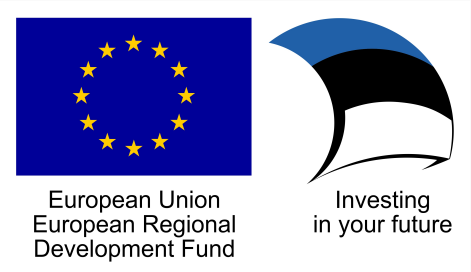
Distance Properties of Short LDPC Codes and Their Impact on the BP, ML and Near-ML Decoding Performance

Irina E. Bocharova[1,2], Boris D. Kudryashov[1], Vitaly Skachek[2], **Yauhen Yakimenka**[2]

[1] St. Petersburg University of Information Technologies, Mechanics and Optics (Russia)
[2] University of Tartu (Estonia)

Acknowledgements

- Norwegian-Estonian Research Cooperation Programme (grant EMP133)
- Estonian Research Council (grant PUT405)
- University of Tartu ASTRA project PER ASPERA Doctoral School of Information and Communication Technologies
- High Performance Computing Centre (University of Tartu)



Outline

1. Code parameters
2. Stopping redundancy hierarchy
3. Considered codes
4. Simulations: FER performance
5. Spectra and bounds
6. But what about BAWGN channel?

Code parameters

Code parameters influence on decoding success (BEC)

Decoding problem on BEC

Solve linear system: $H\,\mathbf{x}^\intercal = \mathbf{0}^\intercal$ for $\mathbf{x} = (x_1, x_2, x_3, x_4, ?, x_6, x_7, ?, x_9, ?)$

[1] BP with extended parity-check matrix

Code parameters influence on decoding success (BEC)

Decoding problem on BEC

Solve linear system: $H\,\mathbf{x}^\intercal = \mathbf{0}^\intercal$ for $\mathbf{x} = (x_1, x_2, x_3, x_4, ?, x_6, x_7, ?, x_9, ?)$

Table 1: Linear $[n, k, d_{\min}]$ code and its parameters

parameter	Decoding algorithm		
	BP (belief propagation)	Near-ML[1]	Maximum-likelihood (ML)
$d_{\min}$, distance spectrum	✓		✓
d_{stop}, stopping spectrum	✓		
d_{dual}		✓	
girth g	✓	✓	
SR hierarchy		✓	

[1] BP with extended parity-check matrix

Stopping redundancy hierarchy

Stopping sets and d_{stop}

$$H = \begin{array}{c} \begin{array}{cccccccccc} x_1 & x_2 & x_3 & x_4 & ? & x_6 & x_7 & ? & x_9 & ? \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 1 & \mathbf{1} & 0 & 1 & \mathbf{1} & 1 & \mathbf{1} \\ 1 & 1 & 0 & 1 & \mathbf{1} & 0 & 0 & \mathbf{0} & 1 & \mathbf{1} \\ 1 & 0 & 1 & 0 & \mathbf{0} & 1 & 0 & \mathbf{1} & 0 & \mathbf{1} \\ 0 & 0 & 1 & 1 & \mathbf{1} & 0 & 1 & \mathbf{1} & 0 & \mathbf{1} \\ 0 & 0 & 1 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 1 & \mathbf{1} \\ 0 & 0 & 0 & 1 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 & \mathbf{0} \\ 1 & 0 & 0 & 1 & \mathbf{0} & 0 & 1 & \mathbf{1} & 0 & \mathbf{1} \end{pmatrix} \end{array}$$

Stopping sets and d_{stop}

$$H = \begin{array}{c} \begin{array}{cccccccccc} x_1 & x_2 & x_3 & x_4 & ? & x_6 & x_7 & ? & x_9 & ? \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 1 & \mathbf{1} & 0 & 1 & \mathbf{1} & 1 & \mathbf{1} \\ 1 & 1 & 0 & 1 & \mathbf{1} & 0 & 0 & \mathbf{0} & 1 & \mathbf{1} \\ 1 & 0 & 1 & 0 & \mathbf{0} & 1 & 0 & \mathbf{1} & 0 & \mathbf{1} \\ 0 & 0 & 1 & 1 & \mathbf{1} & 0 & 1 & \mathbf{1} & 0 & \mathbf{1} \\ 0 & 0 & 1 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 1 & \mathbf{1} \\ 0 & 0 & 0 & 1 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 & \mathbf{0} \\ 1 & 0 & 0 & 1 & \mathbf{0} & 0 & 1 & \mathbf{1} & 0 & \mathbf{1} \end{pmatrix} \end{array}$$

Stopping distance, d_{stop}

Size of the smallest stopping set.

Stopping redundancy hierarchy (by example)

Aim

By adding redundant rows, remove small stopping sets (up to size ℓ)

Stopping redundancy hierarchy (by example)

Aim

By adding redundant rows, remove small stopping sets (up to size ℓ)

$$
H = \begin{array}{c}
\\
c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7
\end{array}
\begin{array}{c}
\begin{array}{cccccccccc} x_1 & x_2 & x_3 & x_4 & ? & x_6 & x_7 & ? & x_9 & ? \end{array} \\
\left(\begin{array}{cccccccccc}
1 & 1 & 0 & 1 & \mathbf{1} & 0 & 1 & \mathbf{1} & 1 & \mathbf{1} \\
1 & 1 & 0 & 1 & \mathbf{1} & 0 & 0 & \mathbf{0} & 1 & \mathbf{1} \\
1 & 0 & 1 & 0 & \mathbf{0} & 1 & 0 & \mathbf{1} & 0 & \mathbf{1} \\
0 & 0 & 1 & 1 & \mathbf{1} & 0 & 1 & \mathbf{1} & 0 & \mathbf{1} \\
0 & 0 & 1 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 1 & \mathbf{1} \\
0 & 0 & 0 & 1 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 & \mathbf{0} \\
1 & 0 & 0 & 1 & \mathbf{0} & 0 & 1 & \mathbf{1} & 0 & \mathbf{1}
\end{array}\right)
\end{array}
$$

Stopping redundancy hierarchy (by example)

Aim

By adding redundant rows, remove small stopping sets (up to size ℓ)

$$H' = \begin{array}{c} \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \\ \hline c_1+c_2+c_3 \\ c_2+c_3 \end{array} \begin{array}{c} \begin{array}{cccccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} \end{array} \\ \left(\begin{array}{cccccccccc} 1 & 1 & 0 & 1 & \mathbf{1} & 0 & 1 & \mathbf{1} & 1 & \mathbf{1} \\ 1 & 1 & 0 & 1 & \mathbf{1} & 0 & 0 & \mathbf{0} & 1 & \mathbf{1} \\ 1 & 0 & 1 & 0 & \mathbf{0} & 1 & 0 & \mathbf{1} & 0 & \mathbf{1} \\ 0 & 0 & 1 & 1 & \mathbf{1} & 0 & 1 & \mathbf{1} & 0 & \mathbf{1} \\ 0 & 0 & 1 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 1 & \mathbf{1} \\ 0 & 0 & 0 & 1 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 & \mathbf{0} \\ 1 & 0 & 0 & 1 & \mathbf{0} & 0 & 1 & \mathbf{1} & 0 & \mathbf{1} \\ \hline 1 & 0 & 1 & 0 & \mathbf{0} & 1 & 1 & \mathbf{0} & 0 & \mathbf{1} \\ 0 & 1 & 1 & 1 & \mathbf{1} & 1 & 0 & \mathbf{1} & 1 & \mathbf{0} \end{array}\right) \end{array}$$

Stopping redundancy hierarchy (by example)

Aim

By adding redundant rows, remove small stopping sets (up to size ℓ)

$$H' = \begin{array}{c} \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \\ \hline c_1+c_2+c_3 \\ c_2+c_3 \end{array} \begin{array}{c} \begin{array}{cccccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} \end{array} \\ \left(\begin{array}{cccccccccc} 1 & 1 & 0 & 1 & \mathbf{1} & 0 & 1 & \mathbf{1} & 1 & \mathbf{1} \\ 1 & 1 & 0 & 1 & \mathbf{1} & 0 & 0 & \mathbf{0} & 1 & \mathbf{1} \\ 1 & 0 & 1 & 0 & \mathbf{0} & 1 & 0 & \mathbf{1} & 0 & \mathbf{1} \\ 0 & 0 & 1 & 1 & \mathbf{1} & 0 & 1 & \mathbf{1} & 0 & \mathbf{1} \\ 0 & 0 & 1 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 1 & \mathbf{1} \\ 0 & 0 & 0 & 1 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 & \mathbf{0} \\ 1 & 0 & 0 & 1 & \mathbf{0} & 0 & 1 & \mathbf{1} & 0 & \mathbf{1} \\ \hline 1 & 0 & 1 & 0 & \mathbf{0} & 1 & 1 & \mathbf{0} & 0 & \mathbf{1} \\ 0 & 1 & 1 & 1 & \mathbf{1} & 1 & 0 & \mathbf{1} & 1 & \mathbf{0} \end{array}\right) \end{array}$$

ℓ-th stopping redundancy

Minimum number of rows ρ_ℓ, s.t. there are no stopping sets of size up to ℓ (except codewords).

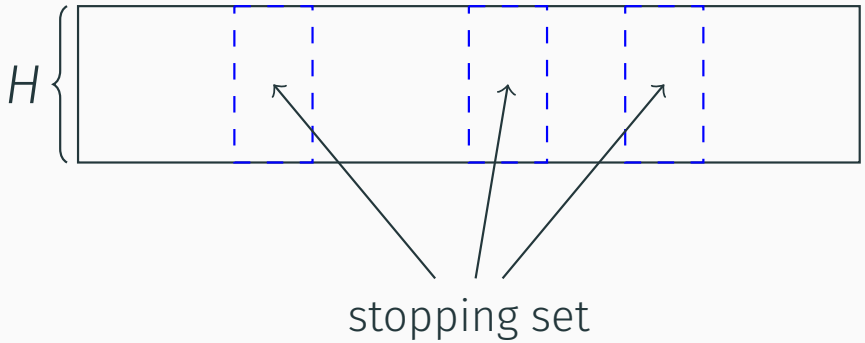
Achieving ML performance

Corrollary from definition

It's possible to build $\rho_{n-k} \times n$ extended parity-check matrix H', s.t. BP decoder **achieves ML performance**.

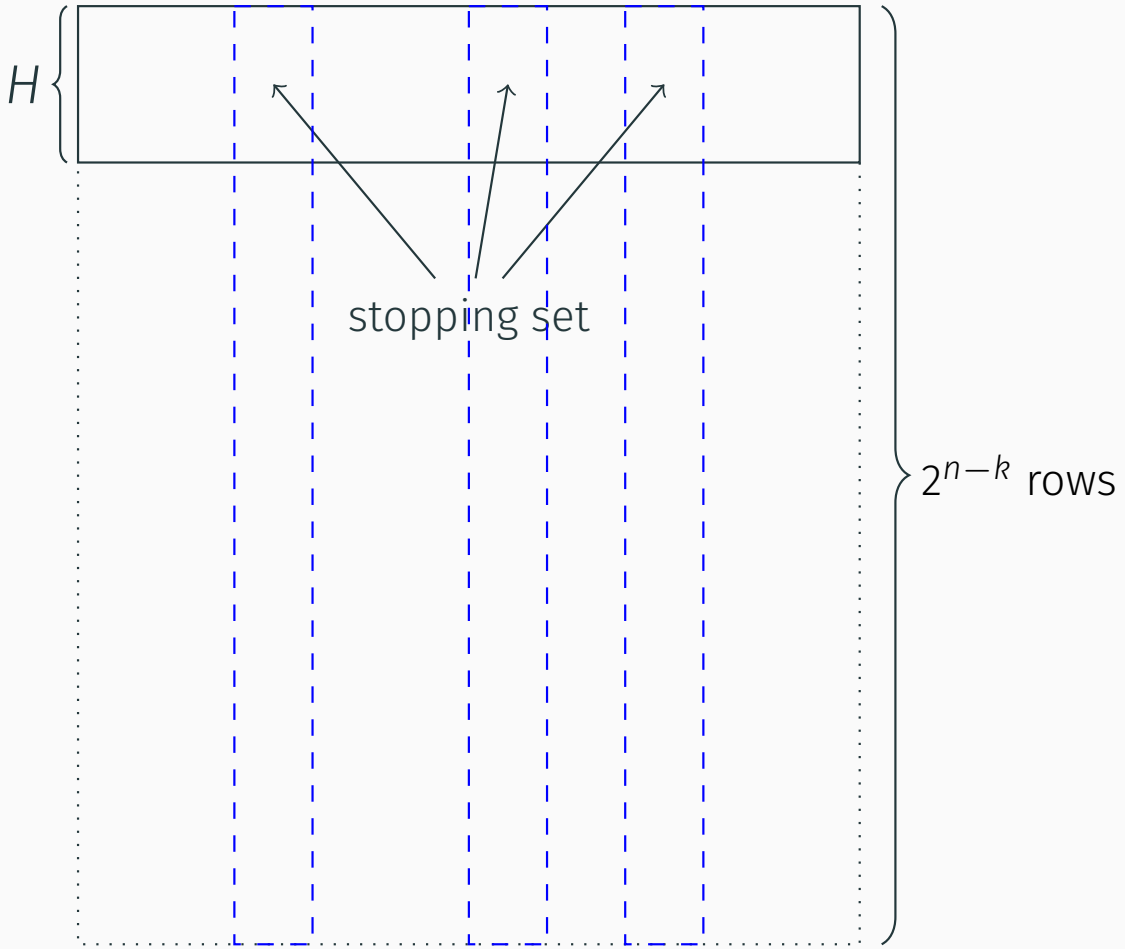
Stopping redundancy – upper bound (i)

Intuition/main observation



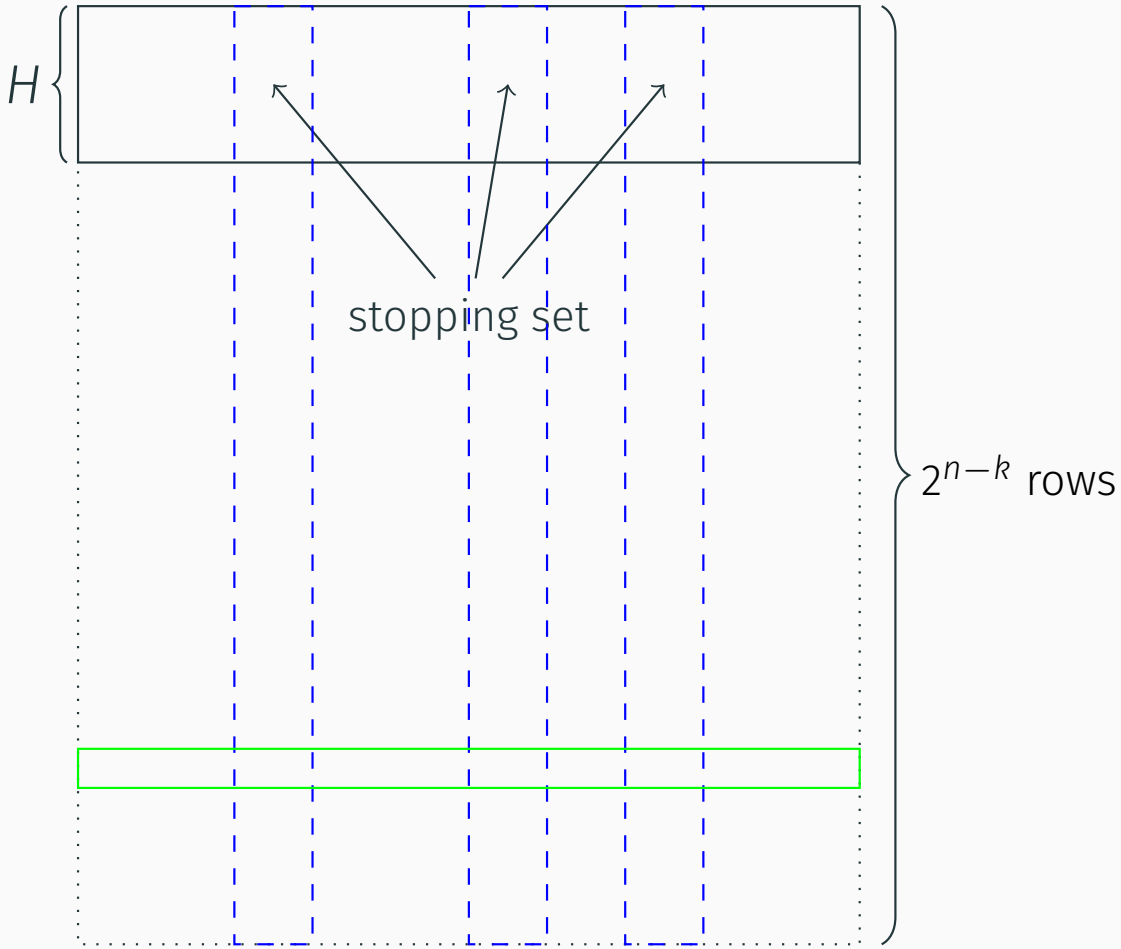
Stopping redundancy – upper bound (i)

Intuition/main observation



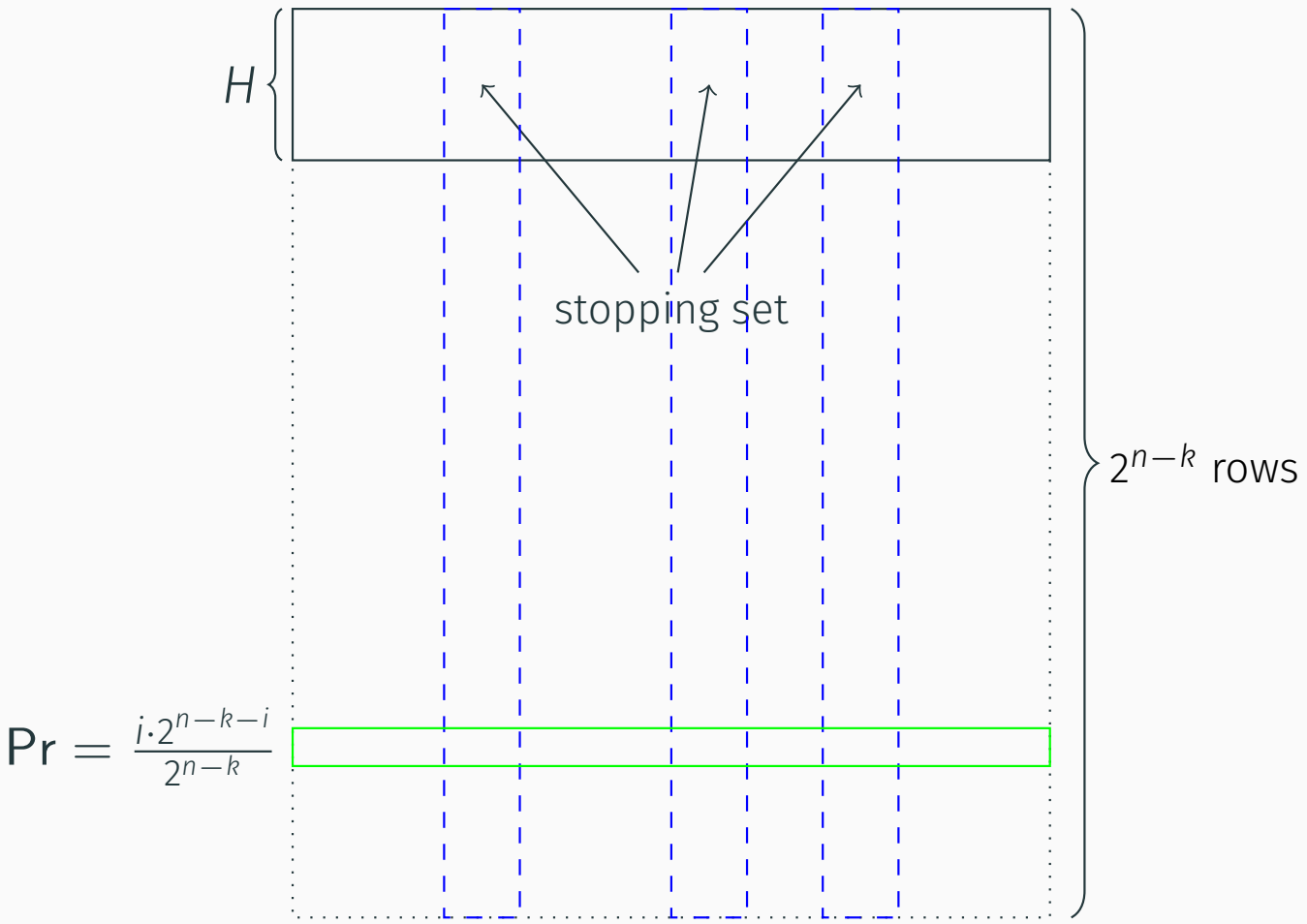
Stopping redundancy – upper bound (i)

Intuition/main observation



Stopping redundancy – upper bound (i)

Intuition/main observation

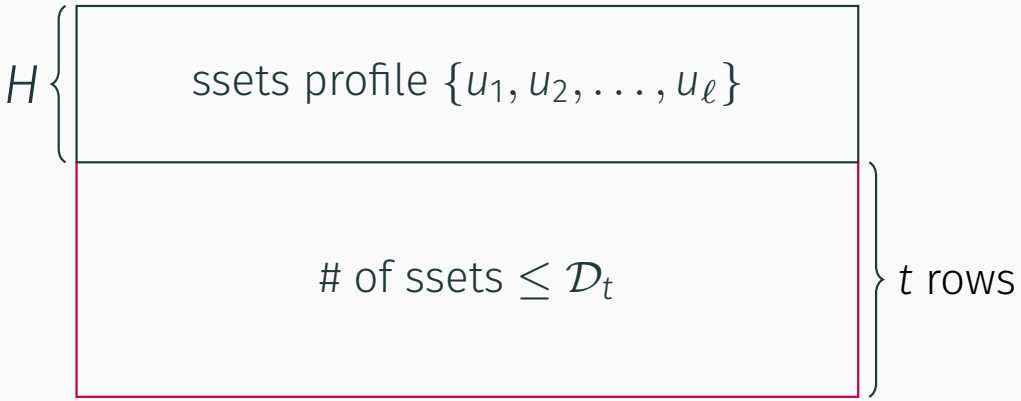


Stopping redundancy – upper bound (ii)

Upper bound construction

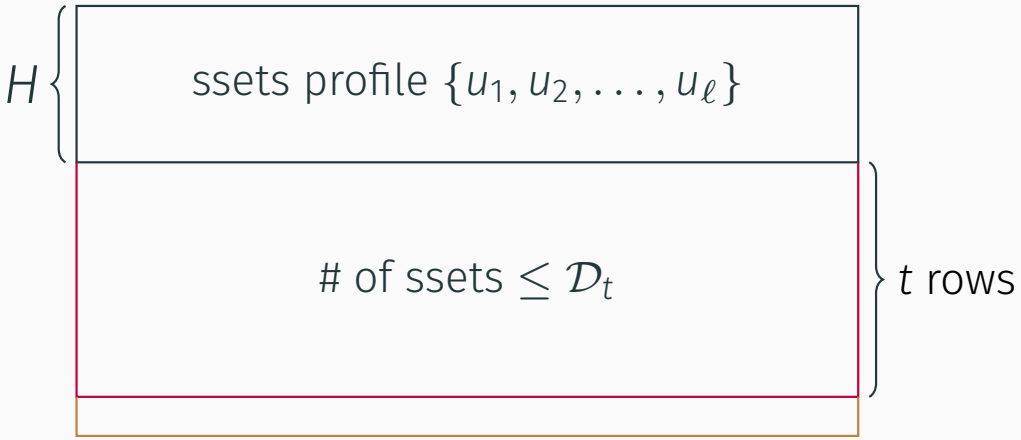
Stopping redundancy – upper bound (ii)

Upper bound construction



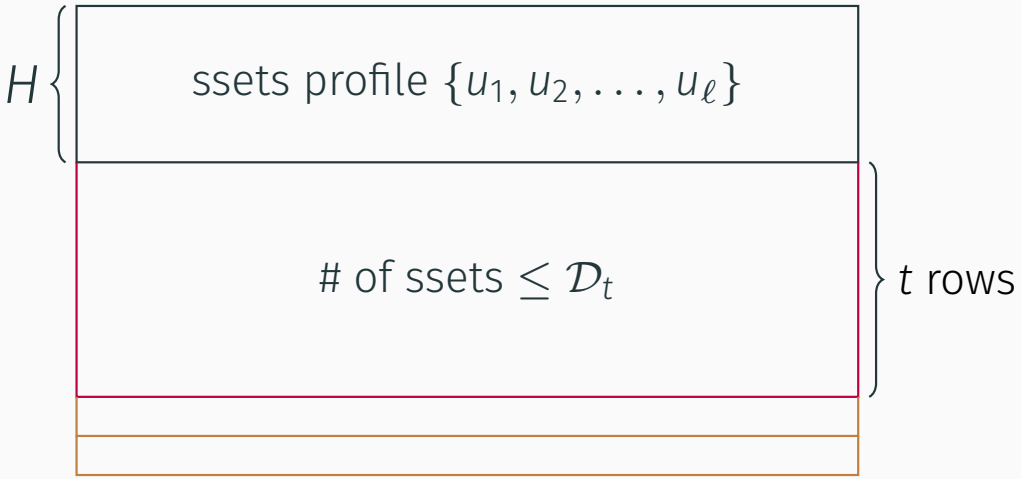
Stopping redundancy – upper bound (ii)

Upper bound construction



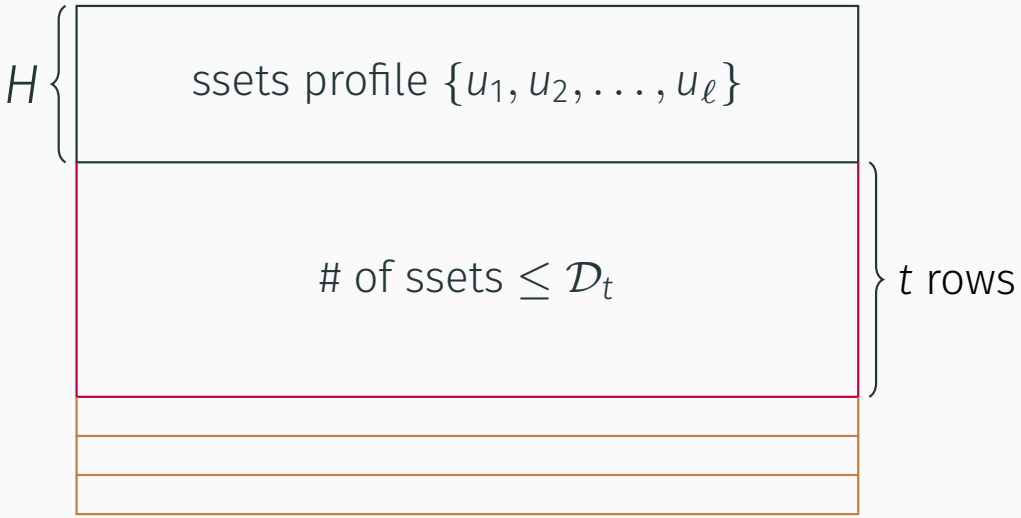
Stopping redundancy – upper bound (ii)

Upper bound construction



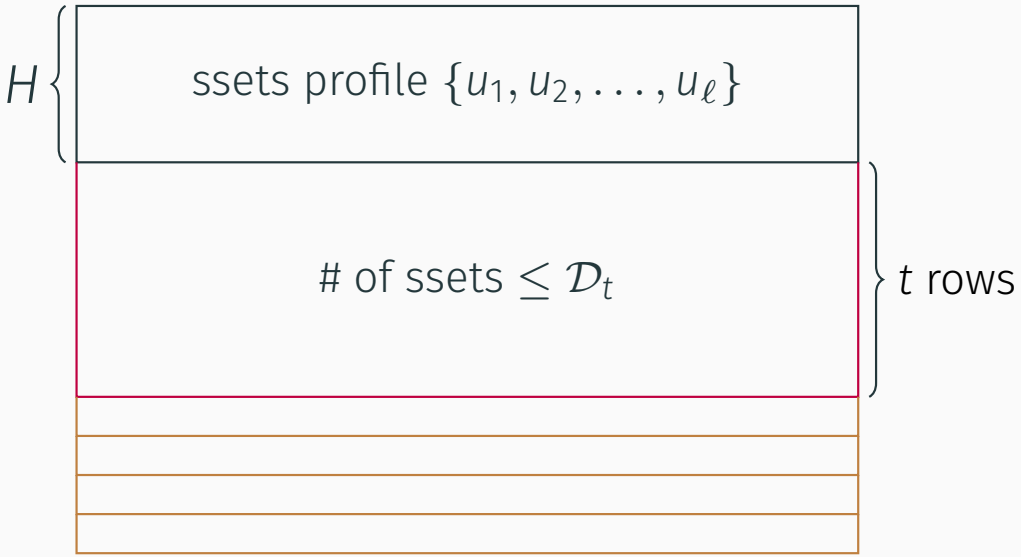
Stopping redundancy – upper bound (ii)

Upper bound construction



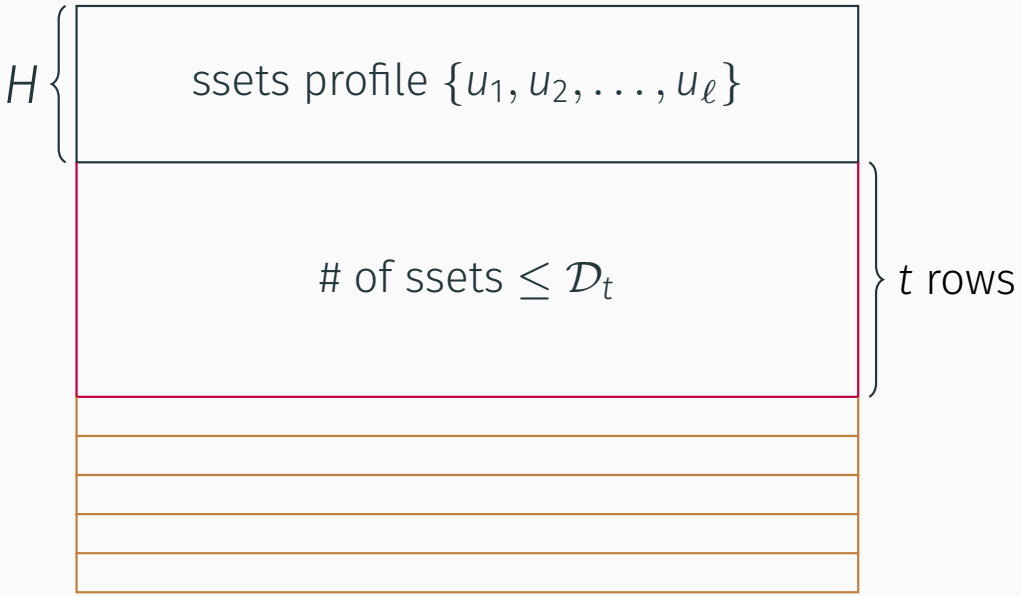
Stopping redundancy – upper bound (ii)

Upper bound construction



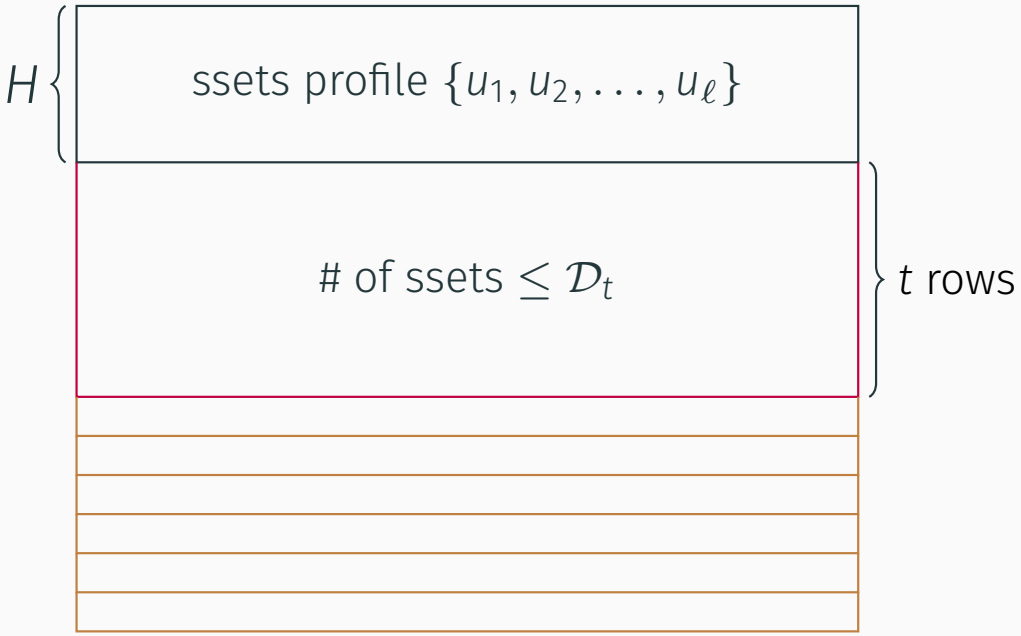
Stopping redundancy – upper bound (ii)

Upper bound construction



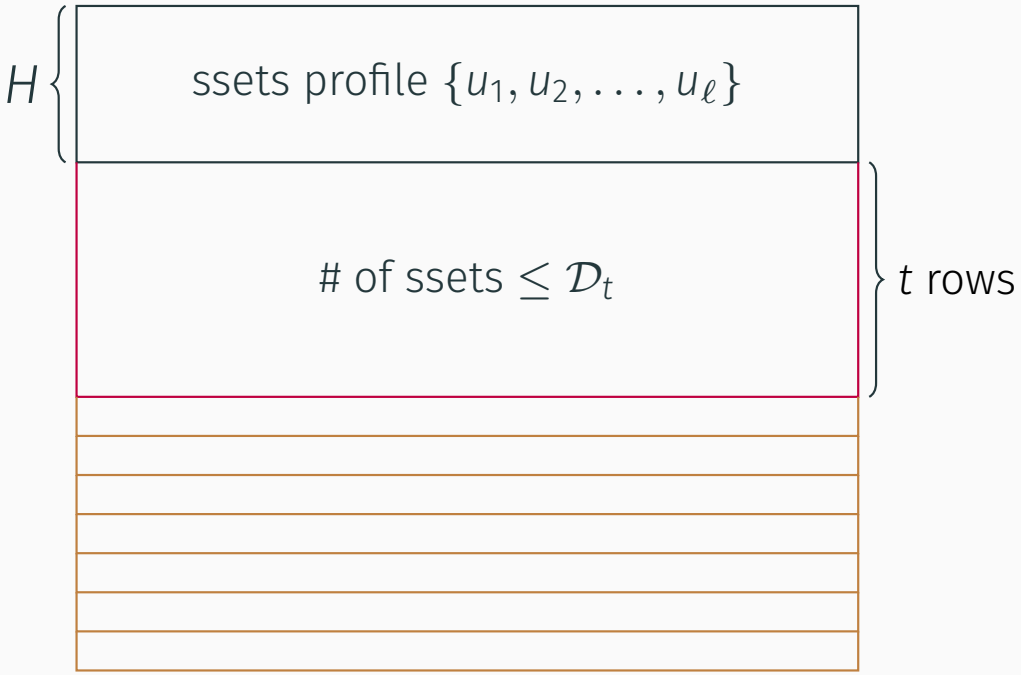
Stopping redundancy – upper bound (ii)

Upper bound construction



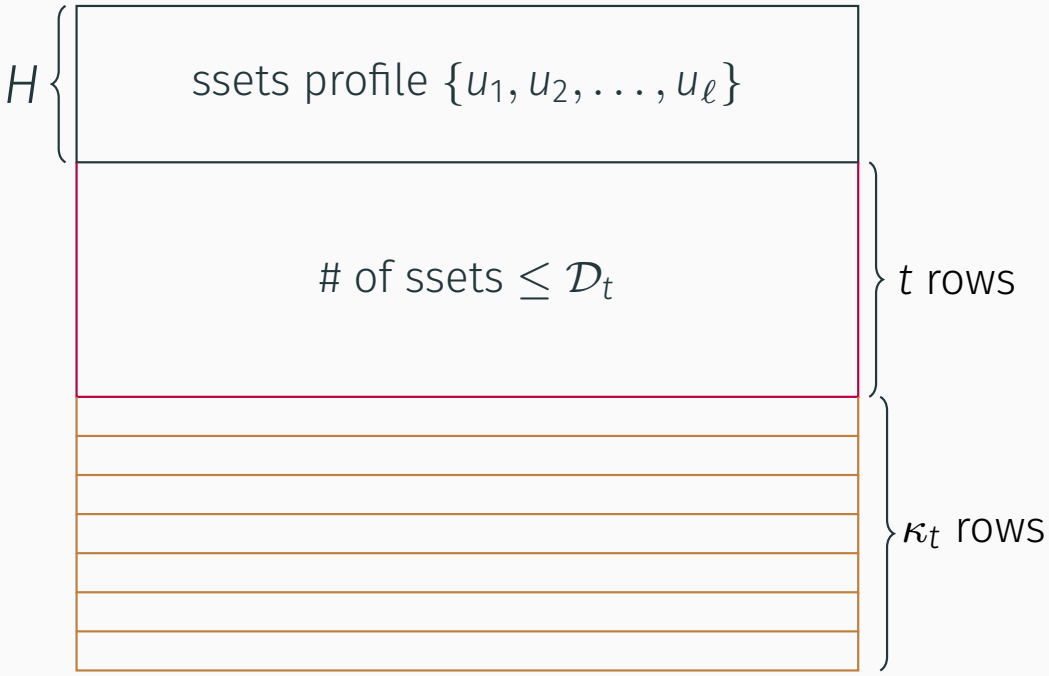
Stopping redundancy – upper bound (ii)

Upper bound construction



Stopping redundancy – upper bound (ii)

Upper bound construction



Stopping redundancy – upper bound (ii)

Upper bound[2,3] on stopping redundancy

$$\rho_\ell \leq n - k + \min_{t \in \mathbb{N}} \{t + \kappa_t\}$$

where

$$\mathcal{D}_t = \sum_{i=1}^{\ell} u_i \prod_{j=n-k+1}^{n-k+t} \left(1 - \frac{i \cdot 2^{n-k-i}}{2^{n-k} - j}\right)$$

$$P_{t,j}(x) = \left\lfloor x \left(1 - \frac{\ell \cdot 2^{n-k-\ell}}{2^{n-k} - (n-k+t+j)}\right) \right\rfloor$$

$$\kappa_t = \min \left\{ j \in \mathbb{N} : P_{t,j}(P_{t,j-1}(\ldots P_{t,1}(\lfloor \mathcal{D}_t \rfloor)\ldots)) = 0 \right\}$$

---

[2] Han, Siegel, Vardy. (2008). Improved probabilistic bounds on stopping redundancy.

[3] Yakimenka, Skachek. (2015). Refined upper bounds on stopping redundancy of binary linear codes.

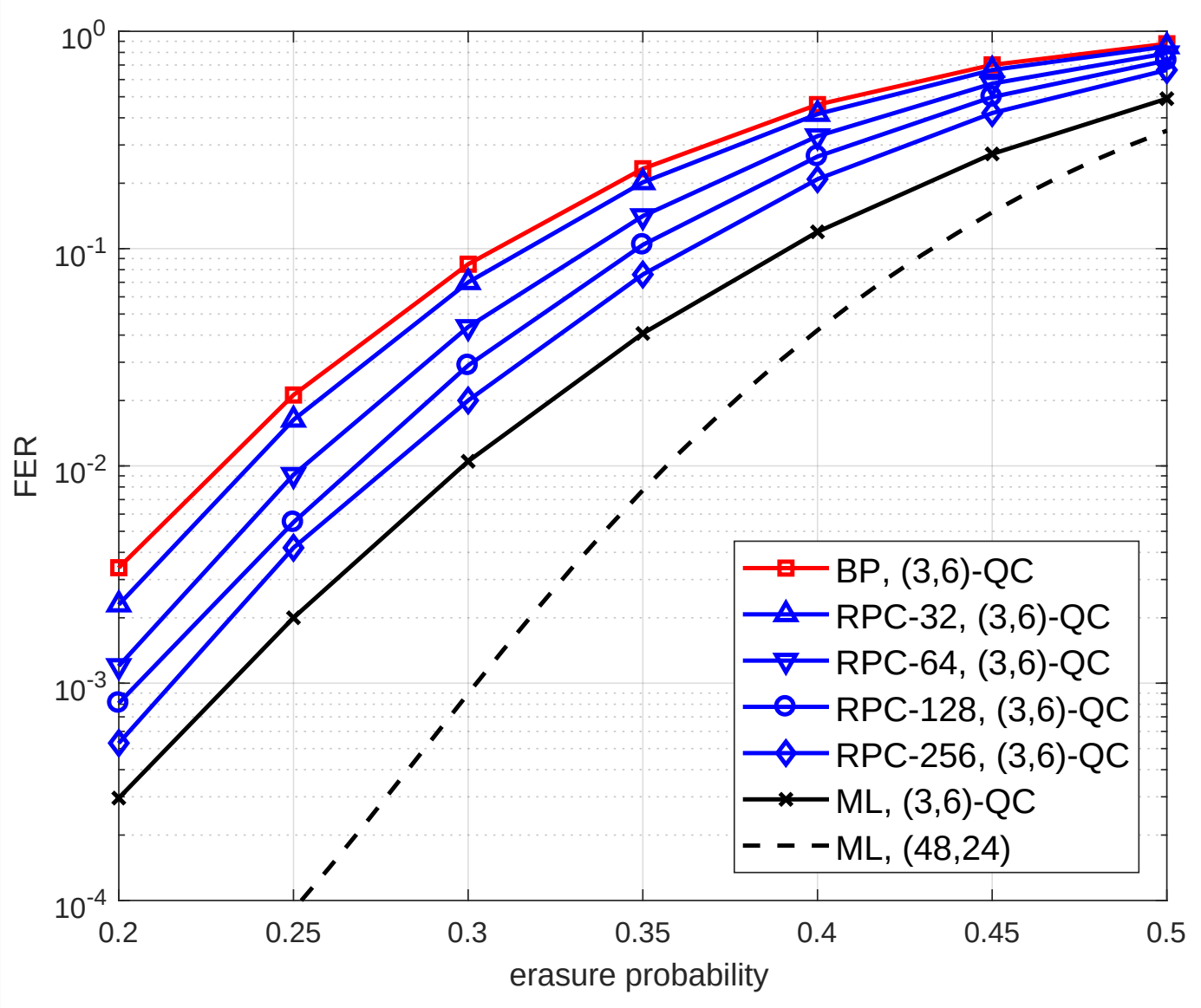
Considered codes

Parameters of studied $[48, 24]$-codes

$d_{\min}$	$A_{d_{\min},n}$	d_{stop}	d_{dual}	g	(J, K)	$\rho_{d_{\min}}, \rho_{d_{\min}+1}$	ρ_r	Type
12	**17296**	**4**	**12**	**4**	$(6, 12)$	**6240,12151**	$\mathbf{1.38 \times 10^7}$	**'L'**
8	13	4	6	4	$(6, 12)$	261,581	1.37×10^7	'RU'
7	1	5	5	4	$(4, 8)$	83,175	1.25×10^7	'RU'
7	**8**	**7**	**5**	**6**	$(3, 6)$	**58,130**	$\mathbf{0.99 \times 10^7}$	**'QC'**
8	**7**	**4**	**7**	**4**	$(3, 6)$	**355,751**	$\mathbf{1.38 \times 10^7}$	**'NB'**

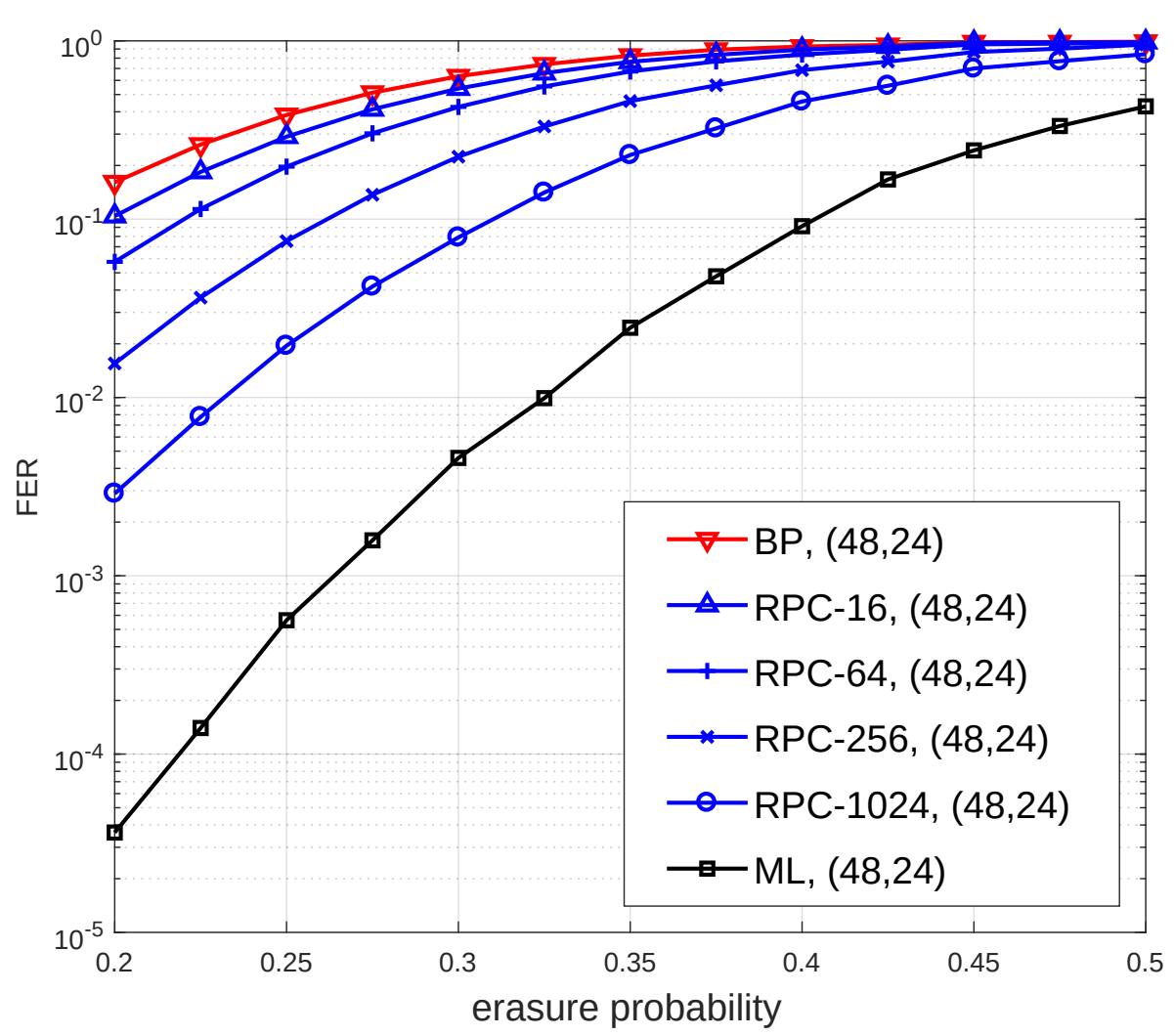
Simulations: FER performance

QC $(3, 6)$-regular code



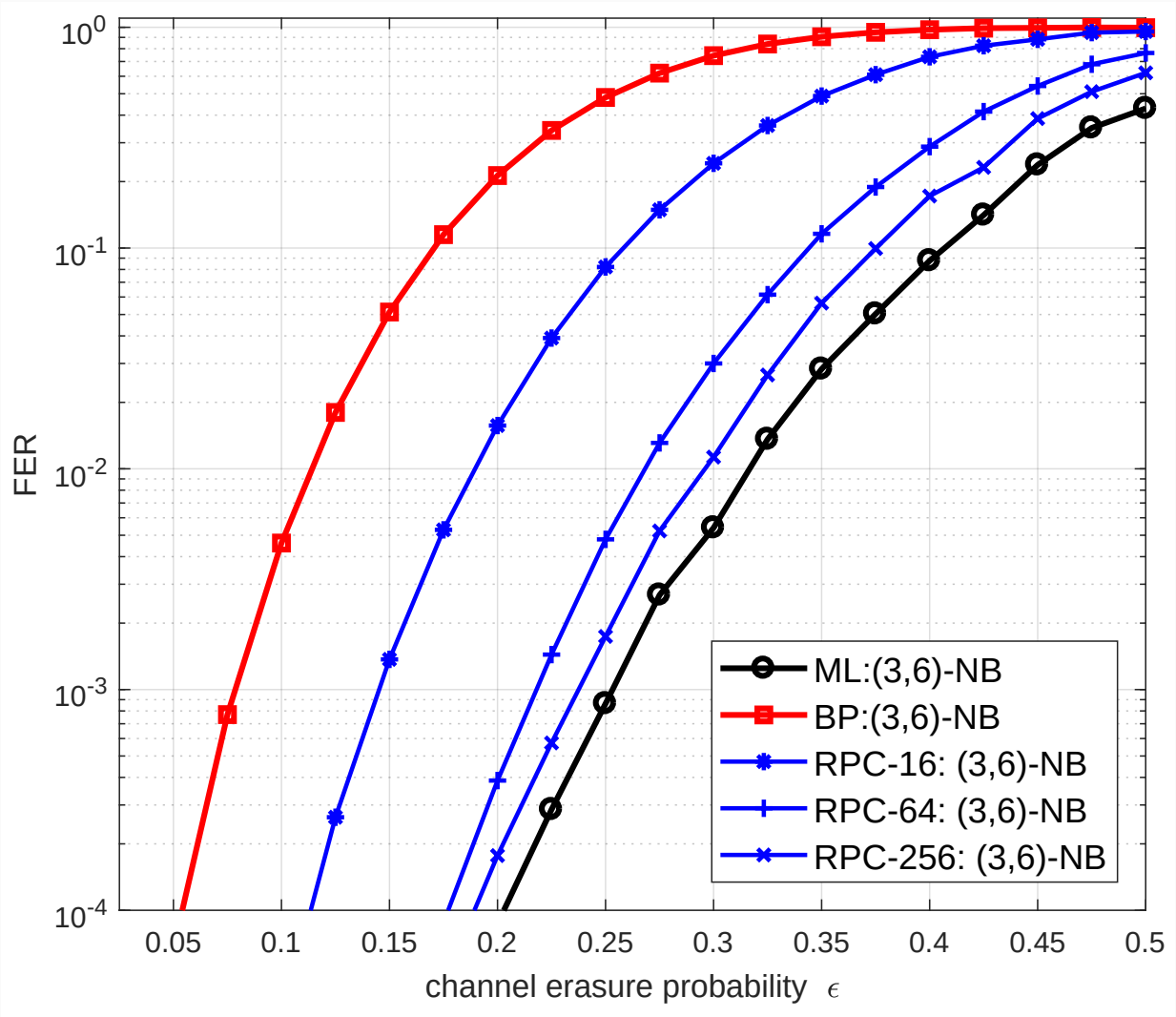
Good convergence to ML, but ML performance is poor

Linear $[48, 24, 12]$ code



RPC is efficient enough, but convergence to ML is slow

Binary image of non-binary $(3, 6)$-code



Both convergence and ML performance are good

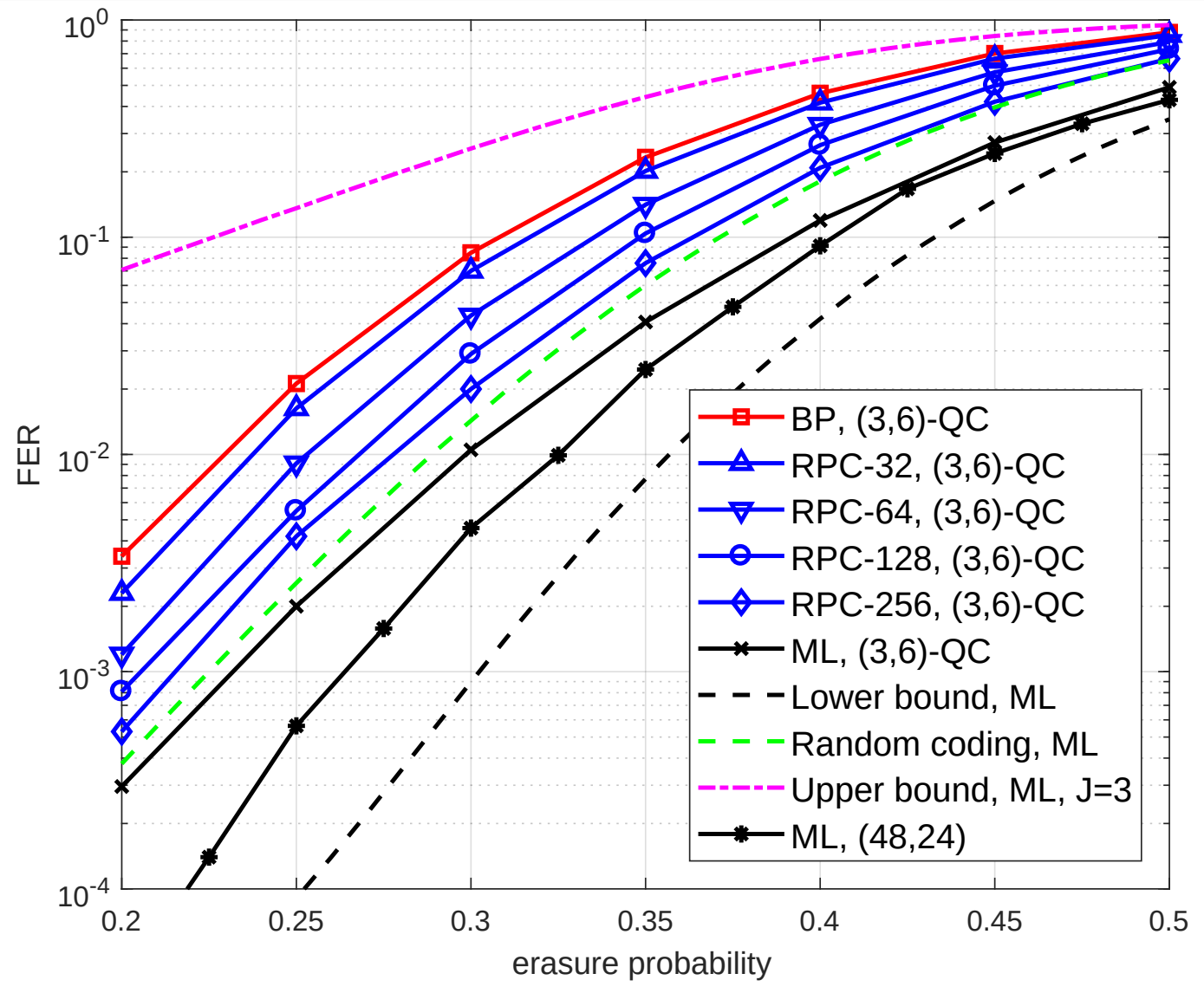
Spectra and bounds

Spectra

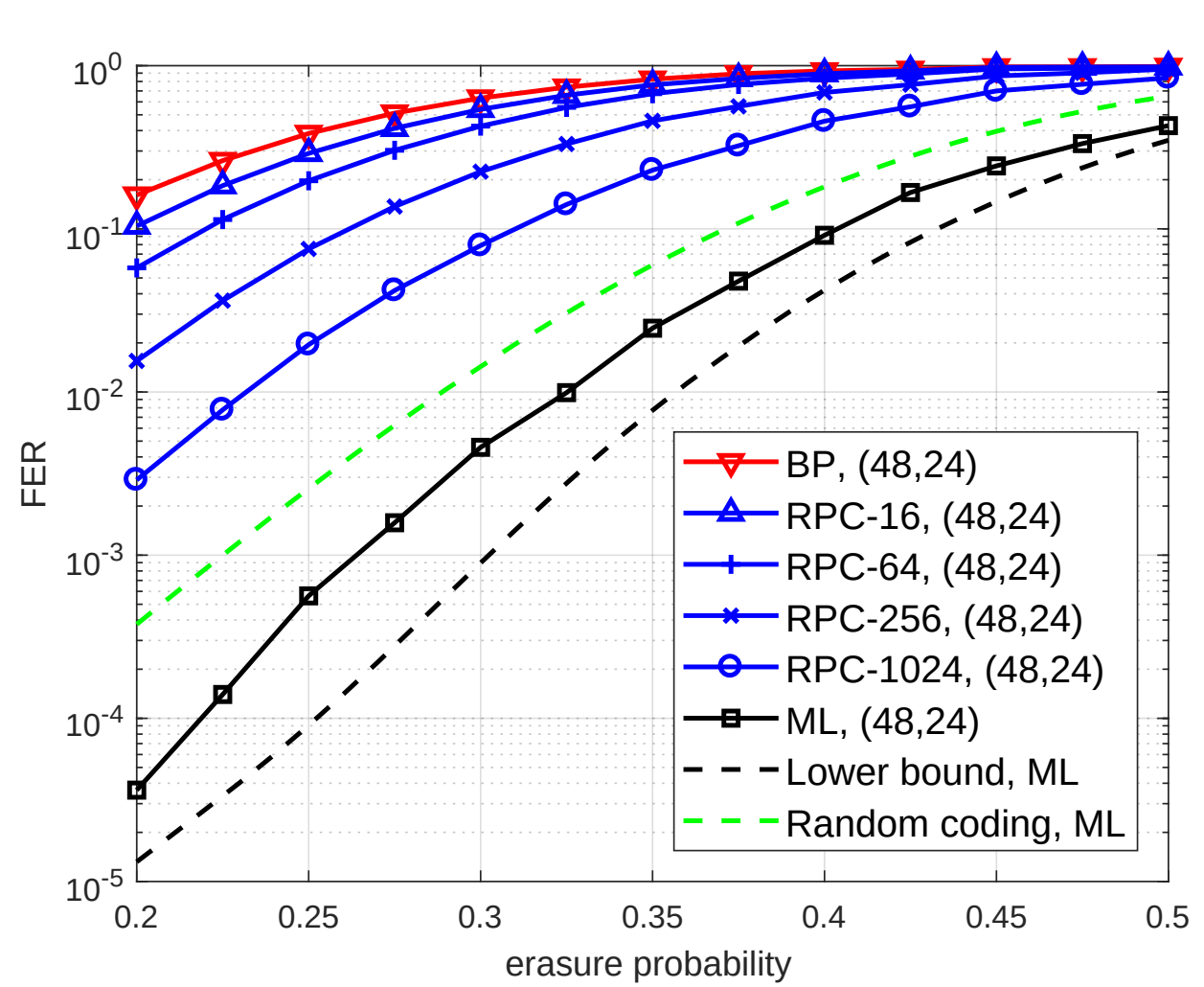
We consider the following spectra for ensembles:

- Distance spectra
- Stopping sets spectra
- Stopping sets spectra for binary images of non-binary codes (**gave us bounds for BP decoding**)

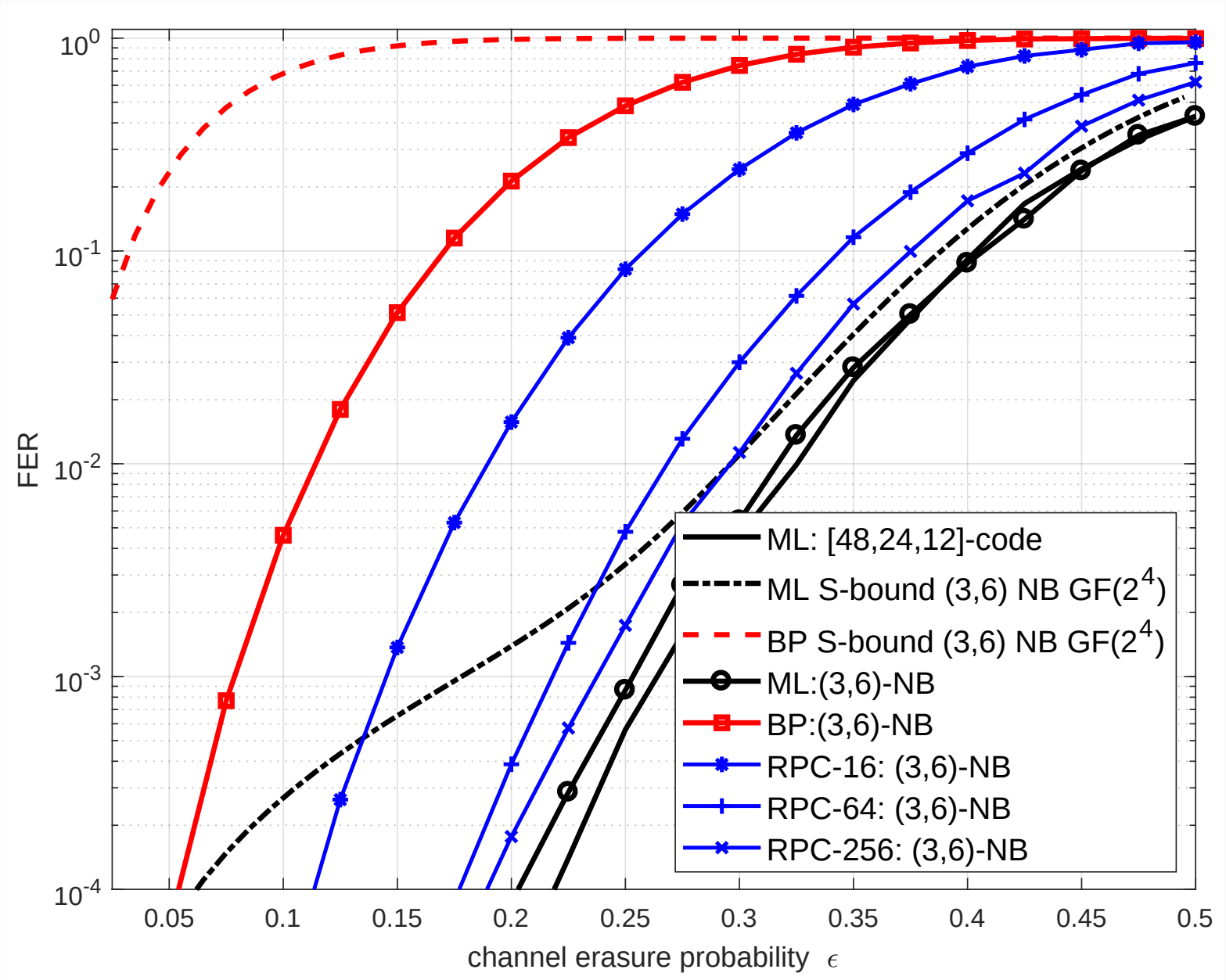
QC $(3, 6)$-regular code



Linear $[48, 24, 12]$ code

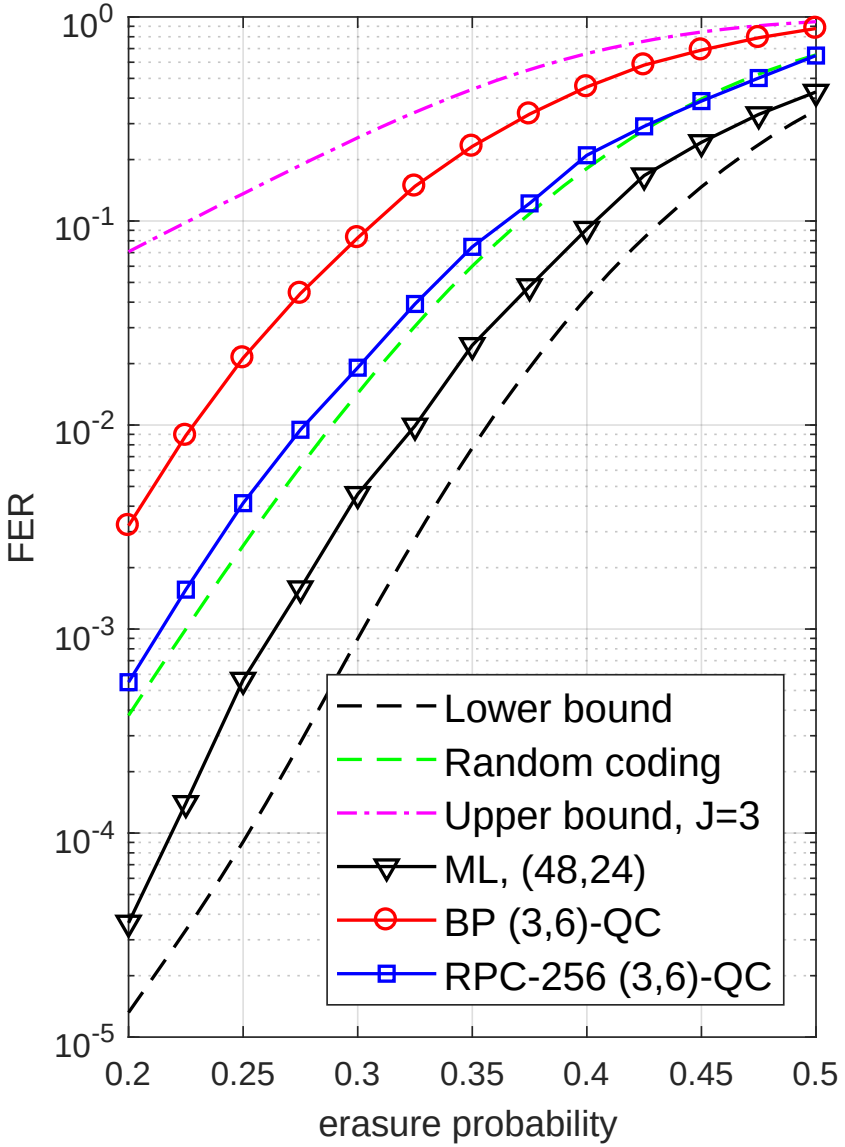


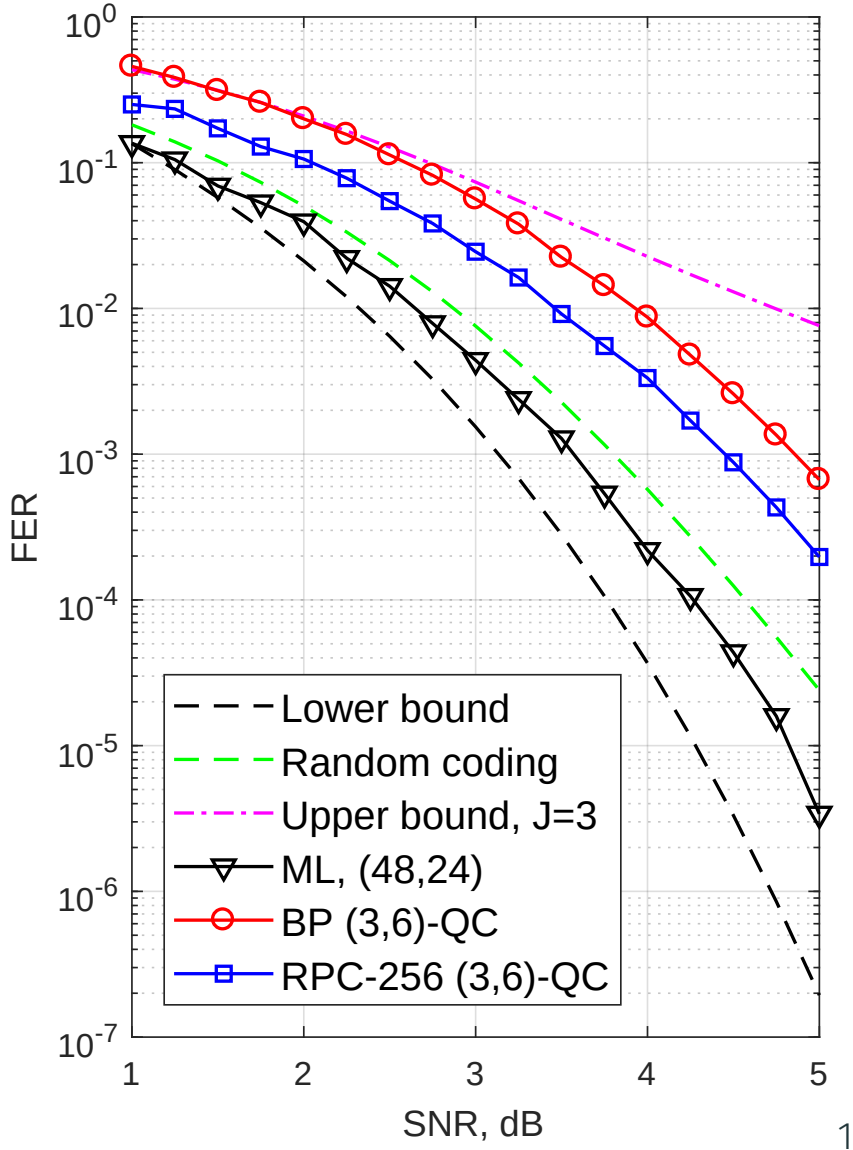
Binary image of non-binary $(3, 6)$-code



But what about BAWGN channel?

BEC vs BAWGNC





Conclusion

- Near-ML decoding converges to ML decoding with increasing number of redundant rows (but requires exponential number of rows)

Conclusion

- Near-ML decoding converges to ML decoding with increasing number of redundant rows (but requires exponential number of rows)
- However, some improvement can be achieved even with a relatively small number of redundant rows

Conclusion

- Near-ML decoding converges to ML decoding with increasing number of redundant rows (but requires exponential number of rows)
- However, some improvement can be achieved even with a relatively small number of redundant rows
- There is a “soft threshold” – to overcome it one needs plenty of redundant rows

Conclusion

- Near-ML decoding converges to ML decoding with increasing number of redundant rows (but requires exponential number of rows)
- However, some improvement can be achieved even with a relatively small number of redundant rows
- There is a “soft threshold” – to overcome it one needs plenty of redundant rows
- NB codes are a good compromise:

Conclusion

- Near-ML decoding converges to ML decoding with increasing number of redundant rows (but requires exponential number of rows)
- However, some improvement can be achieved even with a relatively small number of redundant rows
- There is a “soft threshold” – to overcome it one needs plenty of redundant rows
- NB codes are a good compromise:
    1. ML performance close to the ML performance of best linear codes;

Conclusion

- Near-ML decoding converges to ML decoding with increasing number of redundant rows (but requires exponential number of rows)
- However, some improvement can be achieved even with a relatively small number of redundant rows
- There is a “soft threshold” – to overcome it one needs plenty of redundant rows
- NB codes are a good compromise:
  1. ML performance close to the ML performance of best linear codes;
  2. BP performance converges rather fast (due to their suitability for iterative decoding?)

Conclusion

- Near-ML decoding converges to ML decoding with increasing number of redundant rows (but requires exponential number of rows)
- However, some improvement can be achieved even with a relatively small number of redundant rows
- There is a “soft threshold” – to overcome it one needs plenty of redundant rows
- NB codes are a good compromise:
  1. ML performance close to the ML performance of best linear codes;
  2. BP performance converges rather fast (due to their suitability for iterative decoding?)
- **Adding redundant rows works on BAWGNC too!**

Open problem

What code we want to contstruct for RPC:

- large d_{min}
- large d_{stop}
- small d_{dual}

Thank you

Just in case

Ensemble-Average Spectra

(J, K)-regular Gallager ensemble

Ensemble of (J, K)-regular parity-check matrices of LDPC $[n, k]$-codes

$$H = \begin{pmatrix} 1\,1\,1\,1\,1\,1 & & & \\ & 1\,1\,1\,1\,1\,1 & & \\ & & 1\,1\,1\,1\,1\,1 & \\ & & & 1\,1\,1\,1\,1\,1 \\ \hline \\ \hline \\ \end{pmatrix}$$

Each $M \times n$ strip is a permutation of columns of the first strip.

Weight-generating functions

$$G_n(s) = \sum_{w=0}^{n} A_{n,w} s^w$$

Recurrent coefficient calculation

Let

$$f(s) = \sum_{\ell \geq 0} f_\ell s^\ell$$

$$F_L(s) = (f(s))^L = \sum_{\ell \geq 0} F_{\ell,L} s^\ell$$

then

$$F_{\ell,L} = \begin{cases} f_\ell\,, & L = 1 \\ \sum_{i=0}^{\ell} f_i F_{\ell-i,L-1}\,, & L > 1 \end{cases}$$

Average weight spectrum

One row of H

$$g(s) = \sum_{i \text{ even}} \binom{K}{i} s^i = \frac{(1+s)^K + (1-s)^K}{2}$$

One strip of H

$$G(s) = \sum_{w=0}^{n} N_{n,w} s^w = (g(s))^M$$

Ensemble-average spectrum coefficients

$$E\{A_{n,w}\} = \binom{n}{w} (p(w))^J = \binom{n}{w}^{1-J} N_{n,w}^J$$

Other spectra

Stopping set spectrum

$$g(s) = \sum_{w=0,2,3,\ldots,K} \binom{K}{w} s^w = (1+s)^K - Ks$$

Weight spectrum

$$\phi(s) = \frac{1}{q-1} \sum_{w=1}^{m} \binom{m}{w} s^w = \frac{(1+s)^m - 1}{q-1}$$

$$g(\phi) = \frac{(1+(q-1)\phi)^K + (q-1)(1-\phi)^K}{q}$$

Can calculate fast!

Calculation of $d_{\min}$, d_{stop}

Minimum distance

$$\sum_{w=0}^{d_{\min}-1} A_{n,w} < 1$$

Stopping distance

$$\sum_{w=0}^{d_{\text{stop}}-1} B_{n,w} < 1$$

Numerical Results

Observations

- Random regular LDPC codes (esp. non-binary) with optimised J have minimum distance close to random linear codes

Observations

- Random regular LDPC codes (esp. non-binary) with optimised J have minimum distance close to random linear codes
- ...and gap decreases with decrease of R

Observations

- Random regular LDPC codes (esp. non-binary) with optimised J have minimum distance close to random linear codes
- ...and gap decreases with decrease of R
- for LDPC, stopping distances are about half of min distances (for best LDPC – equal)

Example

Table 2: Examples of codes from the Gallager $(J, 2J)$ ensemble

(n, k, d)	J	$\hat{d}$	d_{GV}	d_{L}	d_{stop}	$\hat{d}_{\mathrm{stop}}$	ρ	$\hat{\rho}$
(40,24,6)	5	4	5	7	6	4	73	20
(60,35,8)	6	6	7	10	6	6	289	76
(90,49,10)	5	8	11	14	8	8	1118	86

Asymptotic Analysis

For binary images ($q = 2^m$)

Asymptotics for binary images of Gallager ensemble of nonbinary LDPC codes over $\mathbb{F}_q = \mathbb{F}_{2^m}$ (following Gallager)

$$E\{A_{n,w}\} \leq \binom{nm}{w}^{1-J} (g(\phi(s)))^{MJ} s^{-wJ}, \qquad \forall s$$

Replace s by e^ρ, find critical value $\delta = w/(mn)$, where

$$\lim_{n\to\infty} \frac{\ln E\{A_{nm,w}\}}{nm} = \min_\rho \left\{ (1-J) h_e(\delta) + \frac{J}{Km} \ln(g(\phi(e^\rho))) - \rho\delta J \right\} = 0$$

Normalised minimum distances

Table 3: Normalized minimum distances for binary and nonbinary LDPC code ensembles. Numbers in parentheses are typical asymptotic normalized stopping distances.

m	J				
	2	3	4	5	10
1	0	0.0227	0.0627	0.0843	0.1083
		(0.0180)	(0.0454)	(0.0580)	(0.0619)
2	0	0.0398	0.0774	0.0943	0.1095
3	0	0.0559	0.0880	0.1007	0.1099
4	0.0018	0.0686	0.0952	0.1046	0.1100
8	0.0396	0.0938	0.1066	0.1094	0.1100
16	0.0688	0.1049	0.1097	0.1100	0.1100