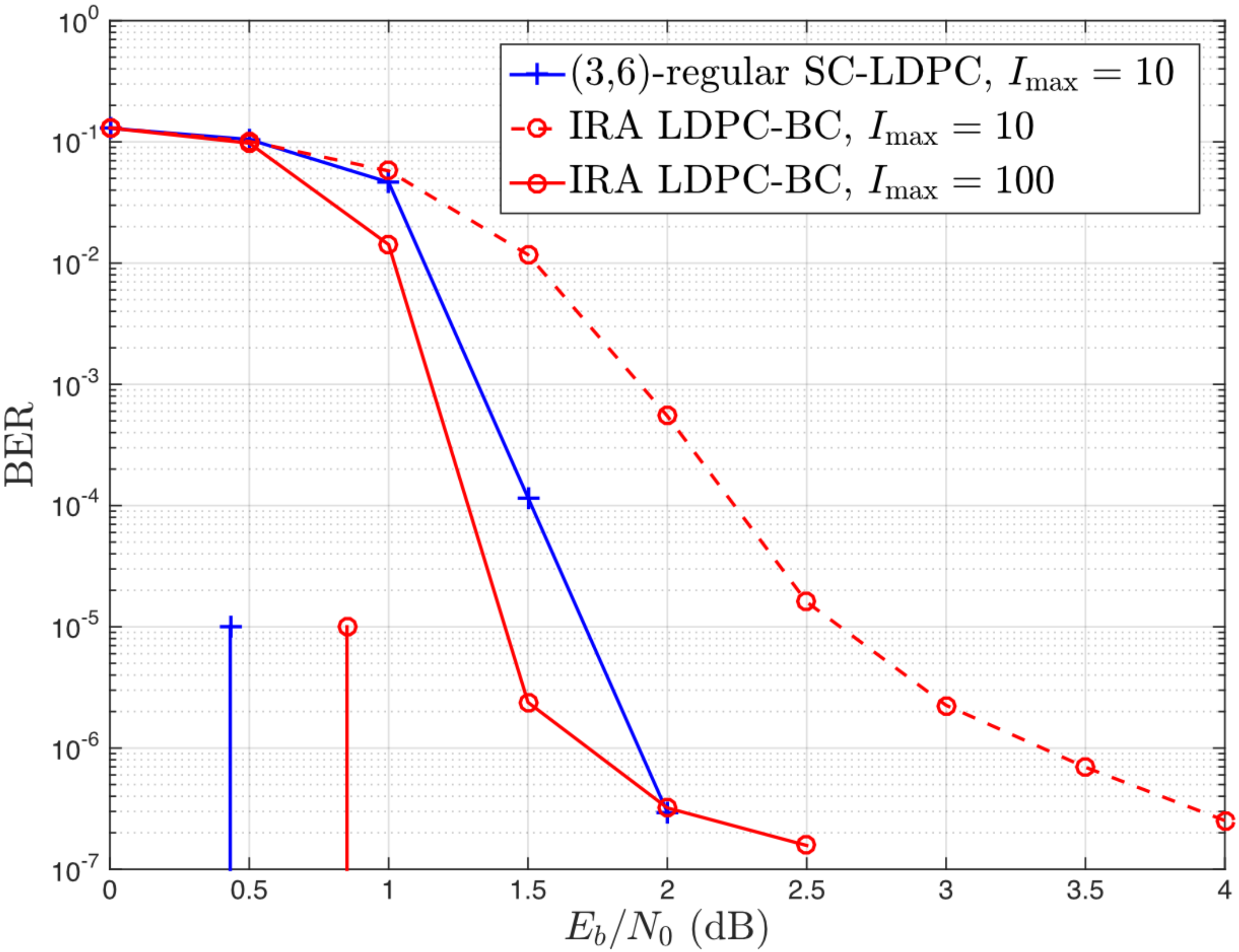
- The IRA LDPC-BC ensemble has rate R=0.5, BEC threshold $\epsilon^* \approx 0.4489$, and AWGNC threshold $(E_b/N_0)^* \approx 0.8216$ dB.
- We compare this to a (3,6)-regular SC-LDPC code ensemble with L=50, R=0.49, and thresholds $\epsilon^* \approx 0.4881$ and $(E_b/N_0)^* \approx 0.4317$ dB.
- For the SC-LDPC code, we choose W=6 and M=500 so that the latency of both codes is 6000 bits. (Since a code symbol is present in W=6 'windows', we allow fewer iterations per position for the SC-LDPC window decoder.)

Regular SC-LDPC Codes vs. Irregular LDPC-BCs



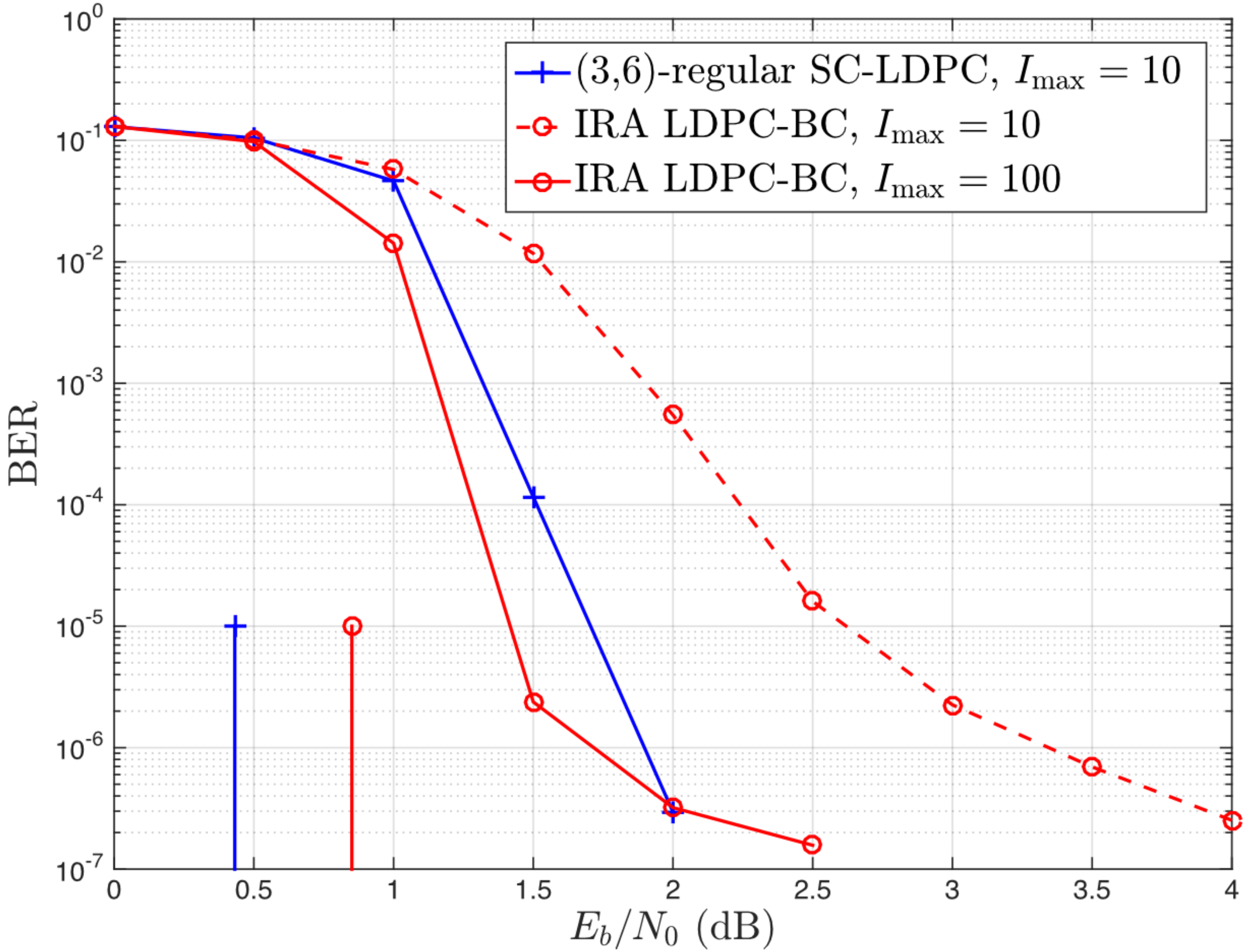
- Gaps to threshold will reduce with increasing latency

Regular SC-LDPC Codes vs. Irregular LDPC-BCs



- Gaps to threshold will reduce with increasing latency
- The asymptotically good regular SC-LDPC code shows no sign of an error floor

Regular SC-LDPC Codes vs. Irregular LDPC-BCs



- Gaps to threshold will reduce with increasing latency
- The asymptotically good regular SC-LDPC code shows no sign of an error floor
- The regular SC-LDPC code structure has implementation advantages

Implementation Aspects

- As a result of their capacity approaching performance and simple structure, regular SC-LDPC codes may be attractive for future coding standards. Several key features will require further investigation:
  - ➡ Hardware advantages of QC designs obtained by circulant liftings
  - ➡ Hardware advantages of the 'asymptotically-regular' structure
  - ➡ Design advantages of flexible frame length and flexible rate obtained by varying M, L, and/or puncturing

Implementation Aspects

- As a result of their capacity approaching performance and simple structure, regular SC-LDPC codes may be attractive for future coding standards. Several key features will require further investigation:
  - ➡ Hardware advantages of QC designs obtained by circulant liftings
  - ➡ Hardware advantages of the 'asymptotically-regular' structure
  - ➡ Design advantages of flexible frame length and flexible rate obtained by varying M, L, and/or puncturing
- Of particular importance for applications requiring extremely low decoded bit error rates (*e.g.*, optical communication, data storage) is an investigation of error floor issues related to **stopping sets**, **trapping sets**, and **absorbing sets**.

Conclusions

University of Notre Dame

- Spatially coupled LDPC code ensembles achieve **threshold saturation**, i.e., their iterative decoding thresholds (for large L and M) approach the MAP decoding thresholds of the underlying LDPC block code ensembles.

- The threshold saturation and linear minimum distance growth properties of (J,K)-regular SC-LDPC codes combine the best asymptotic features of both regular and irregular LDPC-BCs.

- With window decoding, SC-LDPC codes also compare favorably to LDPC-BCs in the finite-length regime, providing flexible tradeoffs between BER performance, decoding latency, and decoding complexity.

- SC-LDPC codes can be punctured to achieve robustly good performance over a wide variety of code rates.