Decoding Complexity of Long Polar and LDPC Codes

Felix Krieg, Marvin Rübenacke,
Andreas Zunker and Stephan ten Brink

Institute of Telecommunications, University of Stuttgart

IT-FuN Workshop Karlsruhe
2nd March 2026

University of Stuttgart
Institute of Telecommunications
Prof. Dr. Ing. Stephan ten Brink

Outline

Agenda

Basis of This Work

- Coding group at INUE, working on, e.g.,
  - Polar & LDPC Coding for Wireless
  - Staircase-like/Spatially Coupled Codes for Optical
  - Ensemble Decoding
  - Expectation Propagation
- Material taken from our recent Globecom-paper
  - Krieg et al., "Long Polar vs. LDPC Codes under Complexity-Constrained Decoding," IEEE Globecom 2025 Workshop on Channel Coding beyond 5G, Taiwan, December 2025 [2508.05485]; not yet on IEEE Xplore...
- Contact the authors
  - krieg@inue.uni-stuttgart.de
  - tenbrink@inue.uni-stuttgart.de

Motivation

New design goals in 6G: **sustainability** and **efficiency**
The new standards aim to include even more device types

➜ Redcap devices

With FEC as the workhorse of the physical layer, new challenges arise:

- Reduced chip area
- Low energy consumption of the decoder

➜ Powerful yet simple decoders will be required

Motivation

New design goals in 6G: **sustainability** and **efficiency**
The new standards aim to include even more device types

➜ Redcap devices

With FEC as the workhorse of the physical layer, new challenges arise:

- Reduced chip area
- Low energy consumption of the decoder

➜ Powerful yet simple decoders will be required

Motivation

New design goals in 6G: **sustainability** and **efficiency**
The new standards aim to include even more device types

➜ Redcap devices

With FEC as the workhorse of the physical layer, new challenges arise:

- Reduced chip area
- Low energy consumption of the decoder

➜ Powerful yet simple decoders will be required

Possible Code Families

The codes also should be easy to standardize:

- We don't expect any major inventions, such as a new family of codes, for 6G
- Rely on the trusted 5G codes:

➜ LDPC codes, and/or

➜ Polar codes

Possible Code Families

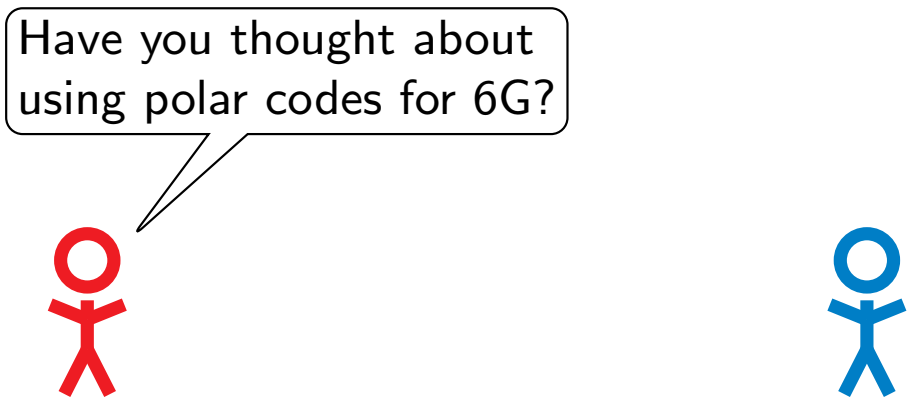
The codes also should be easy to standardize:

- We don't expect any major inventions, such as a new family of codes, for 6G
- Rely on the trusted 5G codes:

➜ LDPC codes, and/or

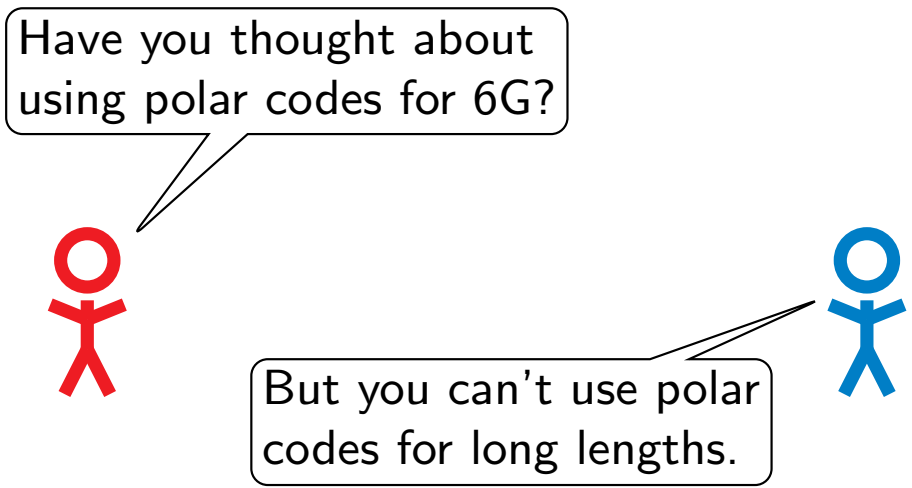
➜ Polar codes

The Common Conception



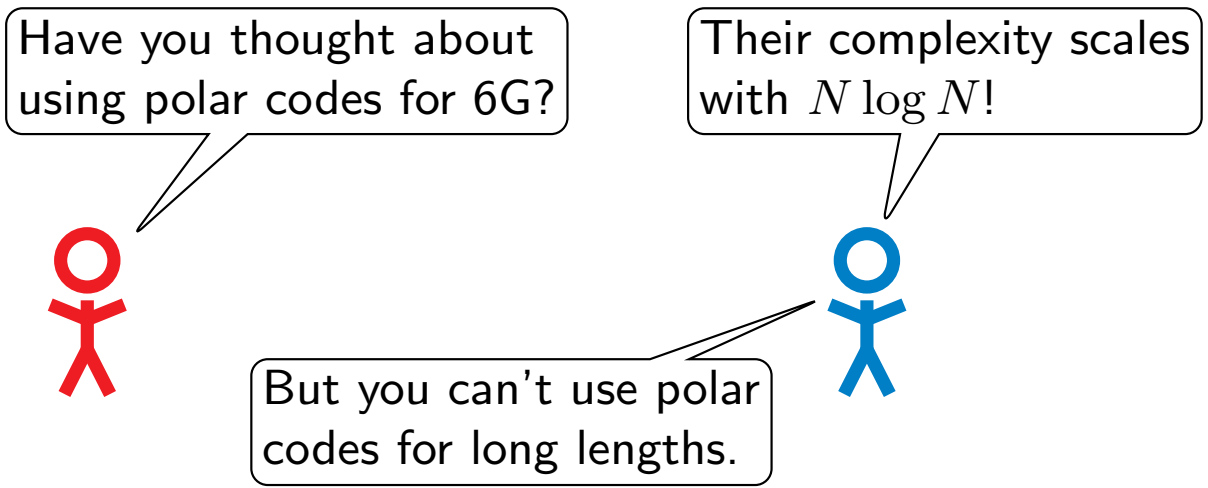
University of Stuttgart
Institute of Telecommunications

The Common Conception



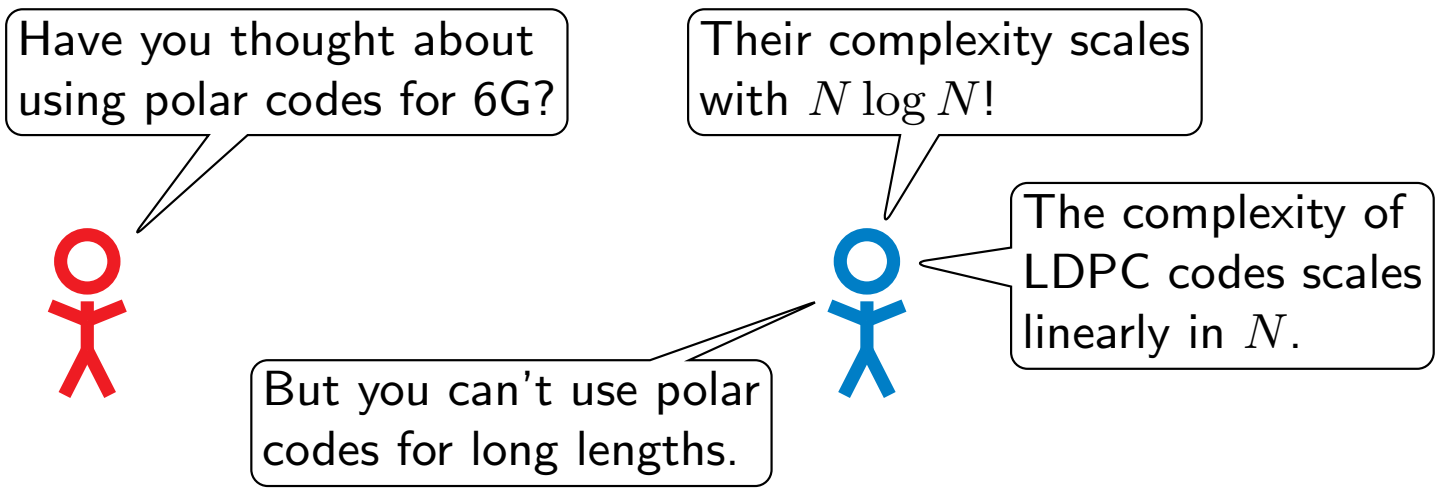
University of Stuttgart
Institute of Telecommunications

The Common Conception

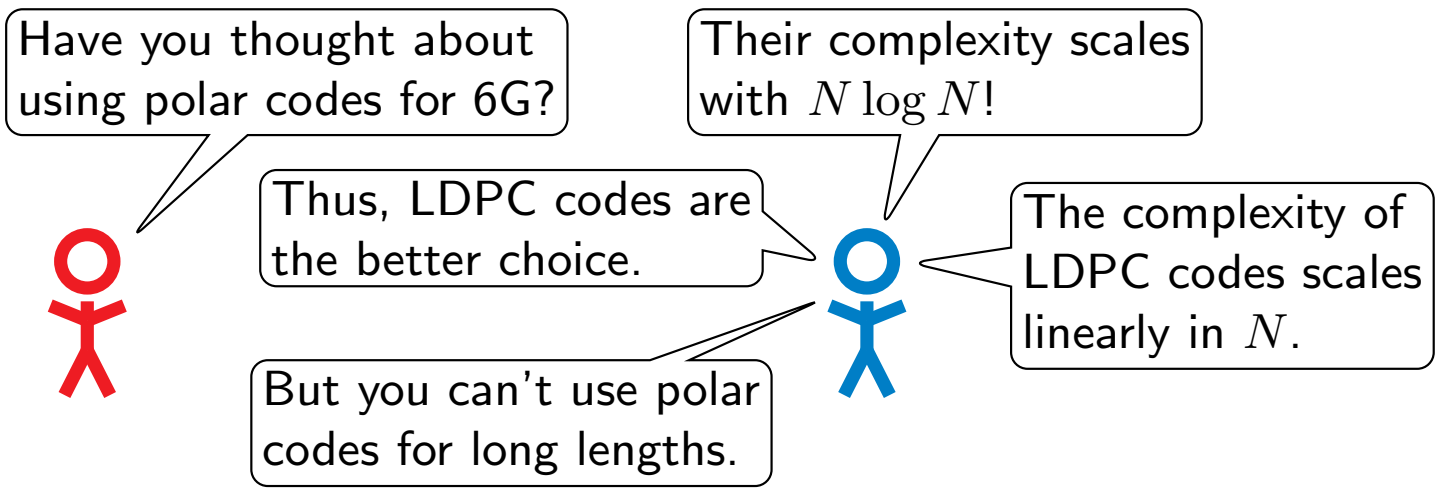


University of Stuttgart
Institute of Telecommunications

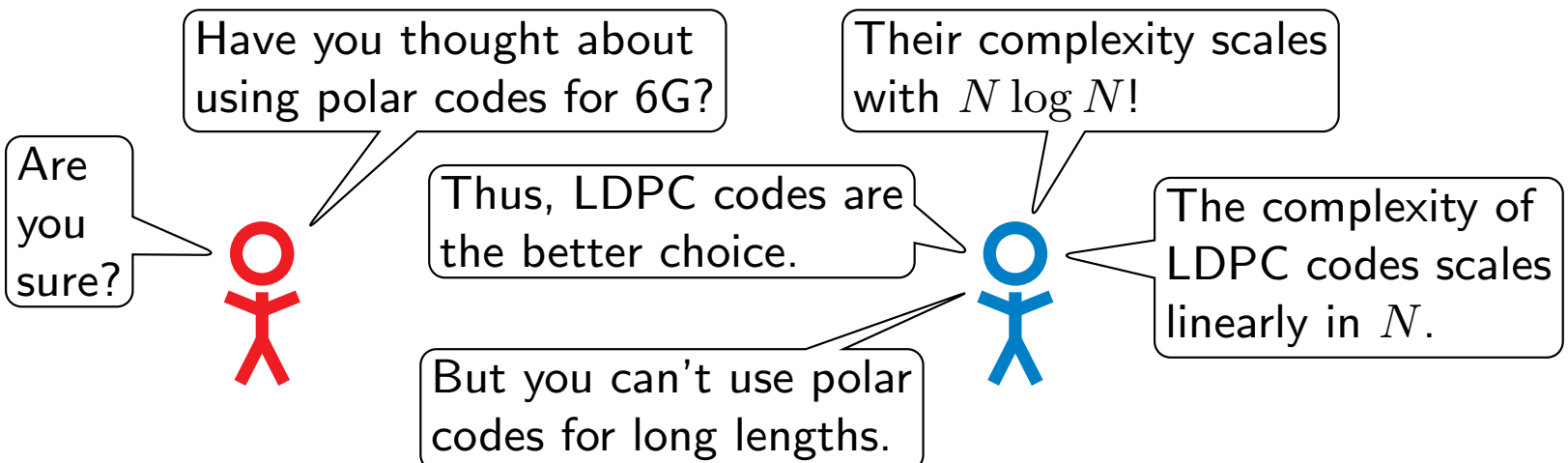
The Common Conception



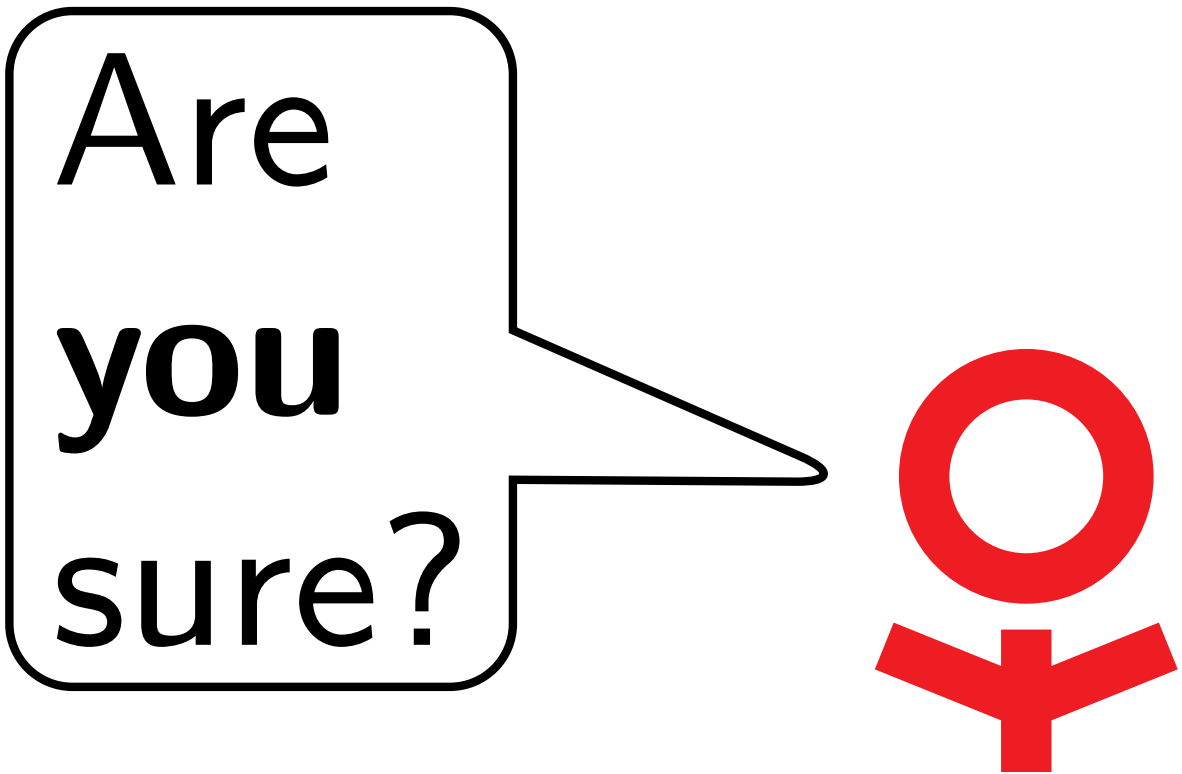
The Common Conception



The Common Conception



The Common Conception



Goal

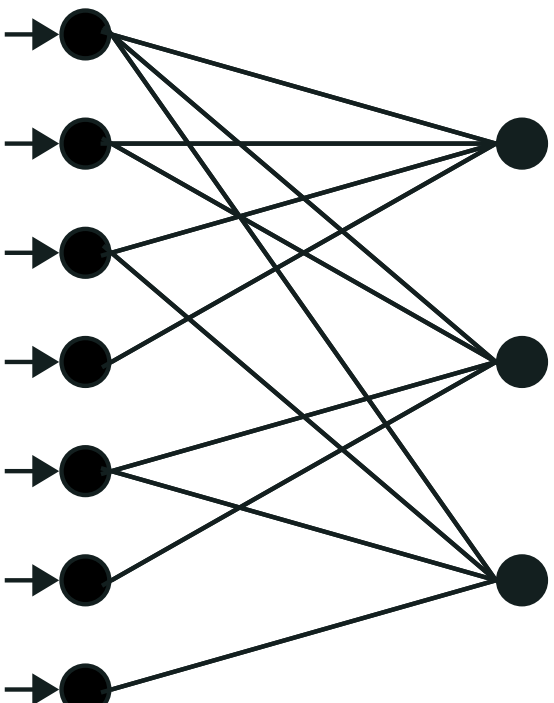
- There is an overwhelming consensus that polar codes do not scale well to longer lengths

➜ Let's make a fair comparison!

Agenda

LDPC Codes & BP Decoding

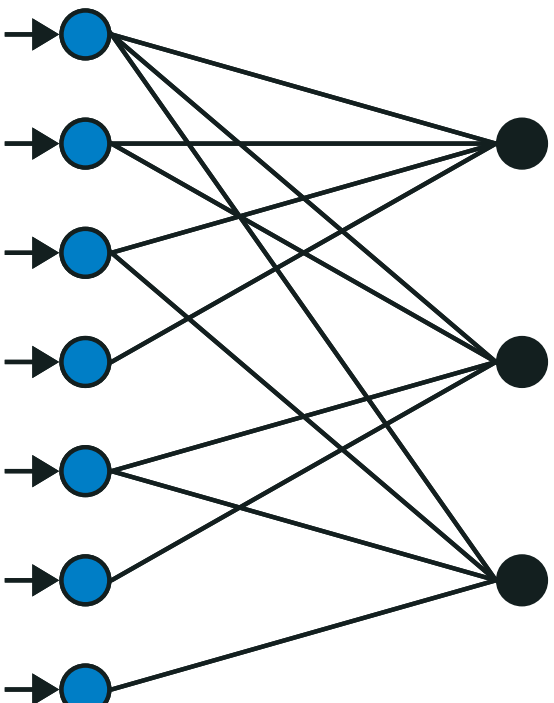
- Represent $\boldsymbol{H}$ as a bipartite graph
- Constraints between variable nodes are considered in check nodes
- Focusing on one check node $j = 3$:
  - The variable nodes V_j propagate their reliabilites to the check node
  - The check node calculates $r_{i \leftarrow j} = \boxplus_{i' \in V_j \setminus i} q_{i' \rightarrow j}$ and propagates back to the variable nodes
  - The check node update can be approximated



University of Stuttgart
Institute of Telecommunications

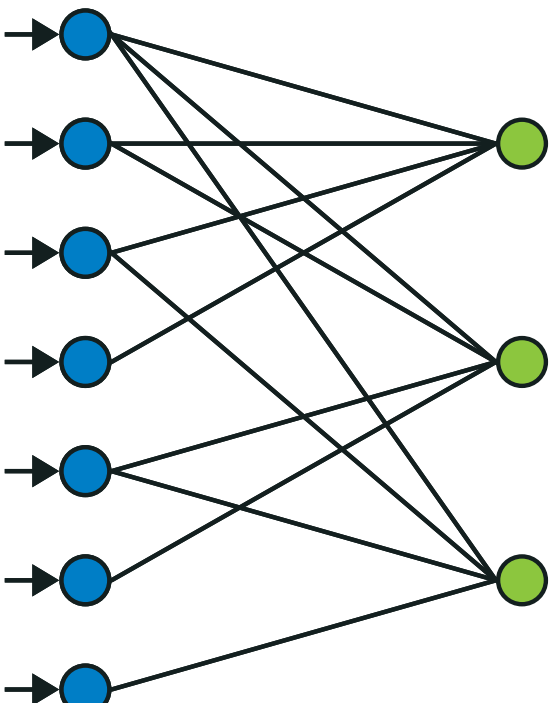
LDPC Codes & BP Decoding

- Represent $\boldsymbol{H}$ as a bipartite graph
- Constraints between variable nodes are considered in check nodes
- Focusing on one check node $j = 3$:
  - The variable nodes V_j propagate their reliabilites to the check node
  - The check node calculates $r_{i \leftarrow j} = \boxplus_{i' \in V_j \setminus i} q_{i' \rightarrow j}$ and propagates back to the variable nodes
  - The check node update can be approximated



LDPC Codes & BP Decoding

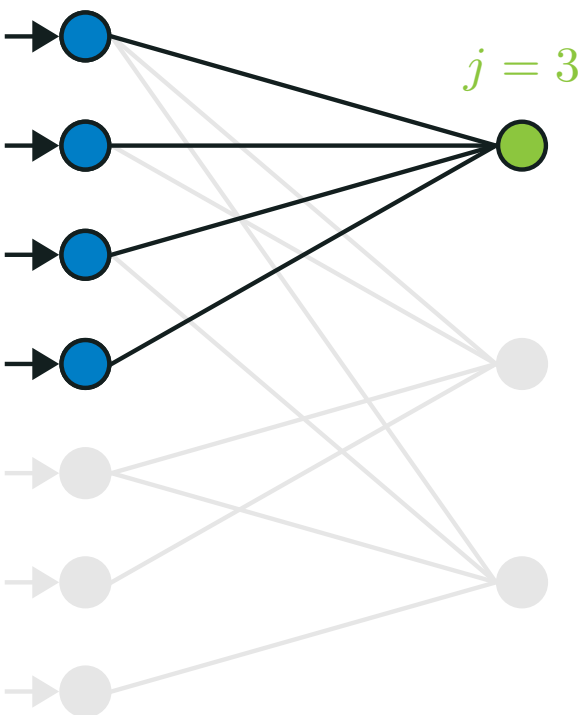
- Represent $\boldsymbol{H}$ as a bipartite graph
- Constraints between variable nodes are considered in check nodes
- Focusing on one check node $j = 3$:
  - The variable nodes V_j propagate their reliabilites to the check node
  - The check node calculates $r_{i \leftarrow j} = \boxplus_{i' \in V_j \setminus i} q_{i' \rightarrow j}$ and propagates back to the variable nodes
  - The check node update can be approximated



University of Stuttgart
Institute of Telecommunications

LDPC Codes & BP Decoding

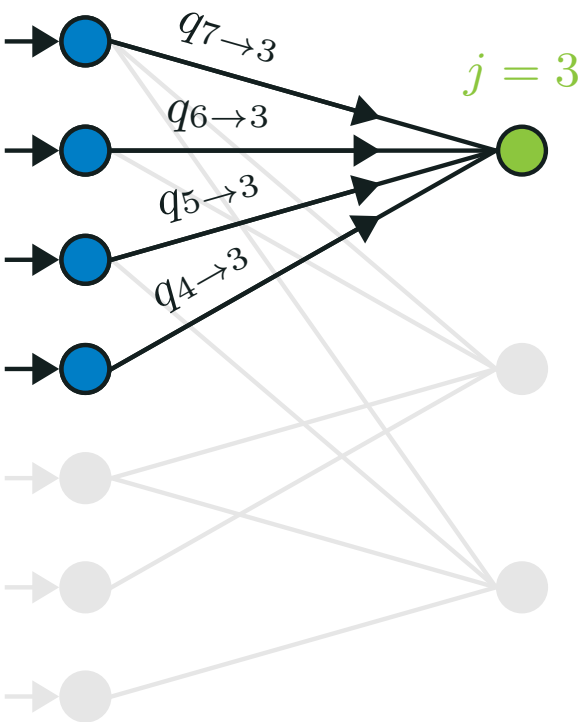
- Represent $\boldsymbol{H}$ as a bipartite graph
- Constraints between variable nodes are considered in check nodes
- Focusing on one check node $j = 3$:
  - The variable nodes V_j propagate their reliabilites to the check node
  - The check node calculates $r_{i \leftarrow j} = \boxplus_{i' \in V_j \setminus i} q_{i' \rightarrow j}$ and propagates back to the variable nodes
  - The check node update can be approximated



University of Stuttgart
Institute of Telecommunications

LDPC Codes & BP Decoding

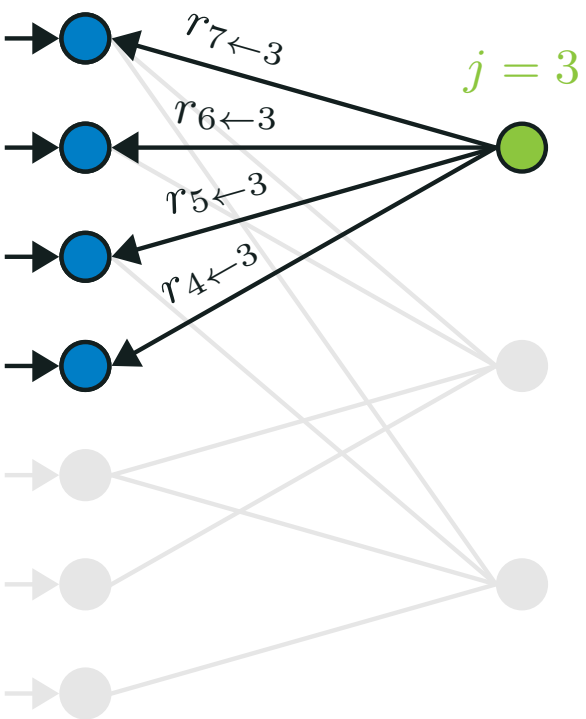
- Represent $\boldsymbol{H}$ as a bipartite graph
- Constraints between variable nodes are considered in check nodes
- Focusing on one check node $j = 3$:
  - The variable nodes V_j propagate their reliabilites to the check node
  - The check node calculates $r_{i \leftarrow j} = \boxplus_{i' \in V_j \setminus i} q_{i' \rightarrow j}$ and propagates back to the variable nodes
  - The check node update can be approximated



University of Stuttgart
Institute of Telecommunications

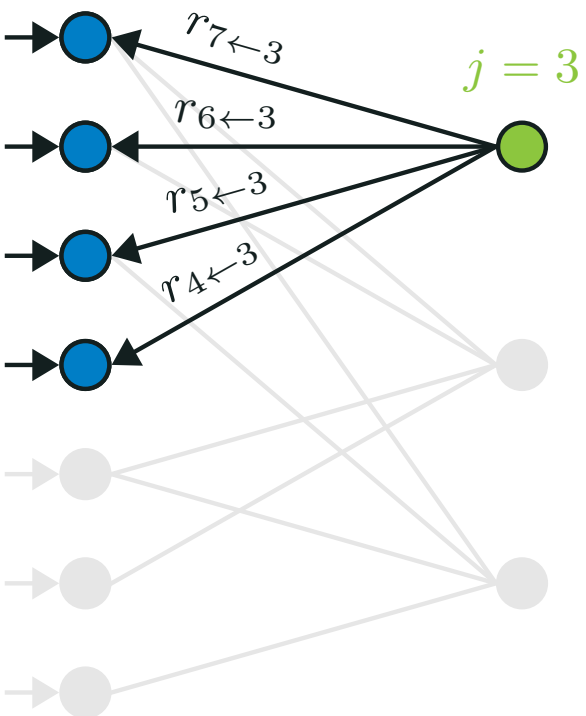
LDPC Codes & BP Decoding

- Represent $\boldsymbol{H}$ as a bipartite graph
- Constraints between variable nodes are considered in check nodes
- Focusing on one check node $j = 3$:
  - The variable nodes V_j propagate their reliabilites to the check node
  - The check node calculates $r_{i \leftarrow j} = \boxplus_{i' \in V_j \setminus i} q_{i' \rightarrow j}$ and propagates back to the variable nodes
  - The check node update can be approximated



LDPC Codes & BP Decoding

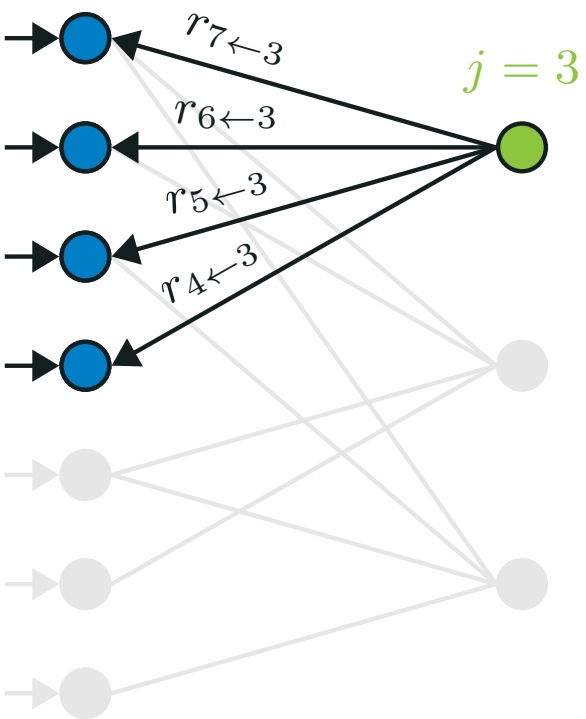
- Represent $\boldsymbol{H}$ as a bipartite graph
- Constraints between variable nodes are considered in check nodes
- Focusing on one check node $j = 3$:
  - The variable nodes V_j propagate their reliabilites to the check node
  - The check node calculates $r_{i \leftarrow j} = \boxplus_{i' \in V_j \setminus i} q_{i' \rightarrow j}$ and propagates back to the variable nodes
  - The check node update can be approximated



University of Stuttgart
Institute of Telecommunications

LDPC Codes & BP Decoding (cont.)

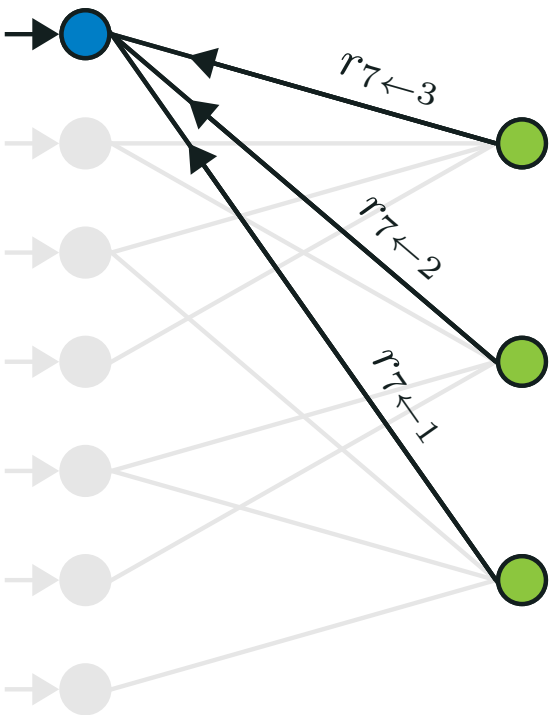
- The variable nodes combine the incoming messages of the check nodes
- Focusing on one variable node
  - The variable node combines the incoming messages $q_{i\to j} = y_i + \sum_{j' \in C_i \setminus j} r_{i \leftarrow j'}$
- The BP algorithm iterates between variable node and check node updates for a fixed number of iterations
- Layered min-sum (LMS) decoding processes CN subsets one at a time rather than all CNs simultaneously



University of Stuttgart
Institute of Telecommunications

LDPC Codes & BP Decoding (cont.)

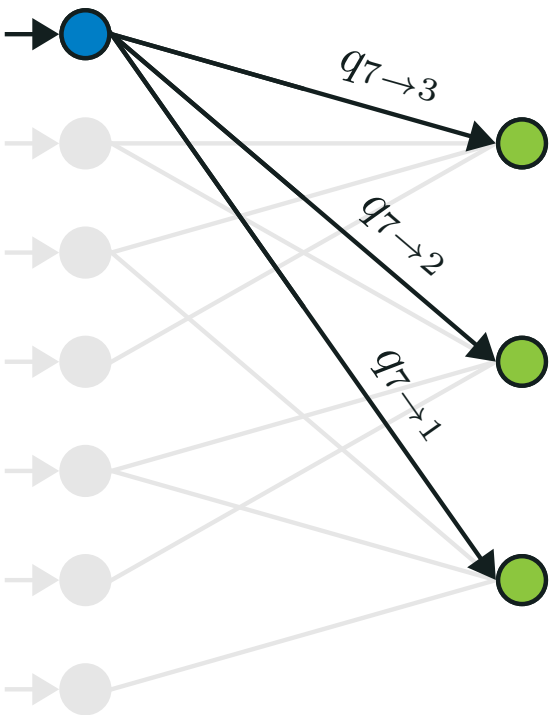
- The variable nodes combine the incoming messages of the check nodes
- Focusing on one variable node
  - The variable node combines the incoming messages $q_{i\to j} = y_i + \sum_{j' \in C_i \setminus j} r_{i \leftarrow j'}$
- The BP algorithm iterates between variable node and check node updates for a fixed number of iterations
- Layered min-sum (LMS) decoding processes CN subsets one at a time rather than all CNs simultaneously



University of Stuttgart
Institute of Telecommunications

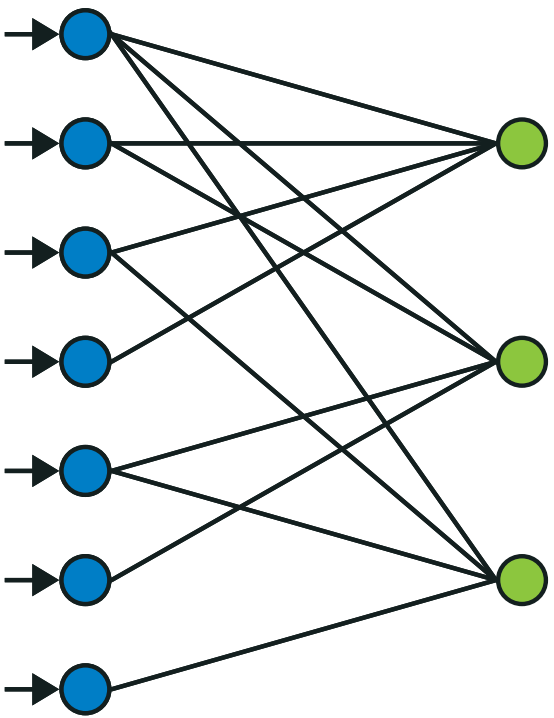
LDPC Codes & BP Decoding (cont.)

- The variable nodes combine the incoming messages of the check nodes
- Focusing on one variable node
  - The variable node combines the incoming messages
    $q_{i\to j} = y_i + \sum_{j'\in C_i\setminus j} r_{i\leftarrow j'}$
- The BP algorithm iterates between variable node and check node updates for a fixed number of iterations
- Layered min-sum (LMS) decoding processes CN subsets one at a time rather than all CNs simultaneously



LDPC Codes & BP Decoding (cont.)

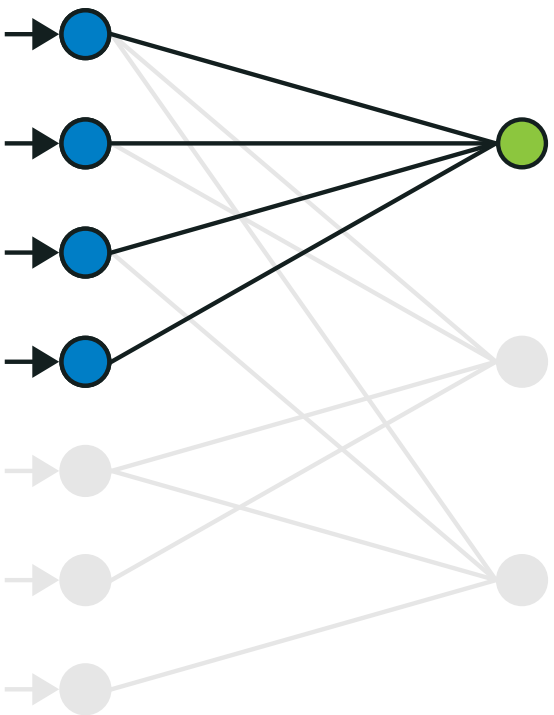
- The variable nodes combine the incoming messages of the check nodes
- Focusing on one variable node
  - The variable node combines the incoming messages
    $q_{i\to j} = y_i + \sum_{j' \in C_i \setminus j} r_{i \leftarrow j'}$
- The BP algorithm iterates between variable node and check node updates for a fixed number of iterations
- Layered min-sum (LMS) decoding processes CN subsets one at a time rather than all CNs simultaneously



University of Stuttgart
Institute of Telecommunications

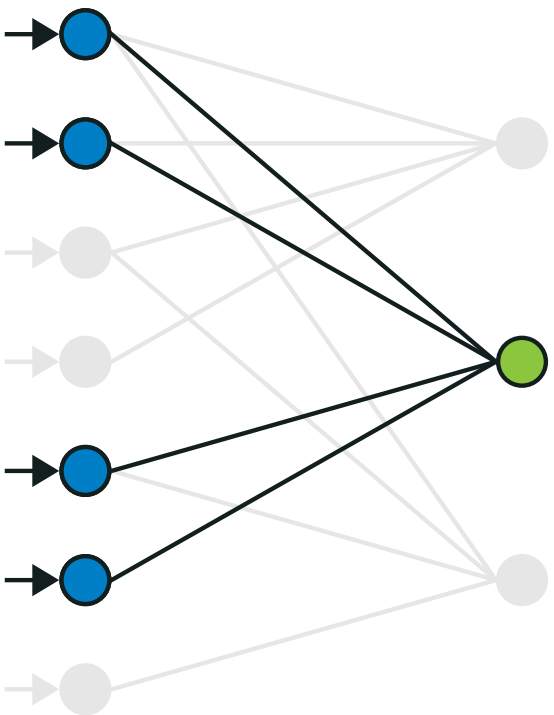
LDPC Codes & BP Decoding (cont.)

- The variable nodes combine the incoming messages of the check nodes
- Focusing on one variable node
  - The variable node combines the incoming messages
    $q_{i\to j} = y_i + \sum_{j' \in C_i \setminus j} r_{i \leftarrow j'}$
- The BP algorithm iterates between variable node and check node updates for a fixed number of iterations
- Layered min-sum (LMS) decoding processes CN subsets one at a time rather than all CNs simultaneously



LDPC Codes & BP Decoding (cont.)

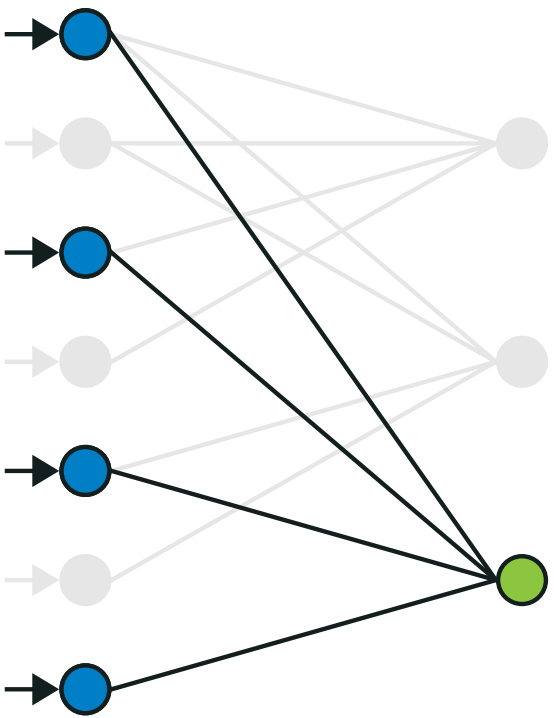
- The variable nodes combine the incoming messages of the check nodes
- Focusing on one variable node
  - The variable node combines the incoming messages
    $q_{i\to j} = y_i + \sum_{j' \in C_i \setminus j} r_{i \leftarrow j'}$
- The BP algorithm iterates between variable node and check node updates for a fixed number of iterations
- Layered min-sum (LMS) decoding processes CN subsets one at a time rather than all CNs simultaneously



University of Stuttgart
Institute of Telecommunications

LDPC Codes & BP Decoding (cont.)

- The variable nodes combine the incoming messages of the check nodes
- Focusing on one variable node
  - The variable node combines the incoming messages $q_{i\to j} = y_i + \sum_{j' \in C_i \setminus j} r_{i \leftarrow j'}$
- The BP algorithm iterates between variable node and check node updates for a fixed number of iterations
- Layered min-sum (LMS) decoding processes CN subsets one at a time rather than all CNs simultaneously



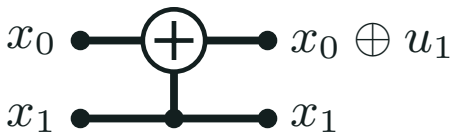
University of Stuttgart
Institute of Telecommunications

Polar Codes & SC Decoding

- Main building block: 2×2 kernel
- Frozen bits $(N - K)$ are always 0
- Remaining bits carry information
- To decode:
  - Decode first channel(s)
  - Decide at the message side
  - Use this decision to decode next channels

Polar Codes & SC Decoding

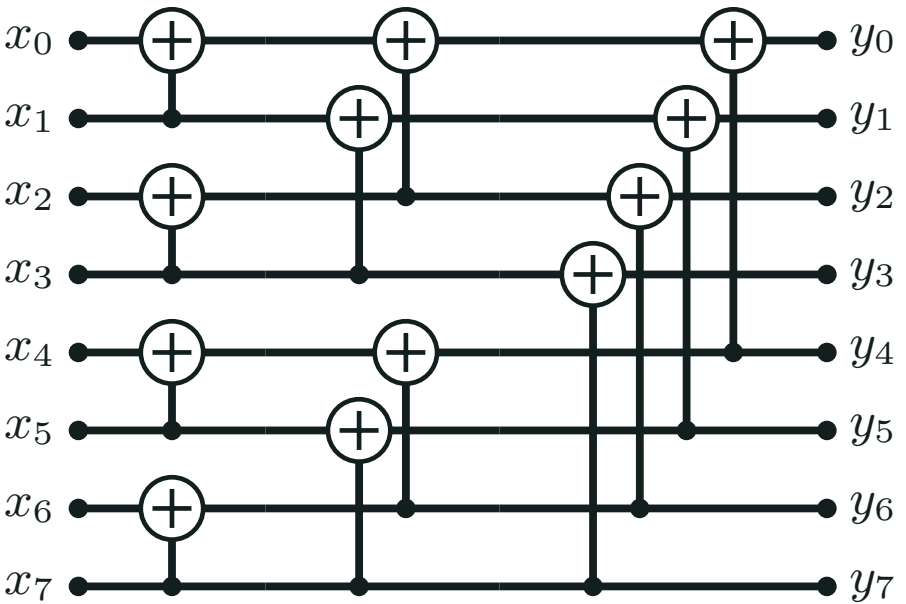
- Main building block: 2×2 kernel
- Frozen bits $(N - K)$ are always 0
- Remaining bits carry information
- To decode:
  - Decode first channel(s)
  - Decide at the message side
  - Use this decision to decode next channels



University of Stuttgart
Institute of Telecommunications

Polar Codes & SC Decoding

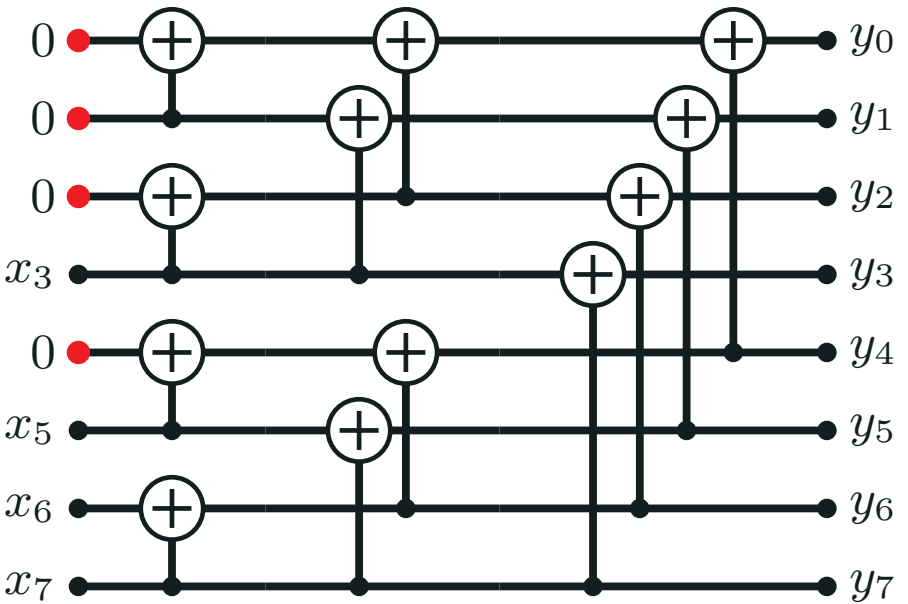
- Main building block: 2×2 kernel
- Frozen bits $(N - K)$ are always 0
- Remaining bits carry information
- To decode:
  - Decode first channel(s)
  - Decide at the message side
  - Use this decision to decode next channels



University of Stuttgart
Institute of Telecommunications

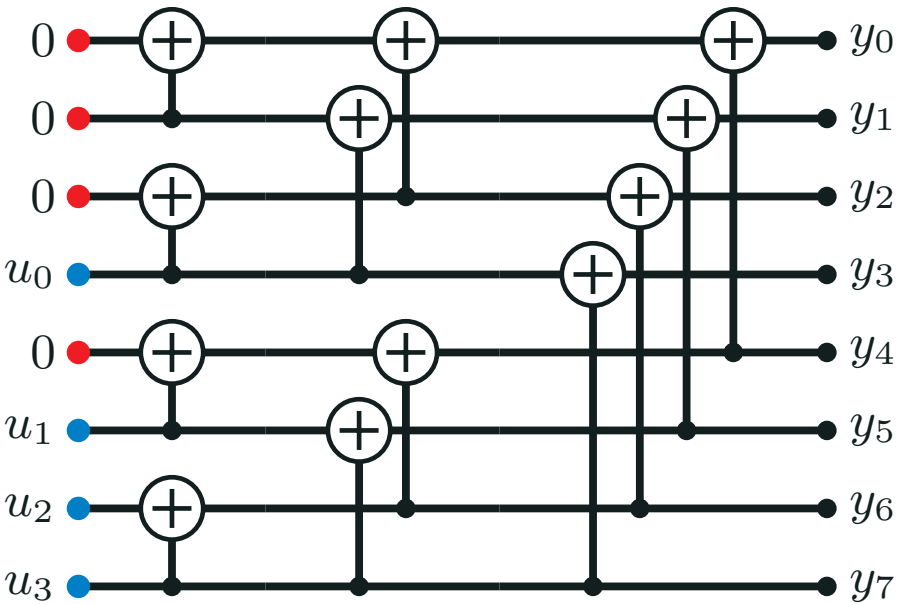
Polar Codes & SC Decoding

- Main building block: 2×2 kernel
- Frozen bits $(N - K)$ are always 0
- Remaining bits carry information
- To decode:
  - Decode first channel(s)
  - Decide at the message side
  - Use this decision to decode next channels



Polar Codes & SC Decoding

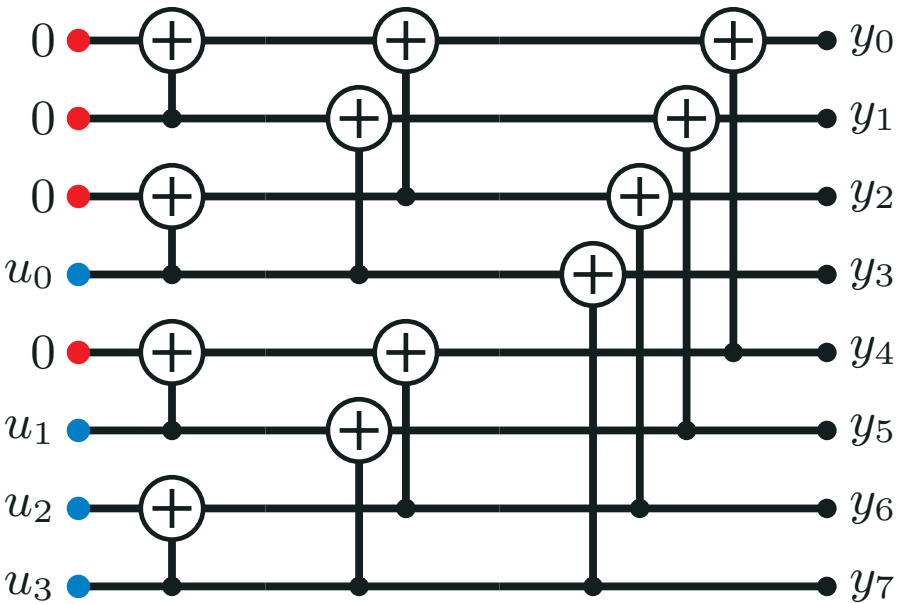
- Main building block: 2×2 kernel
- Frozen bits $(N - K)$ are always 0
- Remaining bits carry information
- To decode:
  - Decode first channel(s)
  - Decide at the message side
  - Use this decision to decode next channels



University of Stuttgart
Institute of Telecommunications

Polar Codes & SC Decoding

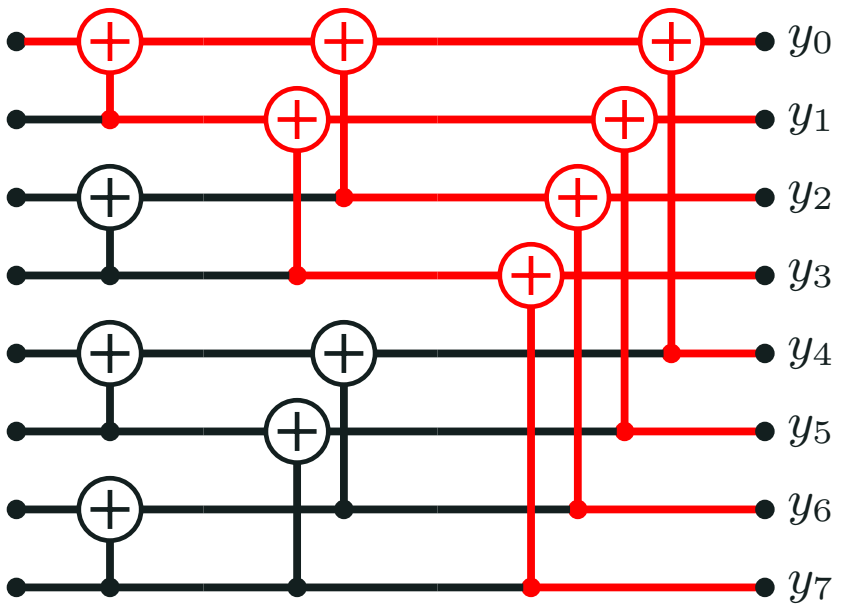
- Main building block: 2×2 kernel
- Frozen bits $(N - K)$ are always 0
- Remaining bits carry information
- To decode:
  - Decode first channel(s)
  - Decide at the message side
  - Use this decision to decode next channels



University of Stuttgart
Institute of Telecommunications

Polar Codes & SC Decoding

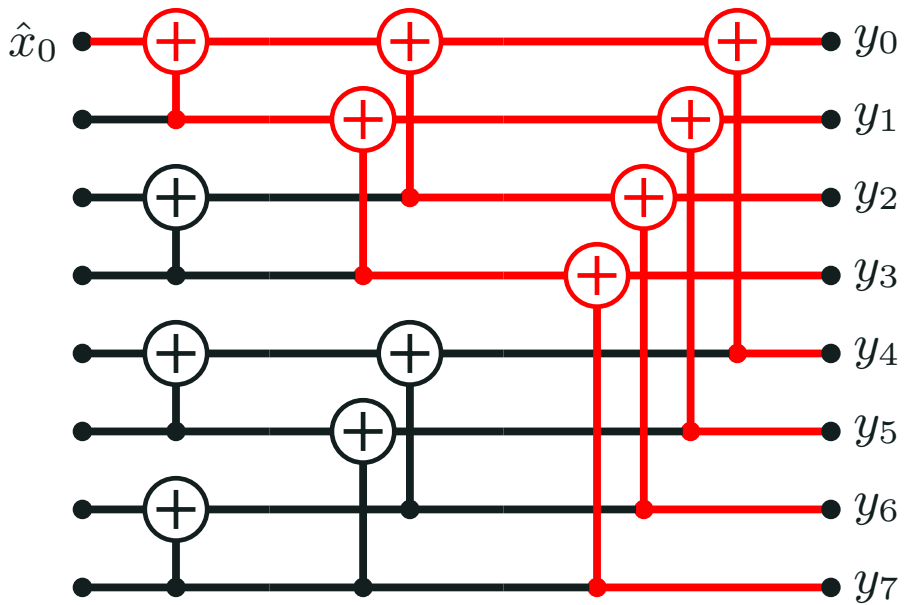
- Main building block: 2×2 kernel
- Frozen bits $(N - K)$ are always 0
- Remaining bits carry information
- To decode:
  - Decode first channel(s)
  - Decide at the message side
  - Use this decision to decode next channels



University of Stuttgart
Institute of Telecommunications

Polar Codes & SC Decoding

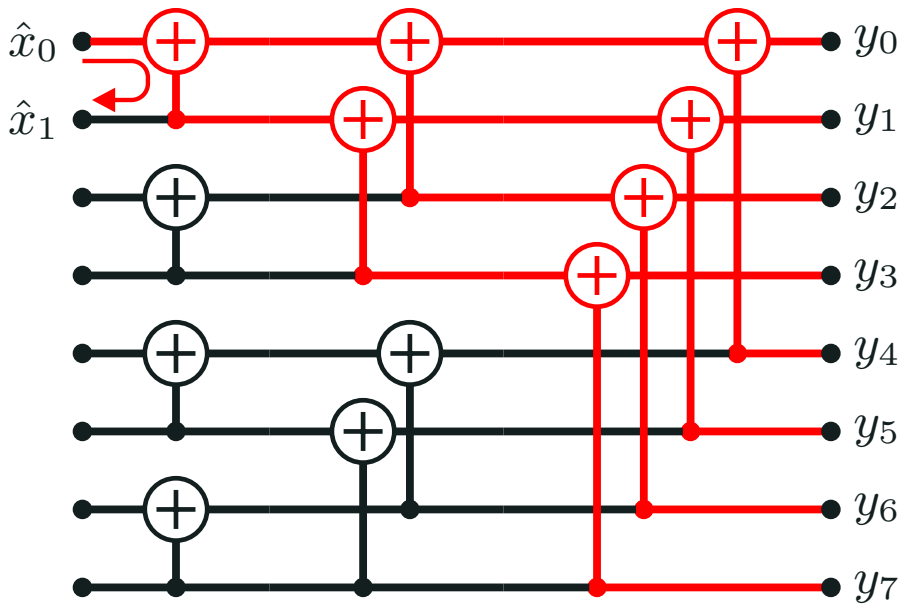
- Main building block: 2×2 kernel
- Frozen bits $(N - K)$ are always 0
- Remaining bits carry information
- To decode:
  - Decode first channel(s)
  - Decide at the message side
  - Use this decision to decode next channels



University of Stuttgart
Institute of Telecommunications

Polar Codes & SC Decoding

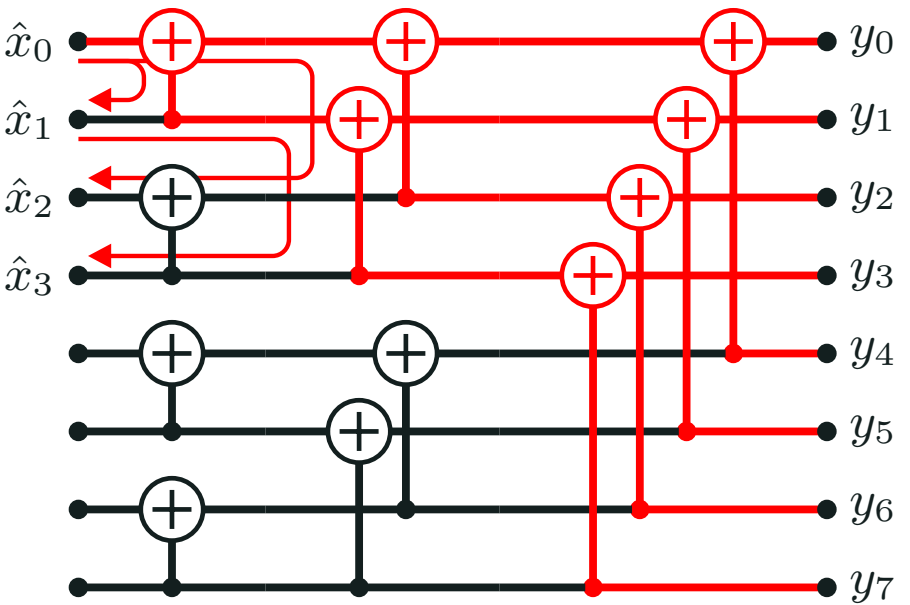
- Main building block: 2×2 kernel
- Frozen bits $(N - K)$ are always 0
- Remaining bits carry information
- To decode:
  - Decode first channel(s)
  - Decide at the message side
  - Use this decision to decode next channels



University of Stuttgart
Institute of Telecommunications

Polar Codes & SC Decoding

- Main building block: 2×2 kernel
- Frozen bits $(N - K)$ are always 0
- Remaining bits carry information
- To decode:
  - Decode first channel(s)
  - Decide at the message side
  - Use this decision to decode next channels

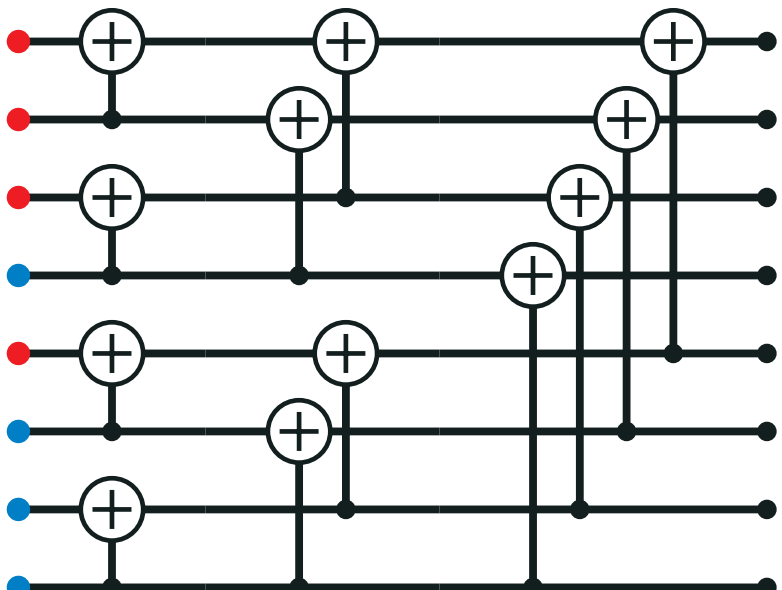


University of Stuttgart
Institute of Telecommunications

Simplified SC Decoding

- Certain combinations of frozen and information bits lead to known subcodes
  - Repetition codes
  - Single parity check codes
  - Rate-0 and rate-1 codes
- These codes can be decoded in one operation
- Simplified SC decoding leverages this to reduce the complexity

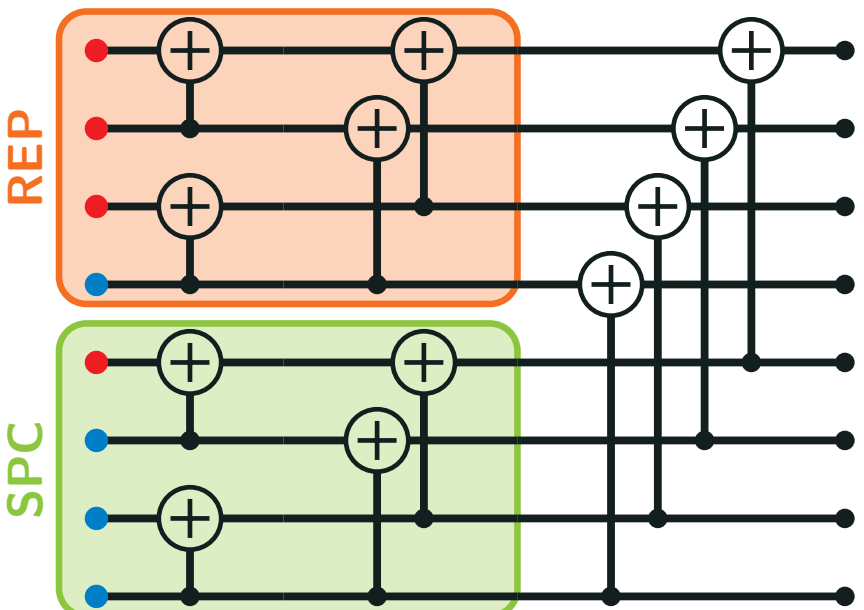
➜ This does not degrade decoding performance!



Simplified SC Decoding

- Certain combinations of frozen and information bits lead to known subcodes
  - Repetition codes
  - Single parity check codes
  - Rate-0 and rate-1 codes
- These codes can be decoded in one operation
- Simplified SC decoding leverages this to reduce the complexity

➜ This does not degrade decoding performance!

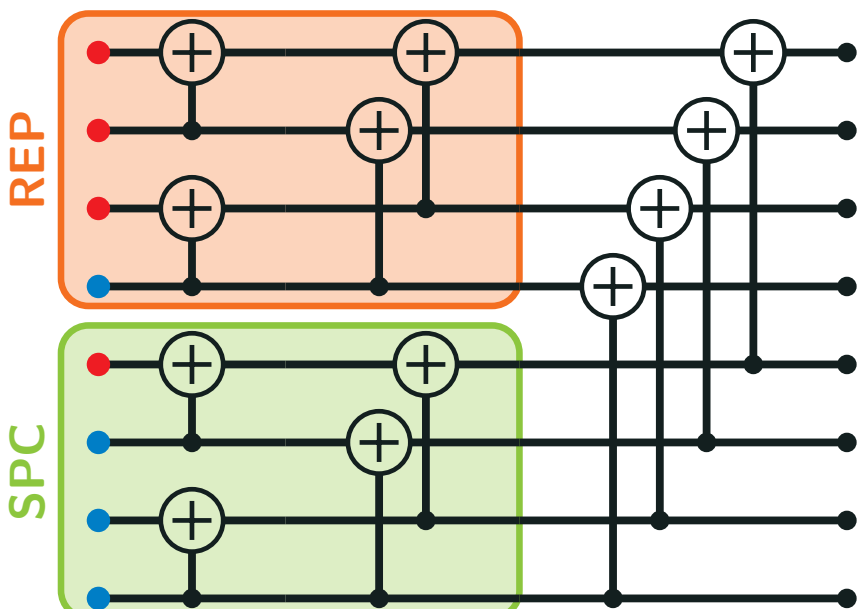


University of Stuttgart
Institute of Telecommunications

Simplified SC Decoding

- Certain combinations of frozen and information bits lead to known subcodes
  - Repetition codes
  - Single parity check codes
  - Rate-0 and rate-1 codes
- These codes can be decoded in one operation
- Simplified SC decoding leverages this to reduce the complexity

➜ This does not degrade decoding performance!

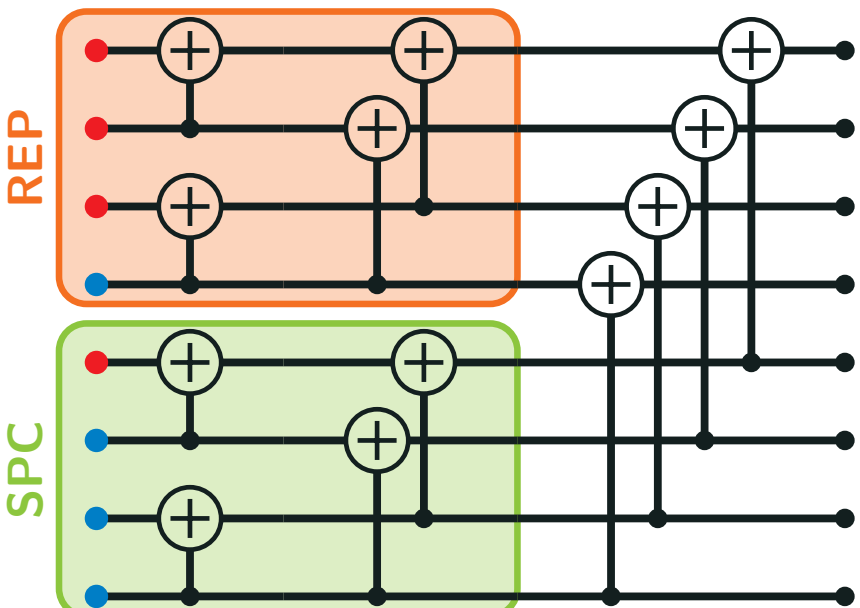


University of Stuttgart
Institute of Telecommunications

Simplified SC Decoding

- Certain combinations of frozen and information bits lead to known subcodes
  - Repetition codes
  - Single parity check codes
  - Rate-0 and rate-1 codes
- These codes can be decoded in one operation
- Simplified SC decoding leverages this to reduce the complexity

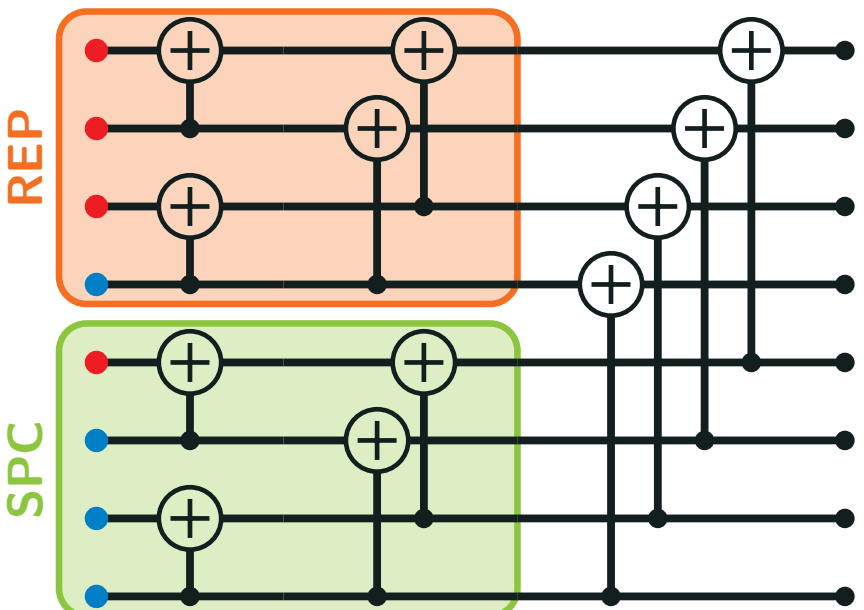
➜ This does not degrade decoding performance!



Simplified SC Decoding

- Certain combinations of frozen and information bits lead to known subcodes
  - Repetition codes
  - Single parity check codes
  - Rate-0 and rate-1 codes
- These codes can be decoded in one operation
- Simplified SC decoding leverages this to reduce the complexity

➜ This does not degrade decoding performance!



Agenda

How to Approximate the Complexity?

- Both polar and LDPC codes need to evaluate the boxplus-function
- For both codes: approximate this function as

$$\boxplus_{x\in\mathcal{X}} x \approx \underbrace{\prod_{x\in\mathcal{X}} \operatorname{sgn}(x)}_{1 \text{ bit op.}} \cdot \underbrace{\alpha \cdot \min_{x\in\mathcal{X}} |x|}_{\geqslant 4 \text{ bit op.}} \tag{1}$$

➜ Complexity dominated by operations of higher bit-width
  - Binary additions of LLRs
  - MIN-operation on a list of LLRs (also an addition)

$$\min(x_1, x_2) = \begin{cases} x_1 & \text{if } x_1 - x_2 \leqslant 0 \\ x_2 & \text{otherwise} \end{cases} \tag{2}$$

How to Approximate the Complexity?

- Both polar and LDPC codes need to evaluate the boxplus-function
- For both codes: approximate this function as

$$\boxplus_{x\in\mathcal{X}} x \approx \underbrace{\prod_{x\in\mathcal{X}} \operatorname{sgn}(x)}_{1 \text{ bit op.}} \cdot \underbrace{\alpha \cdot \min_{x\in\mathcal{X}} |x|}_{\geqslant 4 \text{ bit op.}} \tag{1}$$

➜ Complexity dominated by operations of higher bit-width

- Binary additions of LLRs
- MIN-operation on a list of LLRs (also an addition)

$$\min(x_1, x_2) = \begin{cases} x_1 & \text{if } x_1 - x_2 \leqslant 0 \\ x_2 & \text{otherwise} \end{cases} \tag{2}$$

How to Approximate the Complexity?

- Both polar and LDPC codes need to evaluate the boxplus-function
- For both codes: approximate this function as

$$\boxplus_{x\in\mathcal{X}} x \approx \underbrace{\prod_{x\in\mathcal{X}} \operatorname{sgn}(x)}_{1 \text{ bit op.}} \cdot \underbrace{\alpha \cdot \min_{x\in\mathcal{X}} |x|}_{\geqslant 4 \text{ bit op.}} \tag{1}$$

➜ Complexity dominated by operations of higher bit-width

- Binary additions of LLRs
- MIN-operation on a list of LLRs (also an addition)

$$\min(x_1, x_2) = \begin{cases} x_1 & \text{if } x_1 - x_2 \leqslant 0 \\ x_2 & \text{otherwise} \end{cases} \tag{2}$$

University of Stuttgart
Institute of Telecommunications

How to Approximate the Complexity?

- Both polar and LDPC codes need to evaluate the boxplus-function
- For both codes: approximate this function as

$$\boxplus_{x\in\mathcal{X}} x \approx \underbrace{\prod_{x\in\mathcal{X}} \operatorname{sgn}(x)}_{1 \text{ bit op.}} \cdot \underbrace{\alpha \cdot \min_{x\in\mathcal{X}} |x|}_{\geqslant 4 \text{ bit op.}} \tag{1}$$

➜ Complexity dominated by operations of higher bit-width
  - Binary additions of LLRs
  - MIN-operation on a list of LLRs (also an addition)

$$\min(x_1, x_2) = \begin{cases} x_1 & \text{if } x_1 - x_2 \leqslant 0 \\ x_2 & \text{otherwise} \end{cases} \tag{2}$$

University of Stuttgart
Institute of Telecommunications

How to Approximate the Complexity?

- Both polar and LDPC codes need to evaluate the boxplus-function
- For both codes: approximate this function as

$$\boxplus_{x\in\mathcal{X}} x \approx \underbrace{\prod_{x\in\mathcal{X}} \operatorname{sgn}(x)}_{1 \text{ bit op.}} \cdot \underbrace{\alpha \cdot \min_{x\in\mathcal{X}} |x|}_{\geqslant 4 \text{ bit op.}} \tag{1}$$

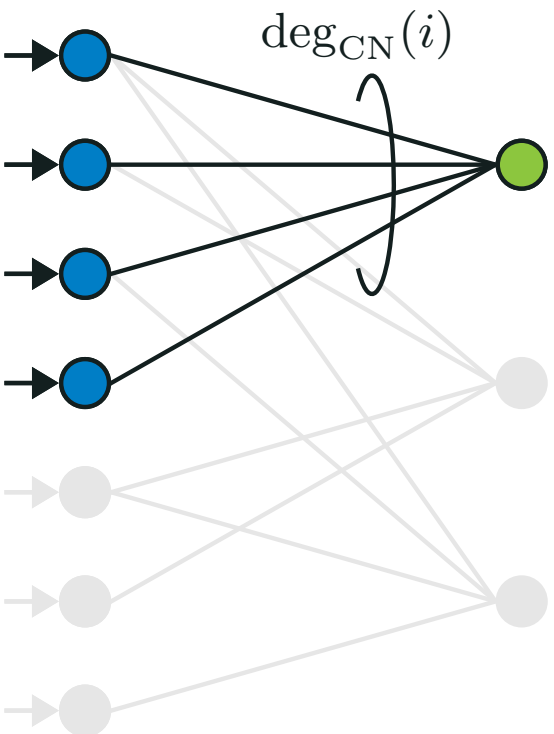
➜ Complexity dominated by operations of higher bit-width
  - Binary additions of LLRs
  - MIN-operation on a list of LLRs (also an addition)

$$\min(x_1, x_2) = \begin{cases} x_1 & \text{if } x_1 - x_2 \leqslant 0 \\ x_2 & \text{otherwise} \end{cases} \tag{2}$$

University of Stuttgart
Institute of Telecommunications

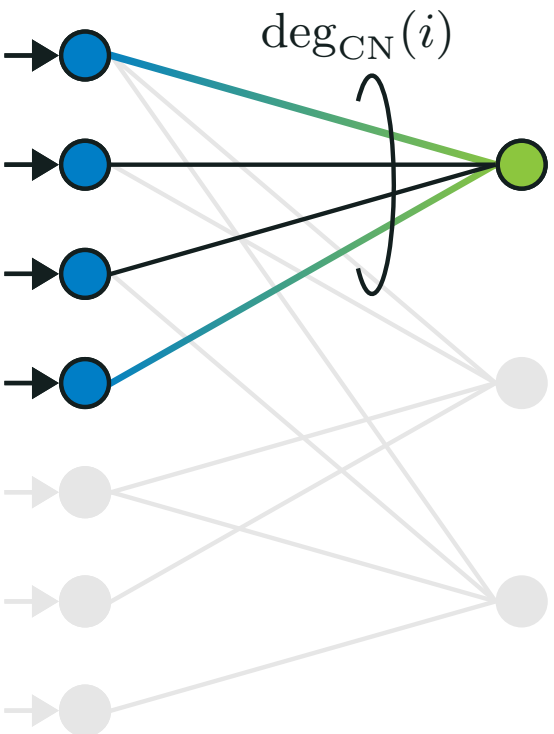
Operations in BP decoding

- For the i-th CN update, we need to compute
  - $2\deg_{\mathrm{CN}}(i) - 3$ comparisons to find the two smallest magnitudes
  - $\deg_{\mathrm{CN}}(i)$ additions to make the belief extrinsic
  - $\deg_{\mathrm{CN}}(i)$ additions to scale with α
- For the j-th VN update, we need to compute
  - $\deg_{\mathrm{VN}}(j)$ additions of incoming beliefs
- In total: $5 \cdot \sum_{i,j} h_{i,j} - 3M$ additions per iteration, depends on code design



Operations in BP decoding

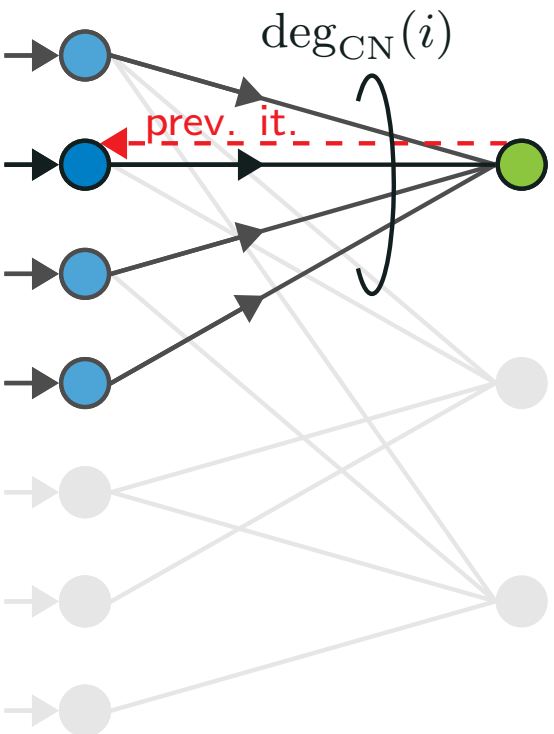
- For the i-th CN update, we need to compute
  - $2\deg_{\mathrm{CN}}(i) - 3$ comparisons to find the two smallest magnitudes
  - $\deg_{\mathrm{CN}}(i)$ additions to make the belief extrinsic
  - $\deg_{\mathrm{CN}}(i)$ additions to scale with α
- For the j-th VN update, we need to compute
  - $\deg_{\mathrm{VN}}(j)$ additions of incoming beliefs
- In total: $5 \cdot \sum_{i,j} h_{i,j} - 3M$ additions per iteration, depends on code design



University of Stuttgart
Institute of Telecommunications

Operations in BP decoding

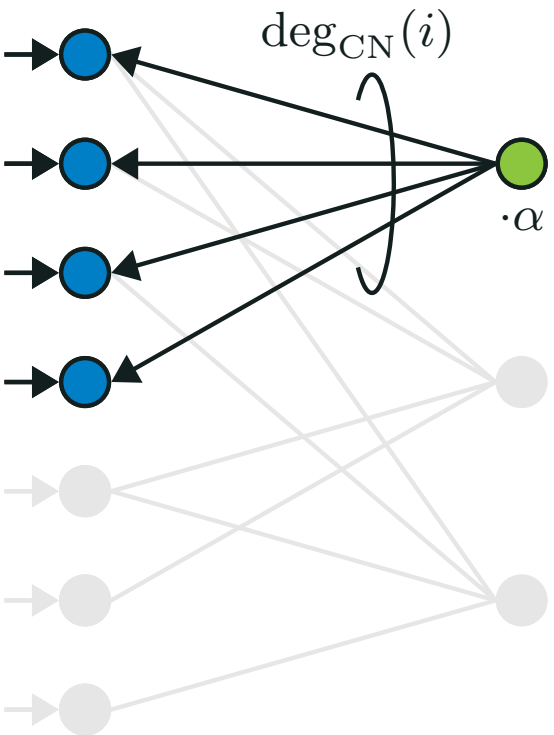
- For the i-th CN update, we need to compute
  - $2\deg_{\mathrm{CN}}(i) - 3$ comparisons to find the two smallest magnitudes
  - $\deg_{\mathrm{CN}}(i)$ additions to make the belief extrinsic
  - $\deg_{\mathrm{CN}}(i)$ additions to scale with α
- For the j-th VN update, we need to compute
  - $\deg_{\mathrm{VN}}(j)$ additions of incoming beliefs
- In total: $5 \cdot \sum_{i,j} h_{i,j} - 3M$ additions per iteration, depends on code design



University of Stuttgart
Institute of Telecommunications

Operations in BP decoding

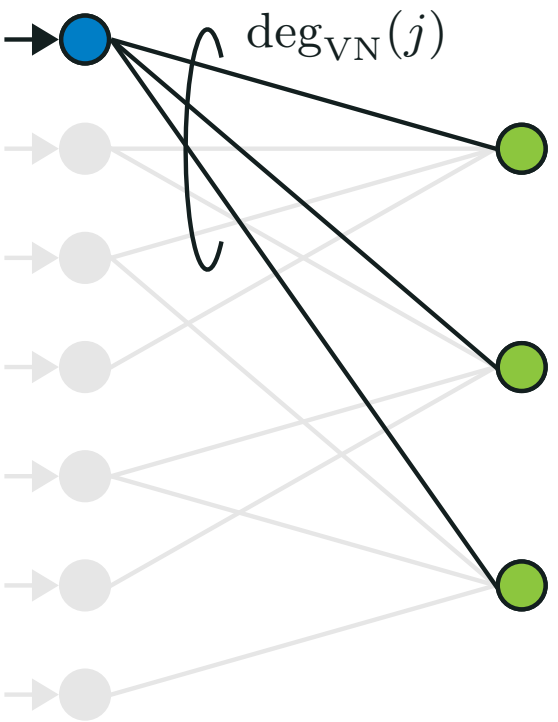
- For the i-th CN update, we need to compute
  - $2\deg_{\mathrm{CN}}(i) - 3$ comparisons to find the two smallest magnitudes
  - $\deg_{\mathrm{CN}}(i)$ additions to make the belief extrinsic
  - $\deg_{\mathrm{CN}}(i)$ additions to scale with α
- For the j-th VN update, we need to compute
  - $\deg_{\mathrm{VN}}(j)$ additions of incoming beliefs
- In total: $5 \cdot \sum_{i,j} h_{i,j} - 3M$ additions per iteration, depends on code design



University of Stuttgart
Institute of Telecommunications

Operations in BP decoding

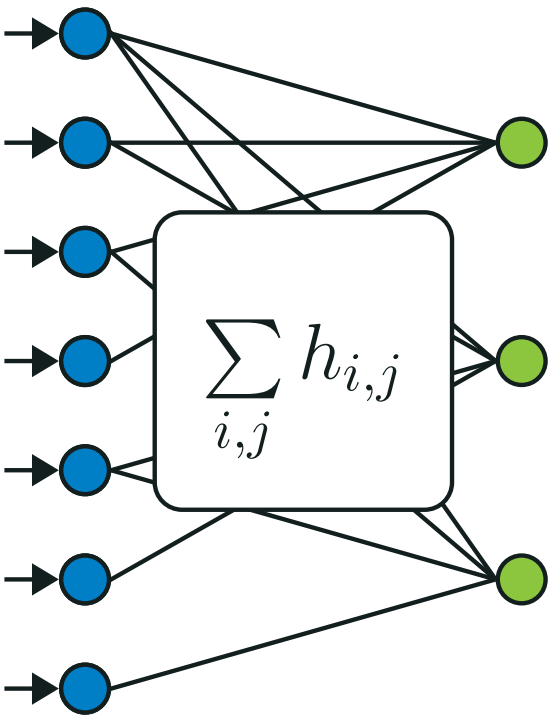
- For the i-th CN update, we need to compute
  - $2\deg_{\mathrm{CN}}(i) - 3$ comparisons to find the two smallest magnitudes
  - $\deg_{\mathrm{CN}}(i)$ additions to make the belief extrinsic
  - $\deg_{\mathrm{CN}}(i)$ additions to scale with α
- For the j-th VN update, we need to compute
  - $\deg_{\mathrm{VN}}(j)$ additions of incoming beliefs
- In total: $5 \cdot \sum_{i,j} h_{i,j} - 3M$ additions per iteration, depends on code design



University of Stuttgart
Institute of Telecommunications

Operations in BP decoding

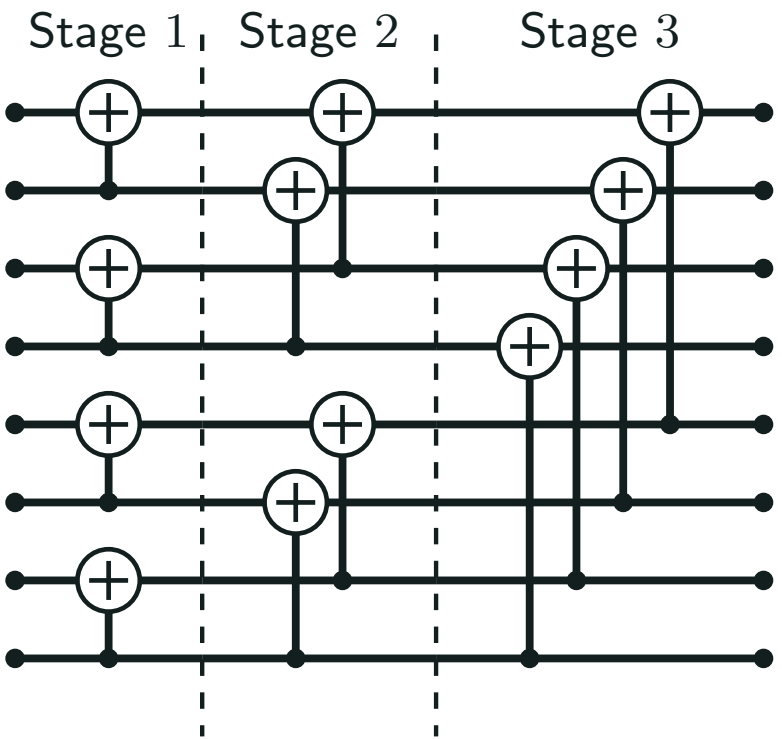
- For the i-th CN update, we need to compute
  - $2\deg_{\mathrm{CN}}(i) - 3$ comparisons to find the two smallest magnitudes
  - $\deg_{\mathrm{CN}}(i)$ additions to make the belief extrinsic
  - $\deg_{\mathrm{CN}}(i)$ additions to scale with α
- For the j-th VN update, we need to compute
  - $\deg_{\mathrm{VN}}(j)$ additions of incoming beliefs
- In total: $5 \cdot \sum_{i,j} h_{i,j} - 3M$ additions per iteration, depends on code design



University of Stuttgart
Institute of Telecommunications

Operations in (S)SC decoding

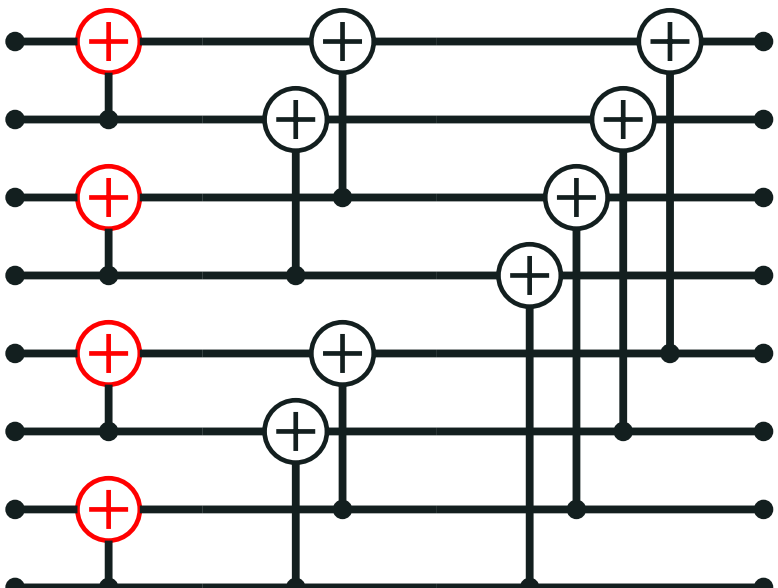
- Plain SC: $\log N$ stages
- Per stage:
  - $N/2$ additions
  - $N/2$ MIN operations
  - ➜ ⊞ does not require scaling with $\alpha \neq 1$
- In total: $N \log N$ LLR operations
- SSC reduces the number of operations
- ➜ Dependent on the code design



University of Stuttgart
Institute of Telecommunications

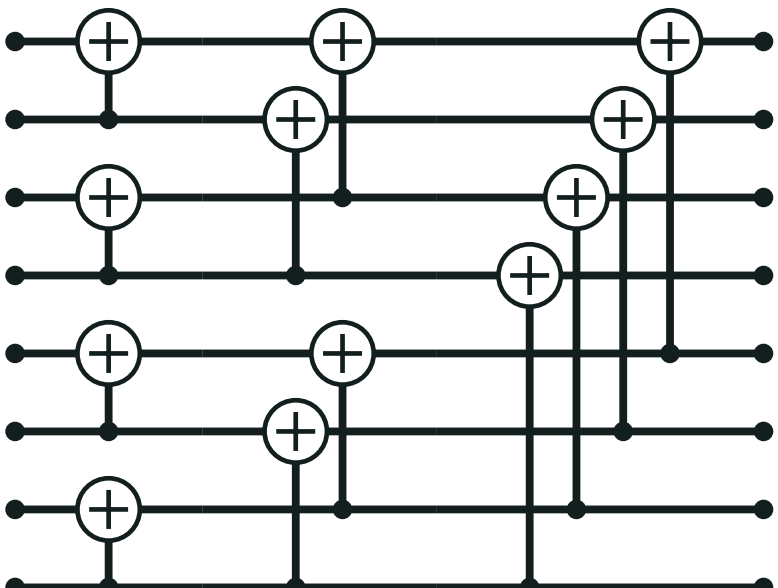
Operations in (S)SC decoding

- Plain SC: $\log N$ stages
- Per stage:
  - $N/2$ additions
  - $N/2$ MIN operations
  - ➜ ⊞ does not require scaling with $\alpha \neq 1$
- In total: $N \log N$ LLR operations
- SSC reduces the number of operations
- ➜ Dependent on the code design



Operations in (S)SC decoding

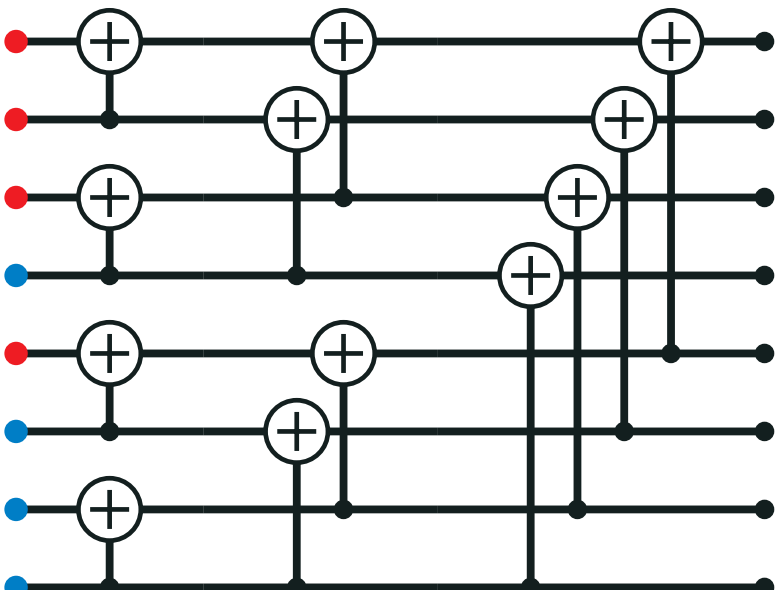
- Plain SC: $\log N$ stages
- Per stage:
  - $N/2$ additions
  - $N/2$ MIN operations
  - ➜ ⊞ does not require scaling with $\alpha \neq 1$
- In total: $N \log N$ LLR operations
- SSC reduces the number of operations
- ➜ Dependent on the code design



Operations in (S)SC decoding

- Plain SC: $\log N$ stages
- Per stage:
  - $N/2$ additions
  - $N/2$ MIN operations
  - ➜ ⊞ does not require scaling with $\alpha \neq 1$
- In total: $N \log N$ LLR operations
- SSC reduces the number of operations

➜ Dependent on the code design



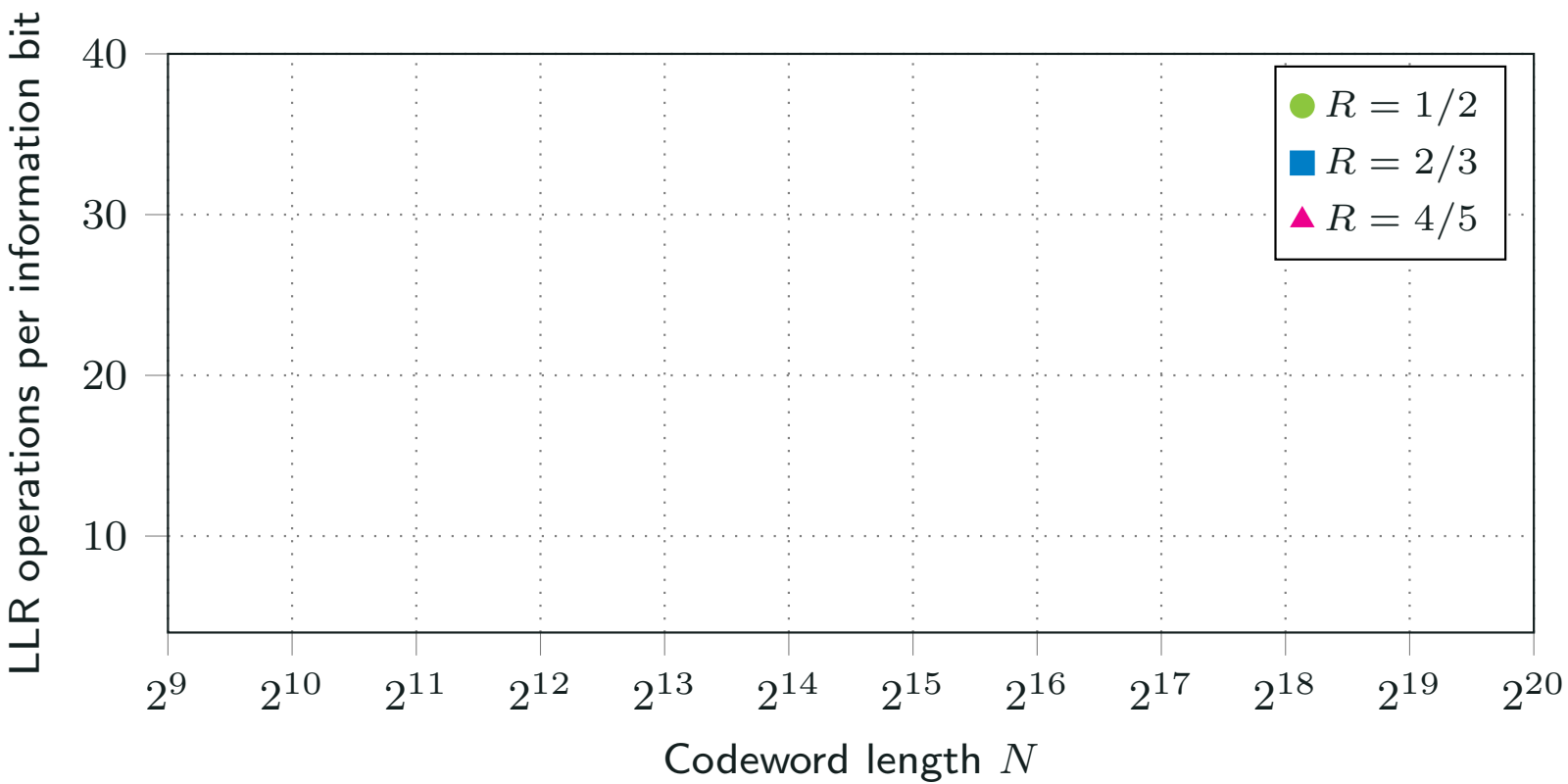
University of Stuttgart
Institute of Telecommunications

Considered Codes

Polar design	Density evolution @ BLER 10^{-6}			
LDPC design	CCSDS			DVB-S2
Dimension K	1 024	4 096	16 384	$N \cdot R$
Length N		K/R		64 800
Code Rate R	$1/2$, $2/3$, $4/5$			

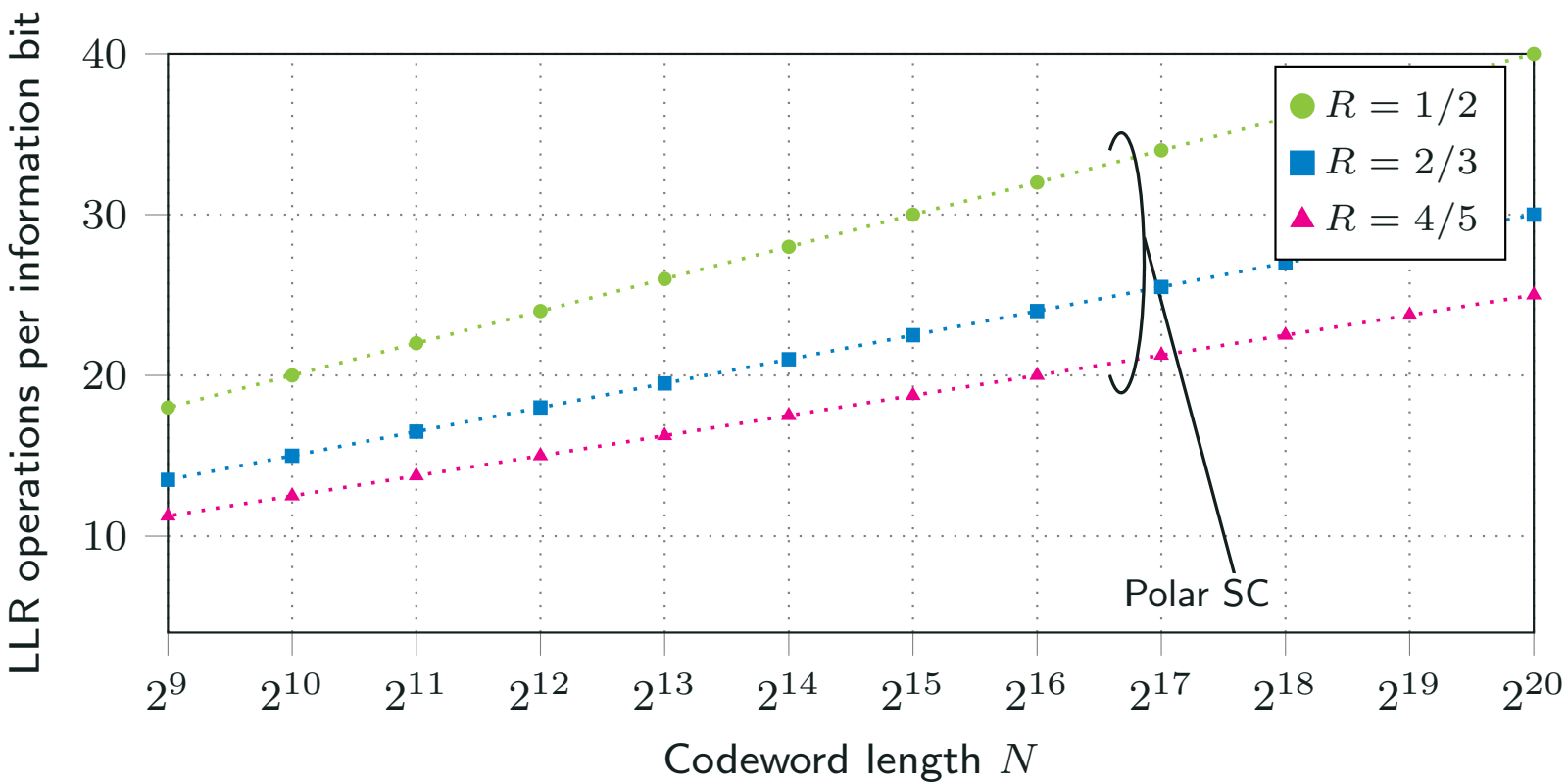
University of Stuttgart
Institute of Telecommunications

Complexity of the Considered Codes

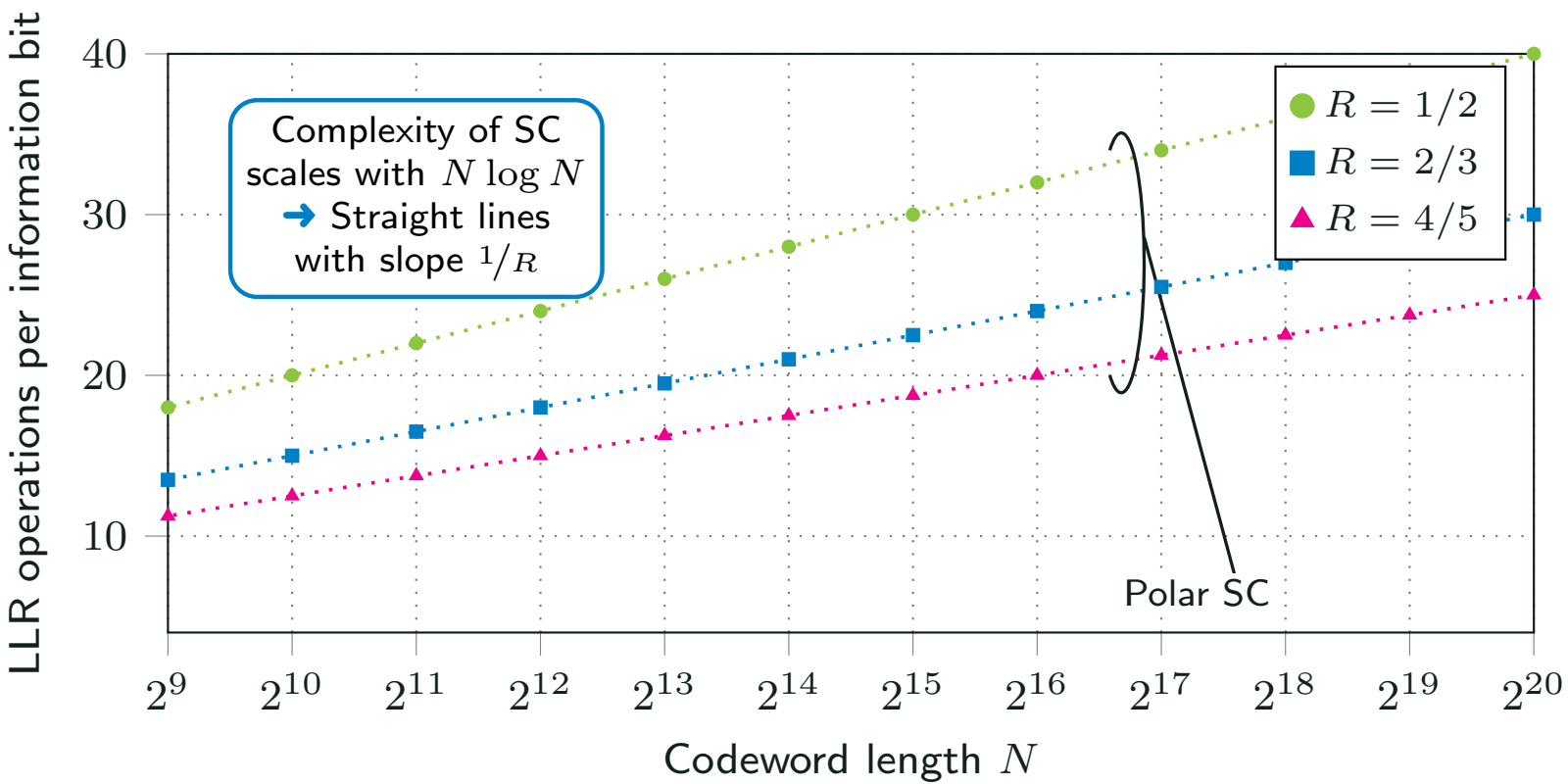


University of Stuttgart
Institute of Telecommunications

Complexity of the Considered Codes

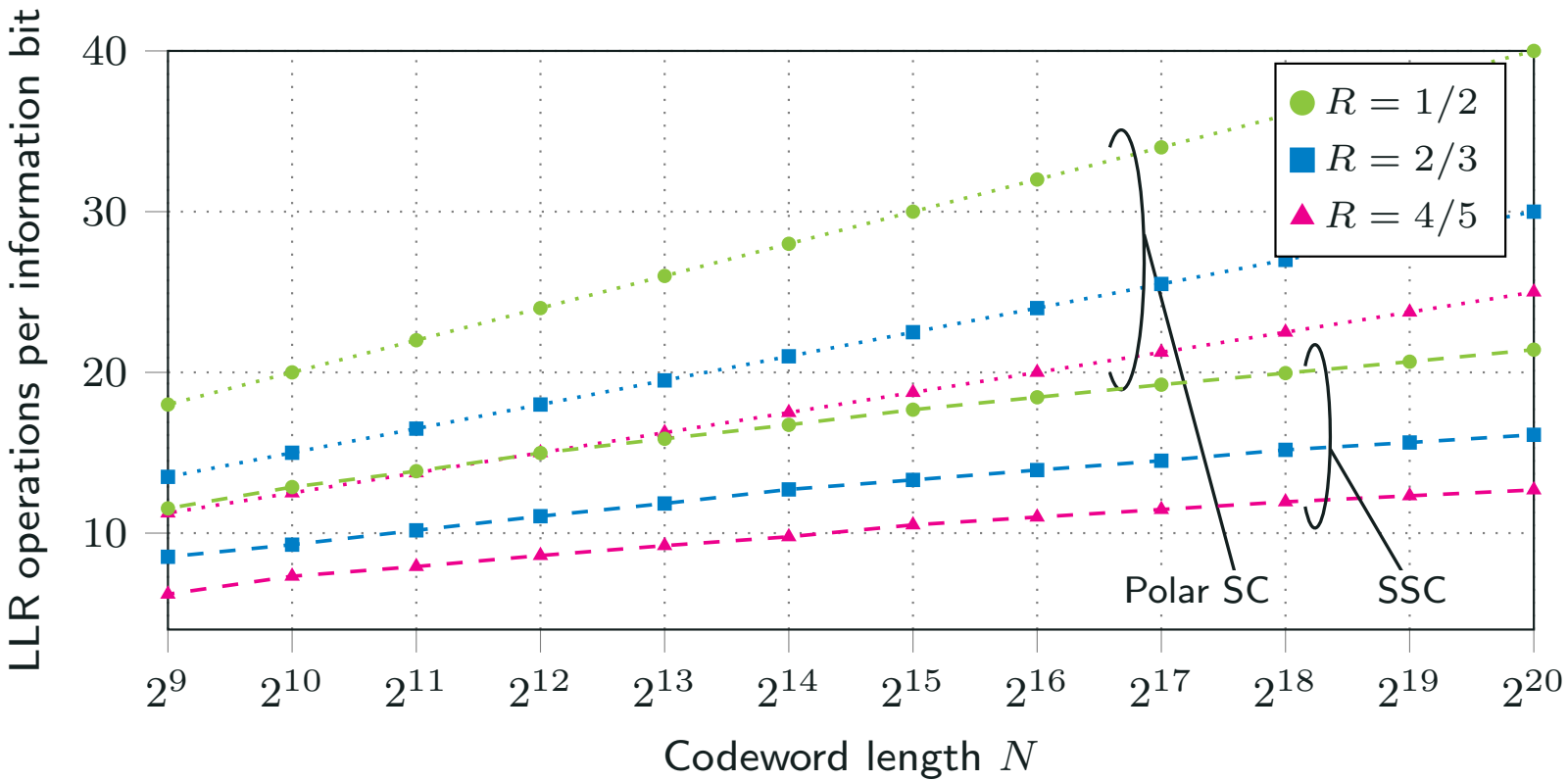


Complexity of the Considered Codes



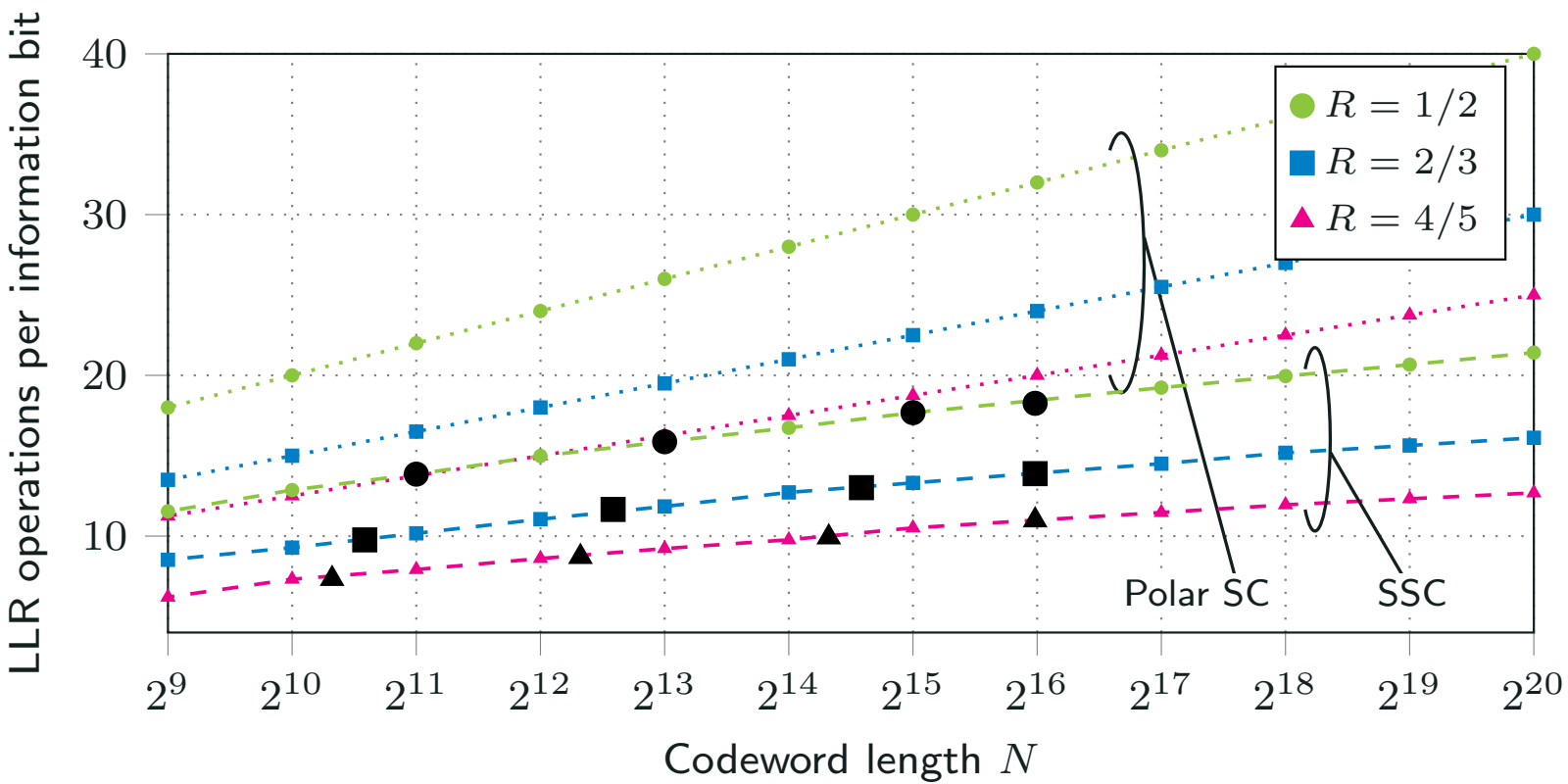
University of Stuttgart
Institute of Telecommunications

Complexity of the Considered Codes



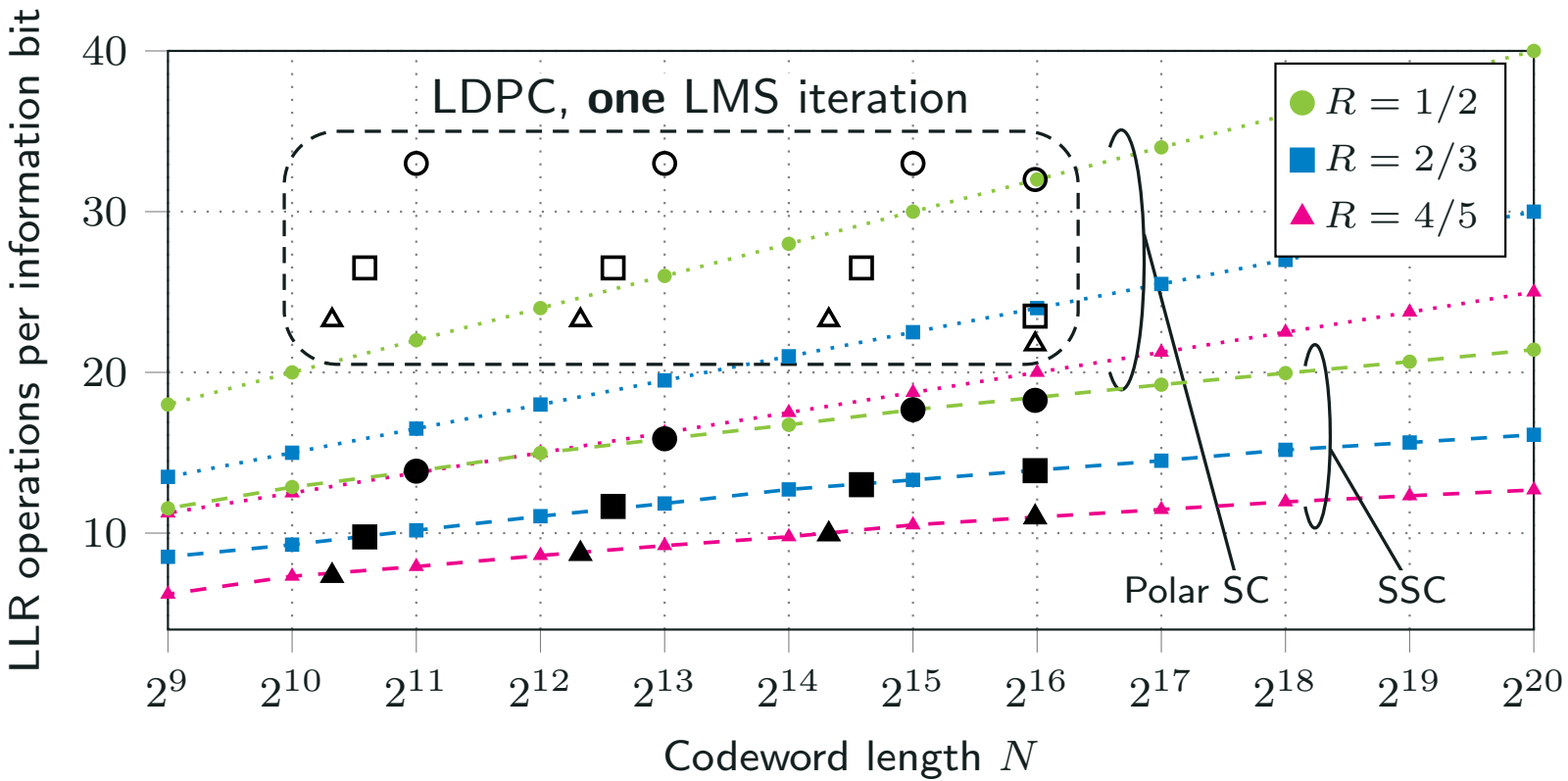
University of Stuttgart
Institute of Telecommunications

Complexity of the Considered Codes



University of Stuttgart
Institute of Telecommunications

Complexity of the Considered Codes



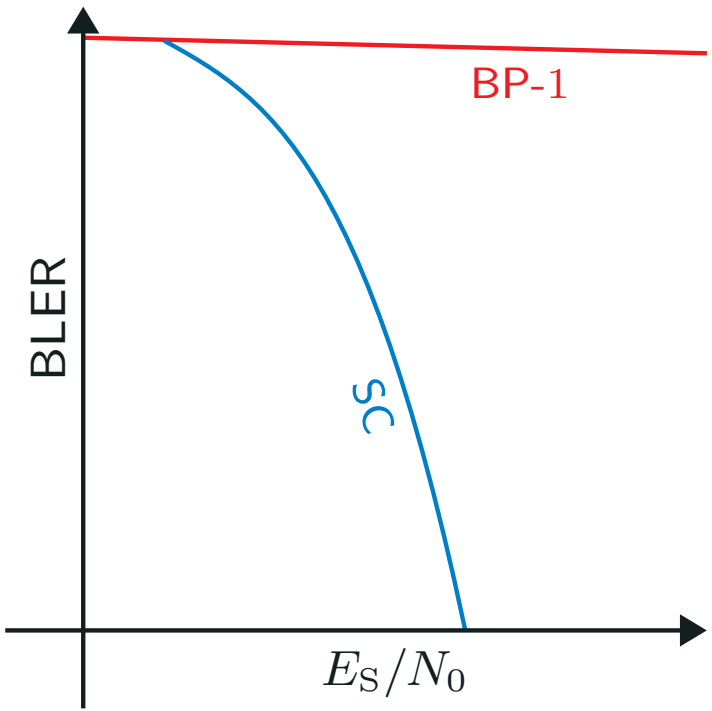
Agenda

University of Stuttgart
Institute of Telecommunications

Fixing the Performance

- BP decoding requires multiple iterations
- Iteration control allows to adjust performance
- We matched the performance by adjusting the max. number of BP iterations

➜ Goal: Similar SNR for BLER of 10^{-4}

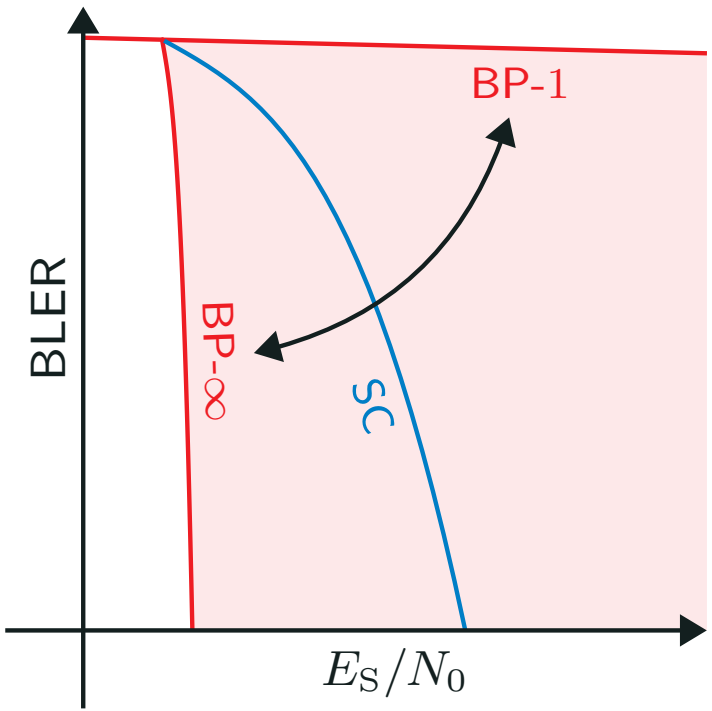


University of Stuttgart
Institute of Telecommunications

Fixing the Performance

- BP decoding requires multiple iterations
- Iteration control allows to adjust performance
- We matched the performance by adjusting the max. number of BP iterations

➜ Goal: Similar SNR for BLER of 10^{-4}

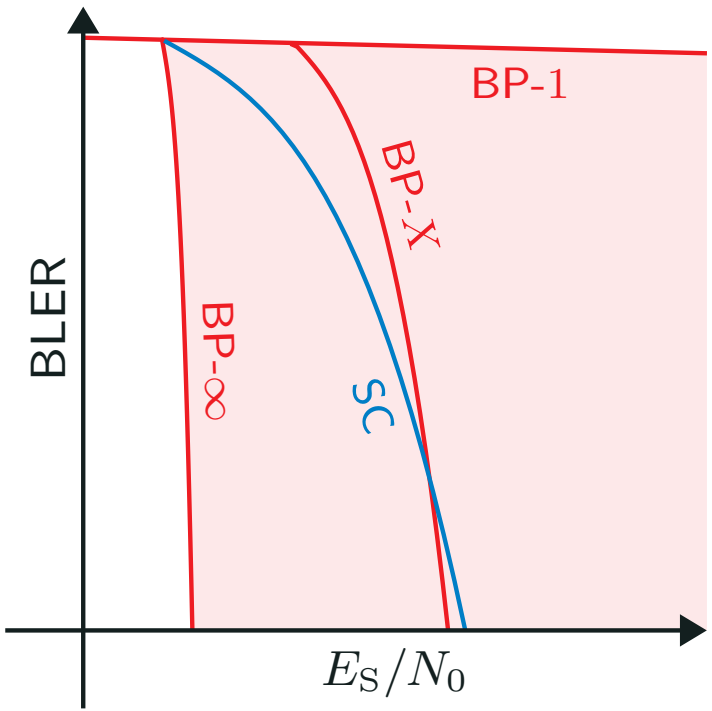


University of Stuttgart
Institute of Telecommunications

Fixing the Performance

- BP decoding requires multiple iterations
- Iteration control allows to adjust performance
- We matched the performance by adjusting the max. number of BP iterations

➜ Goal: Similar SNR for BLER of 10^{-4}

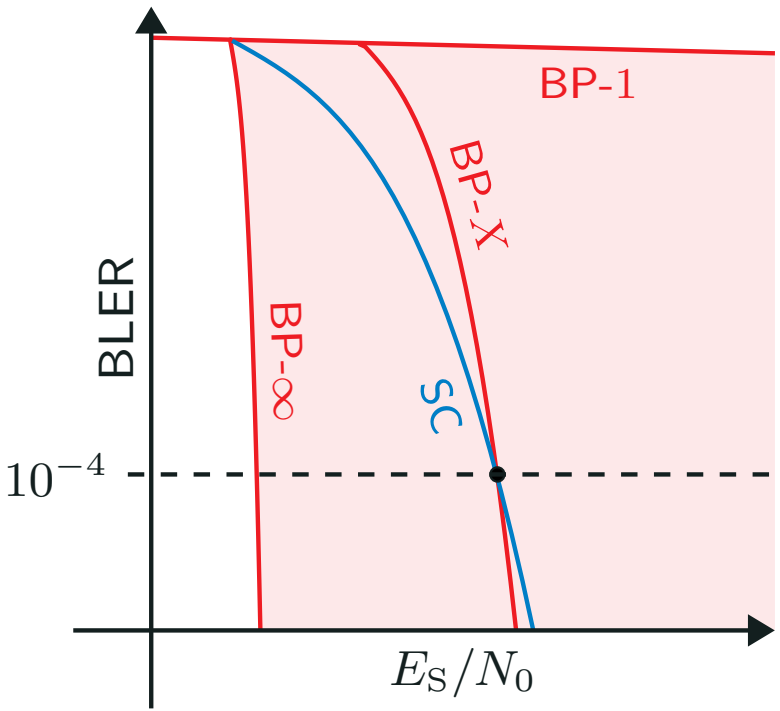


University of Stuttgart
Institute of Telecommunications

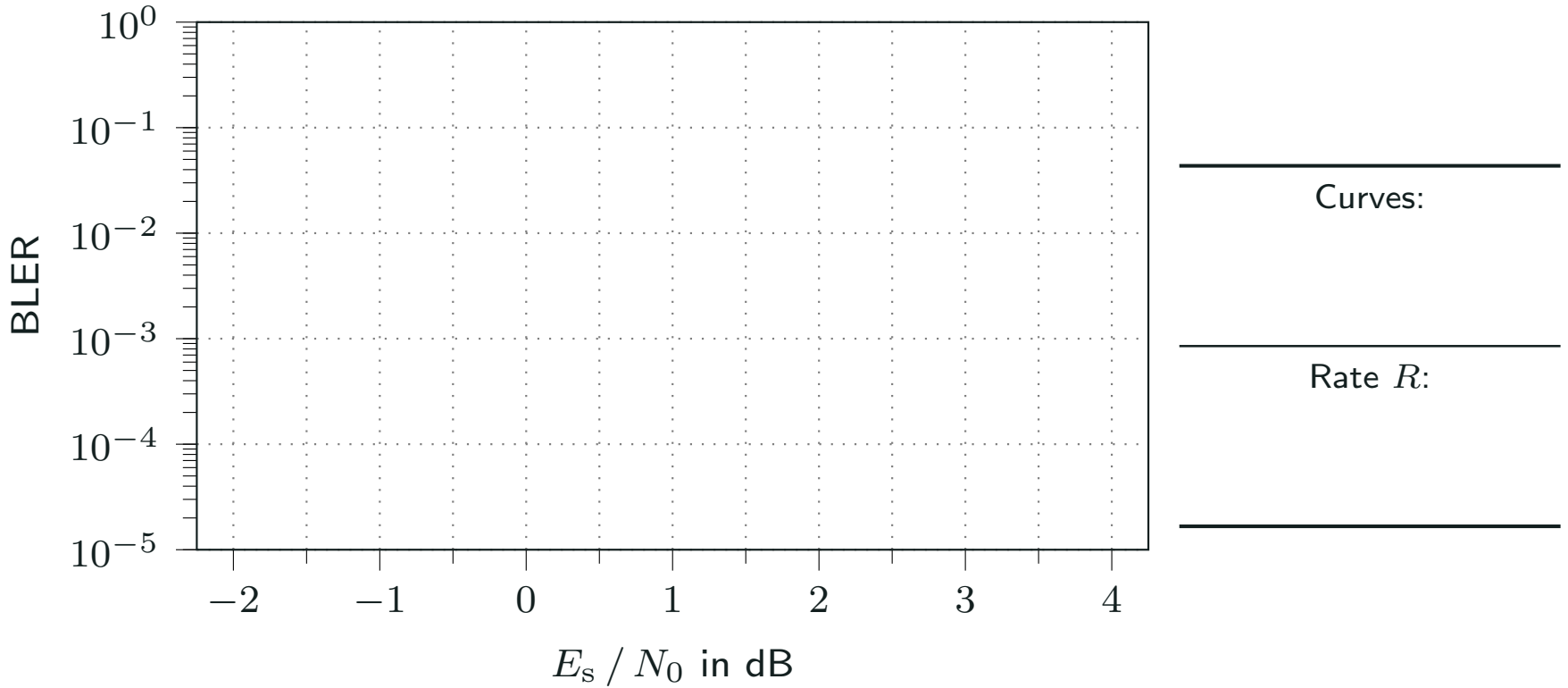
Fixing the Performance

- BP decoding requires multiple iterations
- Iteration control allows to adjust performance
- We matched the performance by adjusting the max. number of BP iterations

➜ Goal: Similar SNR for BLER of 10^{-4}

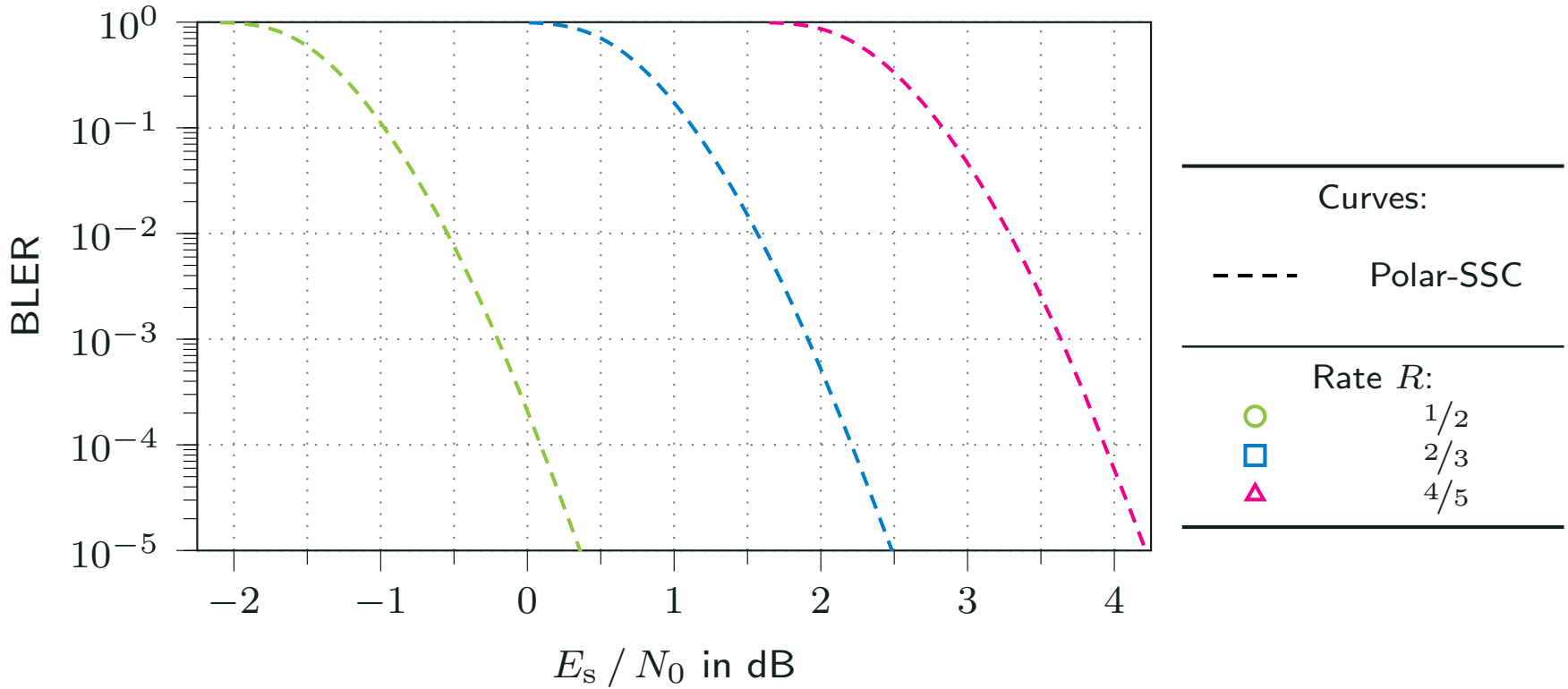


Example 1: $K = 1024$

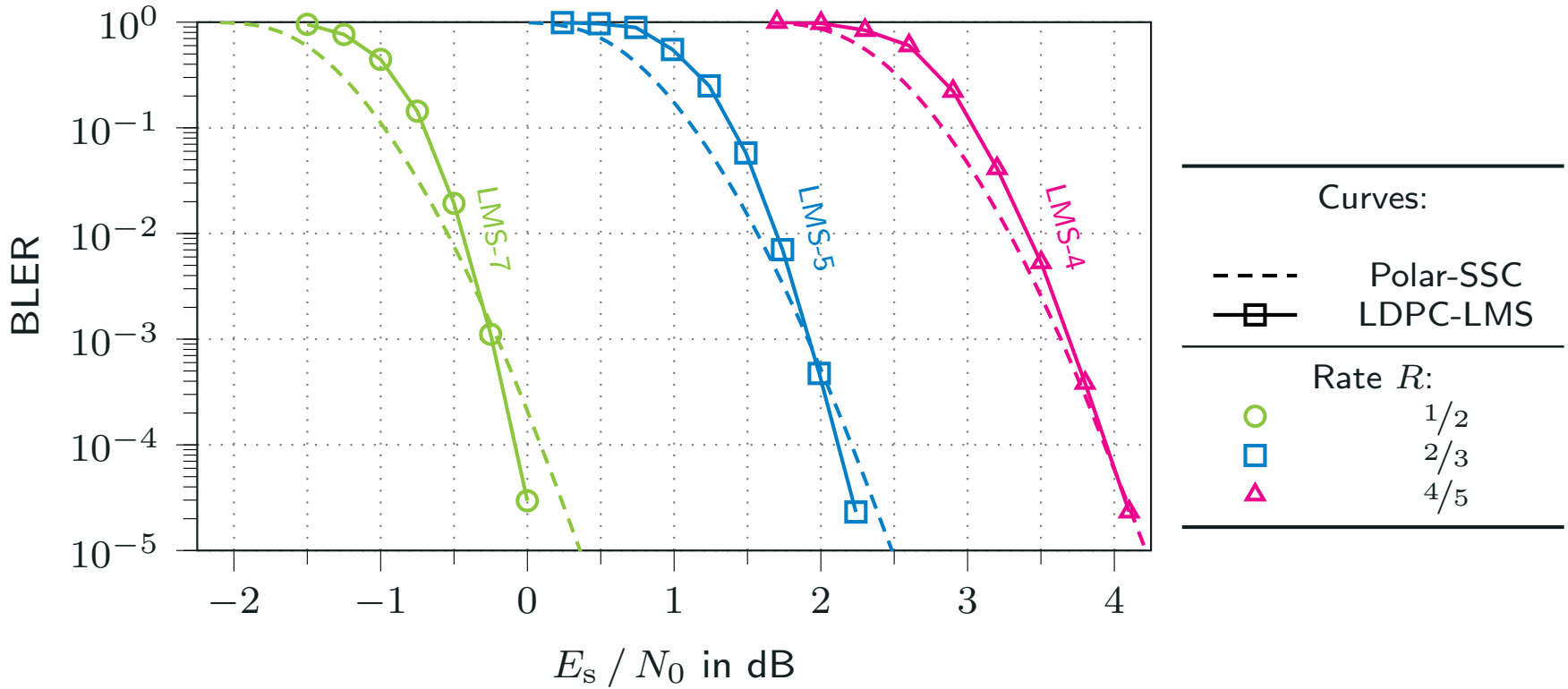


University of Stuttgart
Institute of Telecommunications

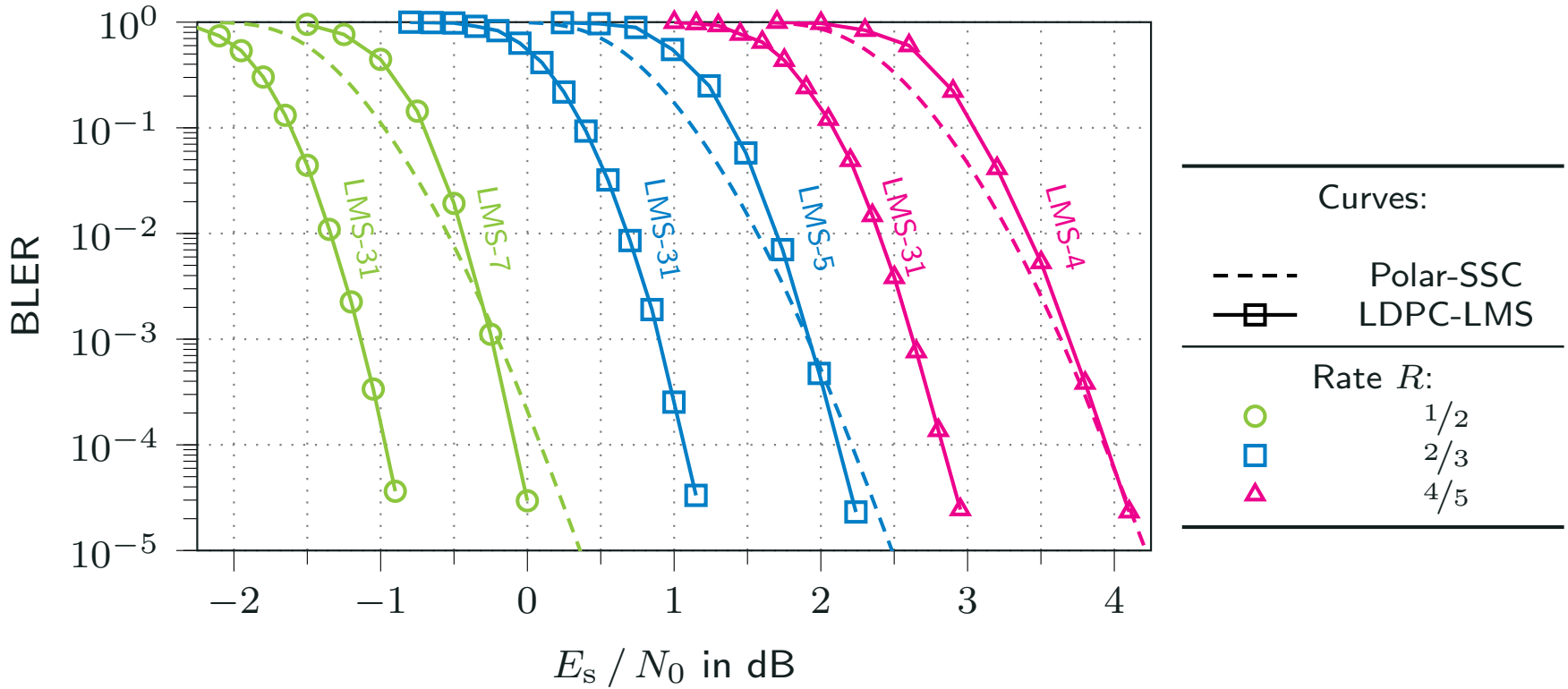
Example 1: $K = 1024$



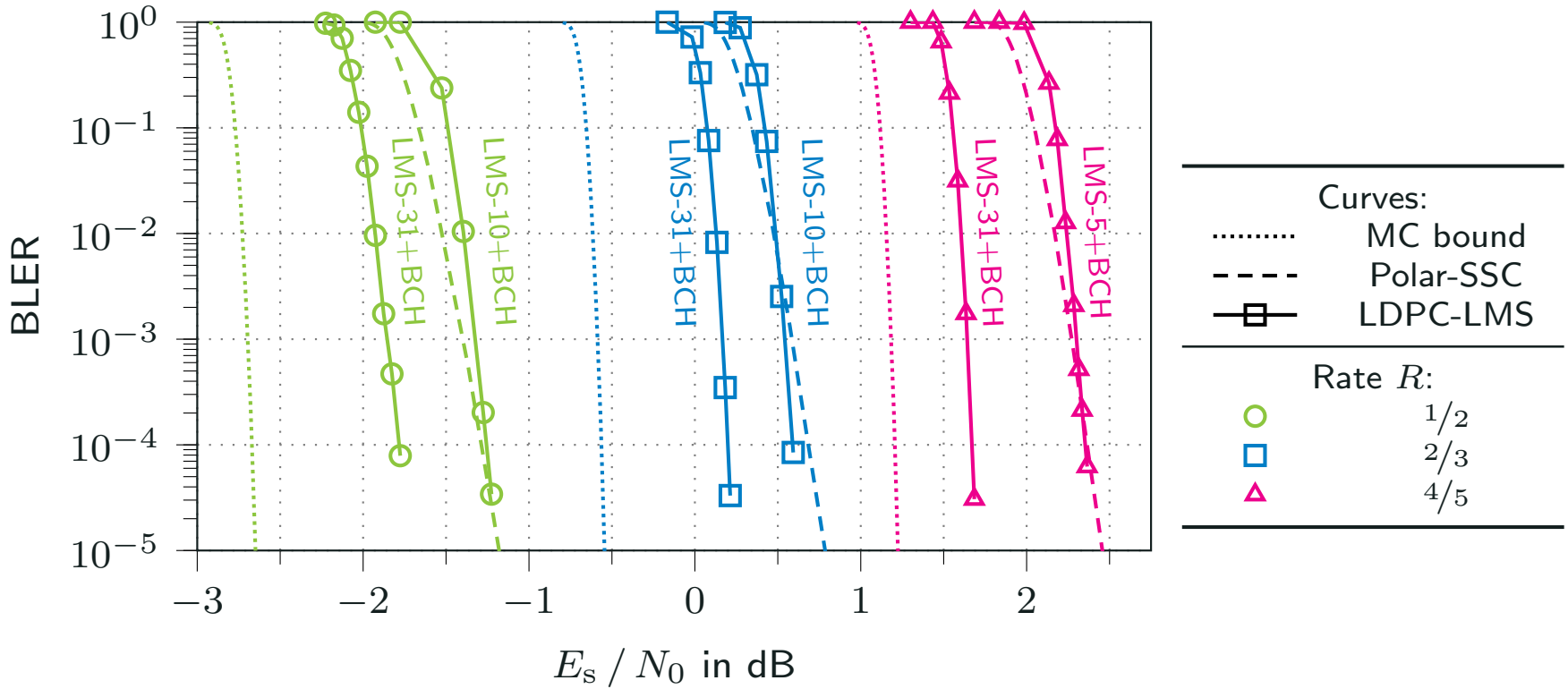
Example 1: $K = 1024$



Example 1: $K = 1024$



Example 2: $N = 64800$



Required Complexity for Matched Performance

	$K = 1024$ (CCSDS)			$N = 64800$ (DVB-S2)		
Code rate R	$1/2$	$2/3$	$4/5$	$1/2$	$2/3$	$4/5$
Polar SSC $N_{\mathrm{Ops,SSC}}/K$	13.85	9.76	7.34	18.26	13.88	10.99
LDPC LMS $N_{\mathrm{Ops,LMS}}/K$	191.40	94.34	75.35	320.00	234.26	108.75
Ratio $N_{\mathrm{Ops,LMS}}/N_{\mathrm{Ops,SSC}}$	13.82	9.67	10.27	17.52	16.88	9.9

Agenda

Conclusion

- We compared the decoding complexity of LDPC and polar codes
- ➜ Hard for LDPC codes to match performance of polar codes at *similar computational complexity*
- At *similar decoding performance* LDPC codes require $\geqslant 17$ times more LLR operations
- Iteration control allows to adjust the decoding performance and complexity
- ➜ If we are willing to invest into more complexity, LDPC codes still reign supreme

Conclusion

- We compared the decoding complexity of LDPC and polar codes
- ➜ Hard for LDPC codes to match performance of polar codes at *similar computational complexity*
- At *similar decoding performance* LDPC codes require $\geqslant 17$ times more LLR operations
- Iteration control allows to adjust the decoding performance and complexity
- ➜ If we are willing to invest into more complexity, LDPC codes still reign supreme

Conclusion

- We compared the decoding complexity of LDPC and polar codes

➜ Hard for LDPC codes to match performance of polar codes at *similar computational complexity*

- At *similar decoding performance* LDPC codes require $\geqslant 17$ times more LLR operations
- Iteration control allows to adjust the decoding performance and complexity

➜ If we are willing to invest into more complexity, LDPC codes still reign supreme

Thank you for your attention!

University of Stuttgart
Institute of Telecommunications

Backup 1: Polar SSC Scales Better Than $N \log N$