RECURSIVE SUBPRODUCT CODES

Siddheshwar, Natarajan and Krishnan, https://arxiv.org/abs/2401.15678

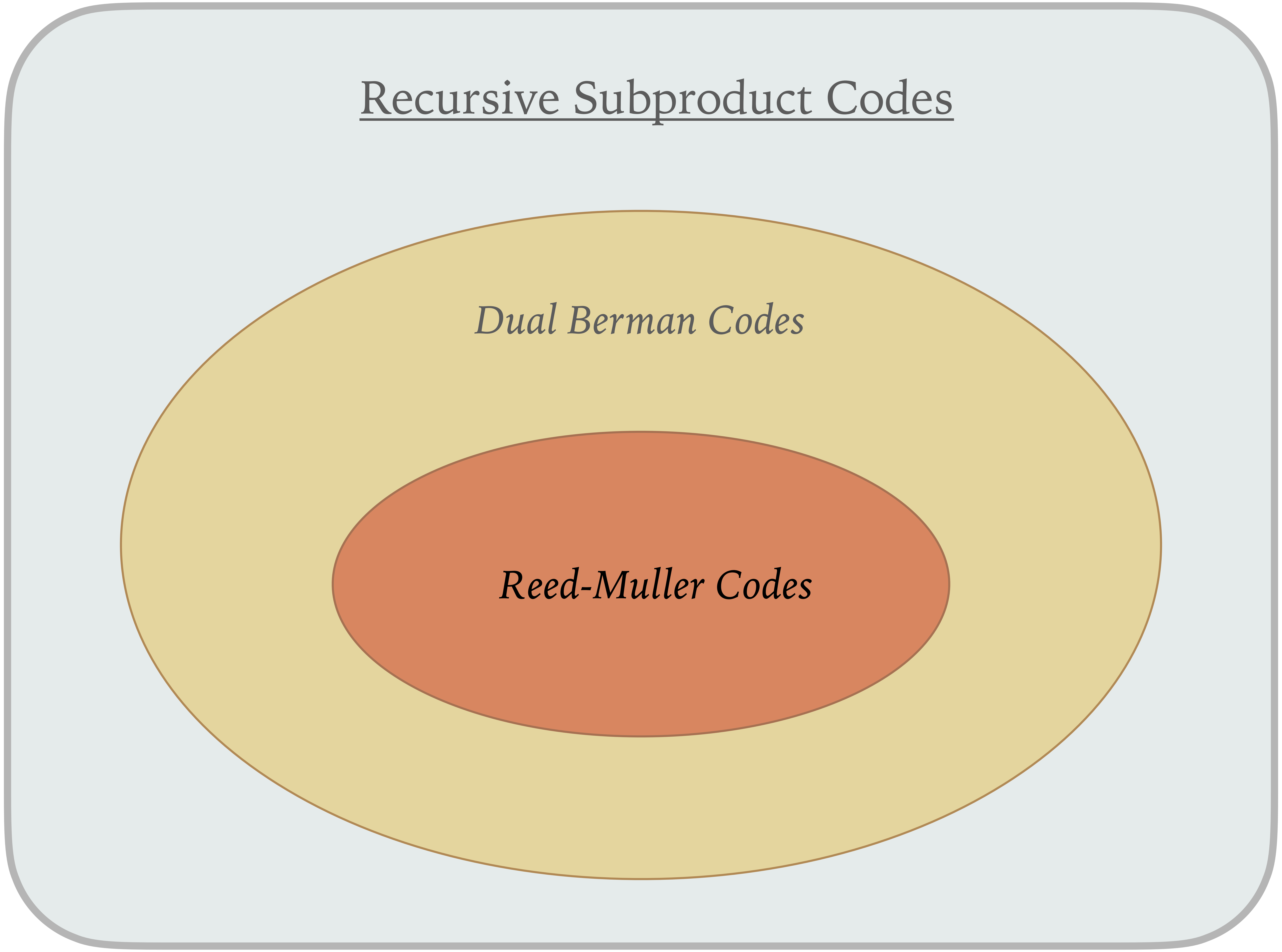
09 Feb 2024

Supported by Qualcomm 6G University Research India Program

MOTIVATION

1. Recently identified a family of codes that are capacity-achieving in erasure channel: *Berman codes and their dual codes* [Natarajan & Krishnan, T-IT '23]
   - Possess several properties (but not all) required to achieve capacity of all binary-input memoryless symmetric (BMS) channels [Reeves & Pfister, T-IT '24]
   - **Can these codes be decoded efficiently in the AWGN channel? Performance?**

2. Identify codes that share the structure of Reed-Muller codes
   - Especially the ones that are needed to prove capacity-achievability [Reeves & Pfister, T-IT '24]
   - A 'projection' property that can be exploited for **iterative decoding**
   - Block **lengths other than** 2^m



- Wider range of rates and block lengths compared to Reed-Muller (RM) codes
- First-order codes can be ML decoded efficiently (similar to RM codes)
- BP decoding and a local graph search decoder for second-order codes (similar to RM codes)
- CER comparable to RM codes and Polar codes

SUMMARY OF THE FAMILY OF CODES

- Choose any $[n, k, d]$ 'base code' $\mathscr{C}$ that contains the all-ones codeword

- Pick r, m such that $0 \leq r \leq m$
  - r is the 'order' of the recursive subproduct code

- Recursive Subproduct Code $\mathscr{C}^{\otimes[r,m]}$ has parameters

$$\left[n^m,\ \sum_{\ell=0}^{r} \binom{m}{\ell} (k-1)^{\ell},\ d^r n^{m-r} \right]$$

SUMMARY OF THE FAMILY OF CODES

- Recursive Subproduct Code $\mathscr{C}^{\otimes[r,m]} = \left[n^m,\ \sum_{\ell=0}^{r} \binom{m}{\ell}(k-1)^{\ell},\ d^r n^{m-r} \right]$

- Chain of codes: $\{\mathbf{0},\mathbf{1}\} = \mathscr{C}^{\otimes[0,m]} \subset \mathscr{C}^{\otimes[1,m]} \subset \cdots \subset \mathscr{C}^{\otimes[m,m]} = \mathscr{C}^{\otimes m}$

  - $\mathscr{C}^{\otimes m}$ is the m-dimensional product code with parameters $[n^m, k^m, d^m]$
    - Codewords are $\underbrace{n \times \cdots \times n}_{m \text{ times}}$ arrays
    - each length-n vector along any of the m dimensions belongs to $\mathscr{C}$

  - $\mathscr{C}^{\otimes[r,m]}$ is a subcode of the product code (*subproduct code*)

SUMMARY OF THE FAMILY OF CODES

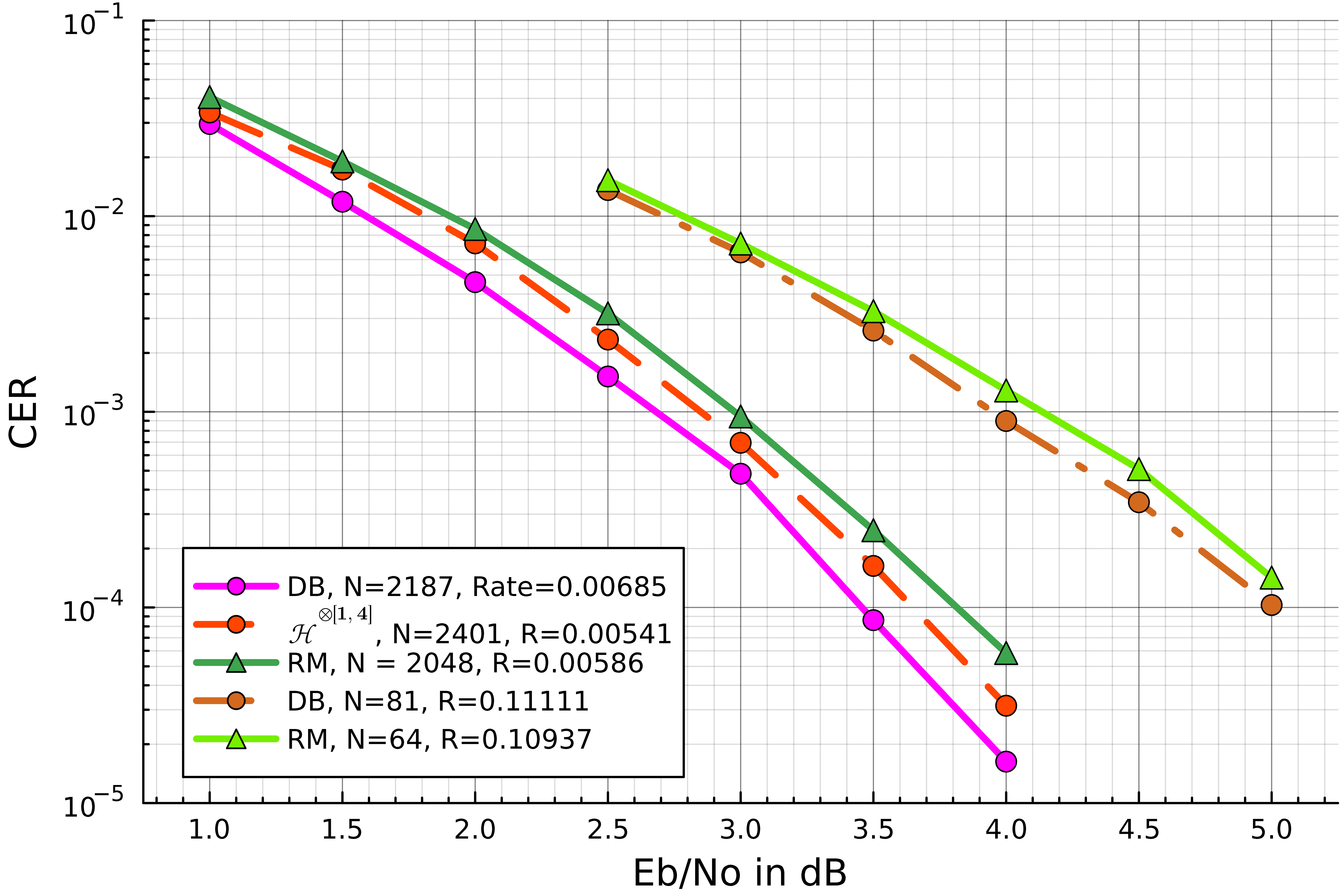
- Recursive Subproduct Code $\mathscr{C}^{\otimes[r,m]} = \left[n^m,\ \sum_{\ell=0}^{r} \binom{m}{\ell}(k-1)^{\ell},\ d^r n^{m-r} \right]$

- Reed-Muller Codes:
  - Choose $\mathscr{C} = \mathbb{F}_2^2 = \{(0,0),(0,1),(1,0),(1,1)\}$, which is a $[n,k,d] = [2,2,1]$ code

- Dual Berman Codes:
  - Choose $\mathscr{C} = \mathbb{F}_2^n$, which is a $[n,k,d] = [n,n,1]$ code

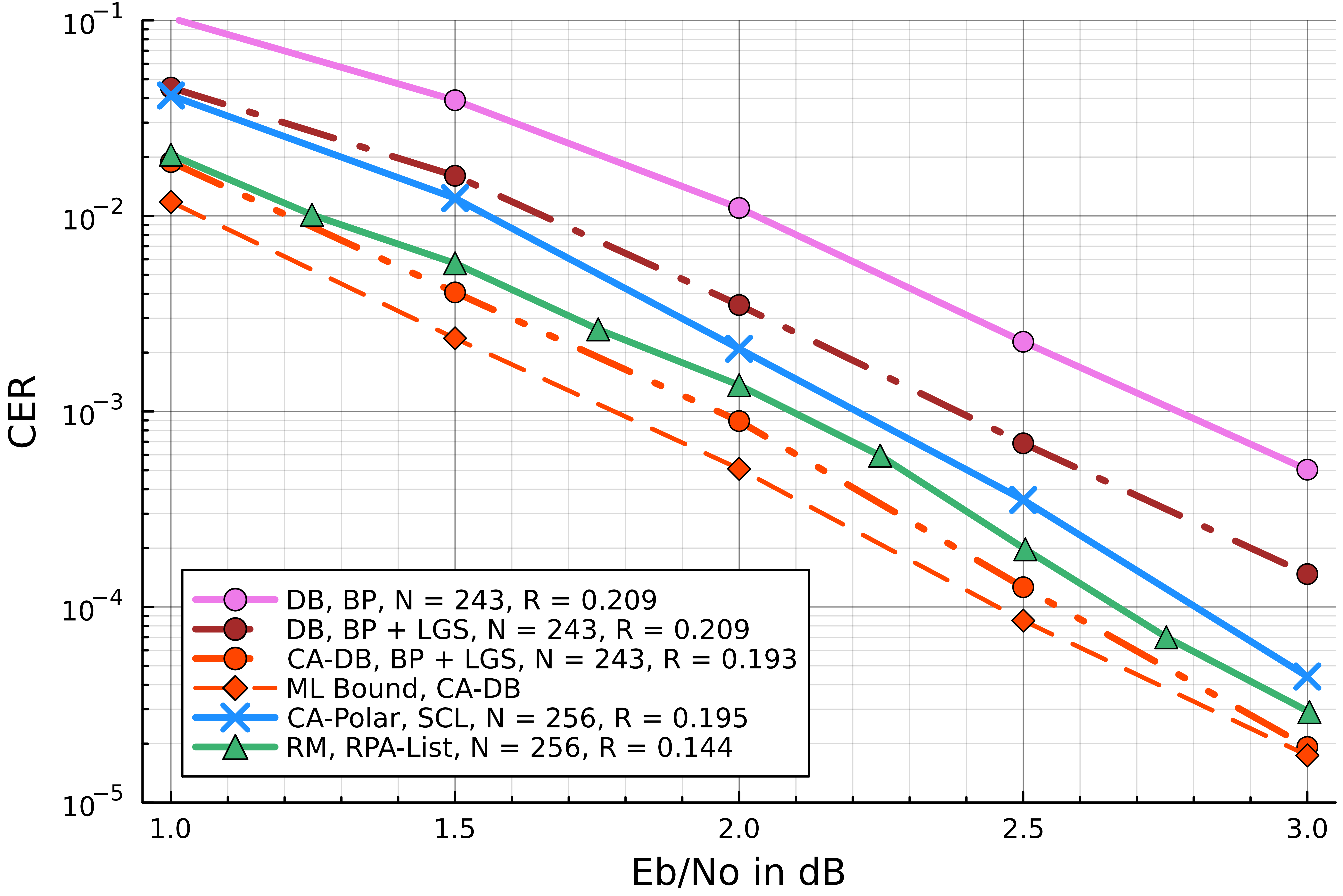
SIMULATION RESULTS

FIRST-ORDER CODES UNDER ML DECODING



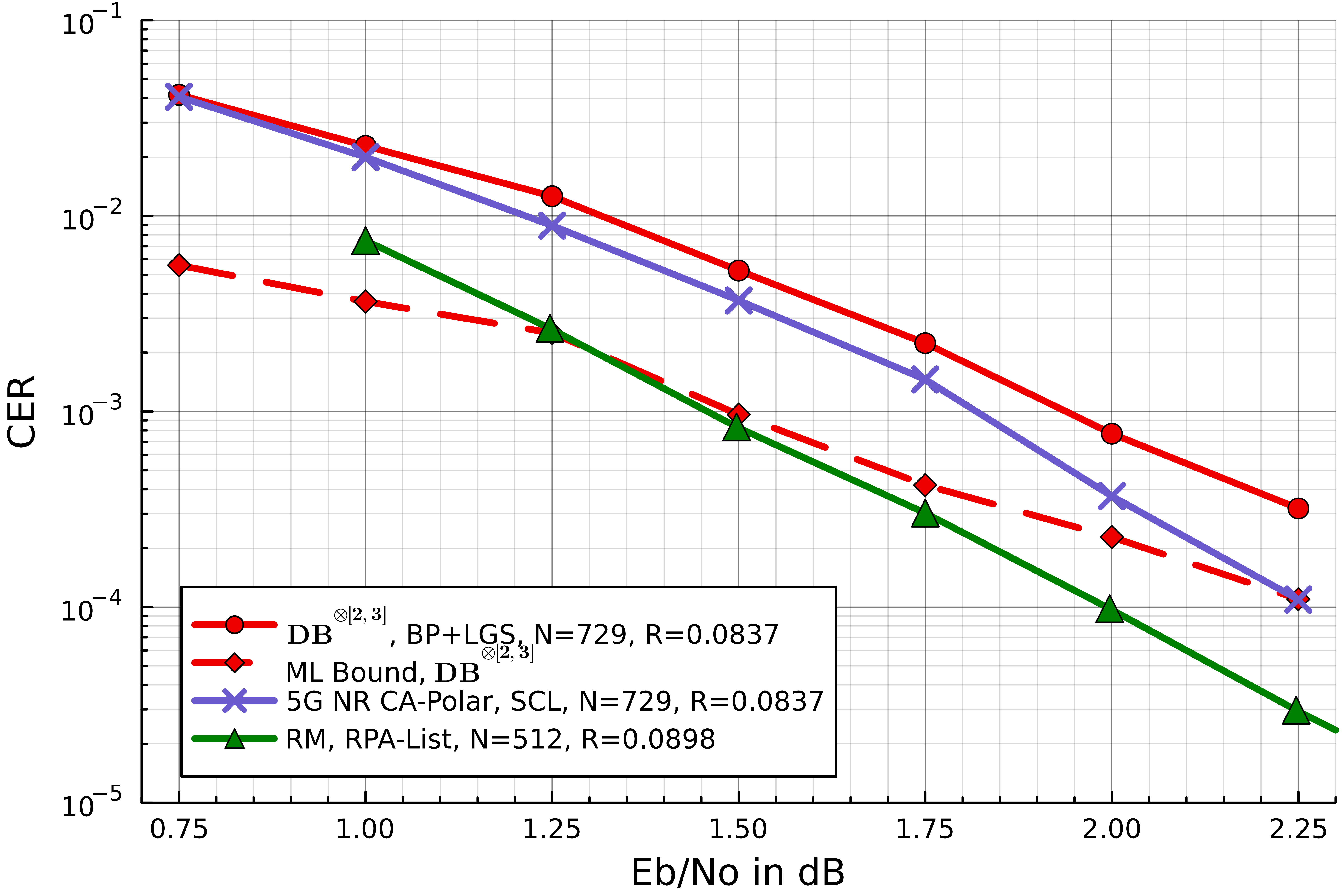
- ‘DB’ is Dual Berman with $\mathscr{C} = \mathbb{F}_2^3$
- $\mathscr{H}$ = [7,4,3] Hamming code

SECOND-ORDER CODES



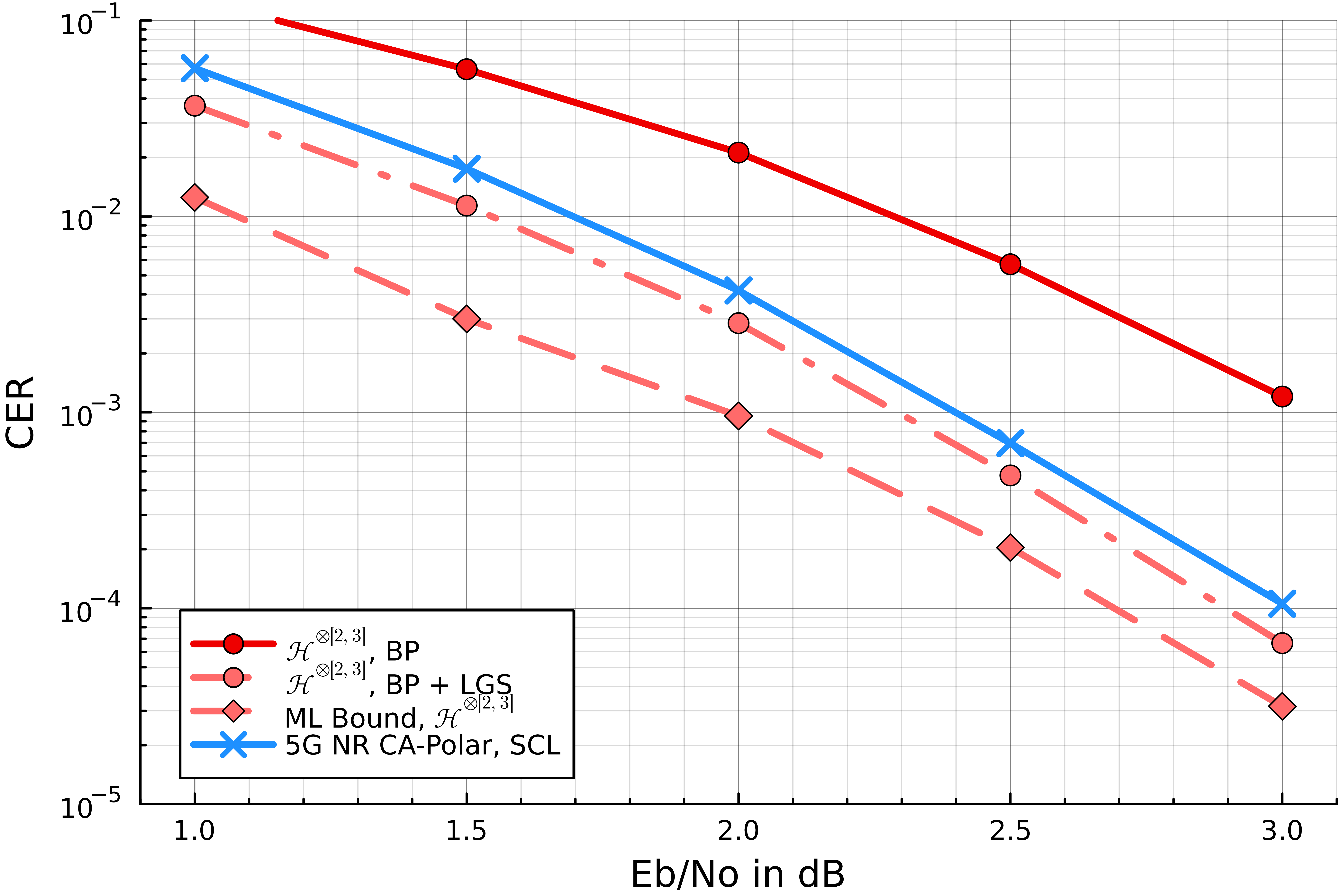
- ‘DB’ is Dual Berman with $\mathscr{C} = \mathbb{F}_2^3$
- ‘CA-DB’ is CRC-aided DB with 4-bit CRC
- ‘CA-Polar’ is CRC-aided Polar with 8-bit CRC, SCL decoding with list size 32

SECOND-ORDER CODES



- DB is the [9,5,3] Dual Berman code
  - used as the base code
- 5G NR Polar code uses 11-bit CRC, rate-matching to get length 729, SCL with list size 32

SECOND-ORDER CODES



- Both codes have length 343 and dimension 37
- $\mathscr{H}$ = [7,4,3] is the base code
- 5G NR Polar code uses 11-bit CRC, rate-matching to get length 343, SCL with list size 32

CONSTRUCTION OF RECURSIVE SUBPRODUCT CODES

CONSTRUCTION

- Start with a generator matrix for the $[n, k, d]$ code $\mathscr{C}$ with first row being all-ones

$$G = \begin{bmatrix} \mathbf{g}_0 = \mathbf{1} \\ \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_{k-1} \end{bmatrix}$$

- For each $\mathbf{j} = (j_0, \ldots, j_{m-1}) \in [k]^m$, where $[k] = \{0,1,\ldots,k-1\}$, define

$$\mathbf{b}_{\mathbf{j}} = \mathbf{g}_{j_0} \otimes \mathbf{g}_{j_1} \otimes \cdots \otimes \mathbf{g}_{j_{m-1}} \quad \in \mathbb{F}_2^{n^m}$$

- Define Hamming weight $\mathrm{w}(\mathbf{j})$ as number of non-zero entries in $\mathbf{j}$

EXAMPLE

- Suppose $\mathscr{C} = [4,3,2]$ code with

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

- If $\quad \mathbf{j} = (j_0, j_1, j_2, j_3) = (1,0,2,0) \quad \in [k]^m = [3]^4 \quad$ then

$$\mathbf{b_j} \quad = \quad (1,1,0,0) \otimes (1,1,1,1) \otimes (0,1,1,0) \otimes (1,1,1,1) \quad \in \mathbb{F}_2^{256}$$

CONSTRUCTION OF RECURSIVE SUBPRODUCT CODES

- For $0 \le r \le m$ define

$$\mathscr{C}^{\otimes[r,m]} = \operatorname{span}\left(\left\{ \mathbf{b_j} \ : \ \mathbf{j} \in [k]^m, \ \mathrm{w}(\mathbf{j}) \le r \right\} \right)$$

- The vectors $\mathbf{b_j} : \mathbf{j} \in [k]^m$ are linearly independent
  - They are the rows of the matrix $G^{\otimes m} = G \otimes \cdots \otimes G$, which is the generator matrix of the product code $\mathscr{C}^{\otimes m}$
  - Subproduct codes are constructed by choosing a submatrix of $G^{\otimes m}$ as gen. mat.

RECURSIVE STRUCTURE

- Construct length-n^m codes via length-n^{m-1} codes:

$$\mathscr{C}^{\otimes[r,m]} = \left\{ \mathbf{d}_0 \otimes \mathbf{g}_0 + \mathbf{d}_1 \otimes \mathbf{g}_1 + \cdots + \mathbf{d}_{k-1} \otimes \mathbf{g}_{k-1} \;:\; \mathbf{d}_0 \in \mathscr{C}^{\otimes[r,m-1]},\ \mathbf{d}_1, \ldots, \mathbf{d}_{k-1} \in \mathscr{C}^{\otimes[r-1,m-1]} \right\}$$

- $G = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ yields the famous $(u \mid u + v)$ Plotkin design of RM codes

- Useful in identifying the min. distance, puncturing and ‘projection’ properties

FAST ML DECODING OF FIRST-ORDER CODES

RECURSIVE STRUCTURE OF FIRST-ORDER CODES

- Construct length-n^m codes via length-n^{m-1} codes:

$$\mathscr{C}^{\otimes[1,m]} = \left\{ \mathbf{d}_0 \otimes \mathbf{g}_0 + \mathbf{d}_1 \otimes \mathbf{g}_1 + \cdots + \mathbf{d}_{k-1} \otimes \mathbf{g}_{k-1} \ : \ \mathbf{d}_0 \in \mathscr{C}^{\otimes[1,m-1]},\ \mathbf{d}_1, \ldots, \mathbf{d}_{k-1} \in \mathscr{C}^{\otimes[0,m-1]} \right\}$$

- BPSK-modulated version: $0 \to +1$ and $1 \to -1$

$$\mathscr{C}^{\otimes[1,m]} = \left\{ \mathbf{d}^b \otimes \mathbf{a}^b \ : \ \mathbf{d} \in \mathscr{C}^{\otimes[1,m-1]}, \mathbf{a} \in \mathscr{C}_{\text{sub}} \right\}$$

where $\mathscr{C}_{\text{sub}} = \text{span}(\mathbf{g}_1, \ldots, \mathbf{g}_{k-1}) \subset \mathscr{C}$

RECURSIVE ML DECODING OF FIRST-ORDER CODES

$$\mathscr{C}^{\otimes[1,m]} = \left\{ \ \mathbf{d}^b \otimes \mathbf{a}^b \ : \ \mathbf{d} \in \mathscr{C}^{\otimes[1,m-1]}, \ \mathbf{a} \in \mathscr{C}_{\mathrm{sub}} \right\}$$

- For a given $\mathbf{a} \in \mathscr{C}_{\mathrm{sub}}$ the best choice of $\mathbf{d}$ can be found by calling the ML decoder of $\mathscr{C}^{\otimes[1,m-1]}$
- Call the ML decoder of $\mathscr{C}^{\otimes[1,m-1]}$ totally 2^{k-1} times to find best choices of $\mathbf{d}, \mathbf{a}$
- Define $\alpha = \dfrac{k-1}{\log_2 n}$. Complexity order (in terms of block length $N = n^m$) is

$$\max\{N, N^{\alpha}\} \text{ if } \alpha \neq 1 \qquad \text{and} \qquad N \log N \text{ if } \alpha = 1$$

SOFT-OUTPUT DECODER FOR FIRST-ORDER CODES

- We need a soft-output decoder for use with iterative decoding of second-order codes

- Identified a recursive version of max-log-MAP decoder
    - Numerically stable (operates in the log domain)
    - Avoids costly operations (exp, log)
    - Good approximation to the optimal decoder (soft-output MAP)

- Complexity order is same as that of the recursive ML decoder

$$\max\{N, N^{\alpha}\} \text{ if } \alpha \neq 1 \qquad \text{and} \qquad N \log N \text{ if } \alpha = 1$$

BELIEF PROPAGATION DECODER FOR SECOND-ORDER CODES

INDEXING USING BASE-n EXPANSION

- Instead of indexing length n^m vectors using $i \in \{1,2,\ldots,n^m\}$ use

$$(i-1) \;=\; i_0\, n^{m-1} + i_1\, n^{m-2} + \cdots + i_{m-2}\, n + i_{m-1}$$

- Indices are vectors $\mathbf{i} = \left(i_0, \ldots, i_{m-1}\right) \in [n]^m$

- Codeword $\mathbf{c} = \left(c_{\mathbf{i}} \;:\; \mathbf{i} \in [n]^m\right)$ where $c_{\mathbf{i}} \in \mathbb{F}_2$
    - Codewords are m-dimensional arrays/tensors

PUNCTURING AND PROJECTION TO LENGTH n^{m-1}

- Pick any coordinate k of $\mathbf{i} = \left(i_0, \ldots, i_{m-1}\right)$ and fix it to some value $u \in [n]$

$$\mathscr{H} = \left\{(i_0, \ldots, i_{m-1}) \in [n]^m \ : \ i_k = u\right\}$$

- Similarly, for the same k, fix this coordinate to some other value $u' \in [n]$

$$\mathscr{H}' = \left\{(i_0, \ldots, i_{m-1}) \in [n]^m \ : \ i_k = u'\right\}$$

- Puncture a codeword $\mathbf{c}$ by retaining only the coordinates in $\mathscr{H}$ and $\mathscr{H}'$

$$\mathscr{P}_{\mathscr{H}}(\mathbf{c}) \quad \text{and} \quad \mathscr{P}_{\mathscr{H}'}(\mathbf{c})$$

PUNCTURING AND PROJECTION TO LENGTH n^{m-1}

$\mathscr{H} = \left\{(i_0, \ldots, i_{m-1}) \in [n]^m \ : \ i_k = u\right\}$

$\mathscr{H}' = \left\{(i_0, \ldots, i_{m-1}) \in [n]^m \ : \ i_k = u'\right\}$

Puncture a codeword $\mathbf{c}$: $\mathscr{P}_{\mathscr{H}}(\mathbf{c})$ and $\mathscr{P}_{\mathscr{H}'}(\mathbf{c})$

- Puncturing Property:

$$\mathscr{P}_{\mathscr{H}}(\mathbf{c}),\ \mathscr{P}_{\mathscr{H}'}(\mathbf{c}) \in \mathscr{C}^{\otimes[r,m-1]} \quad \text{for every } \mathbf{c} \in \mathscr{C}^{\otimes[r,m]}$$

- Projection Property:

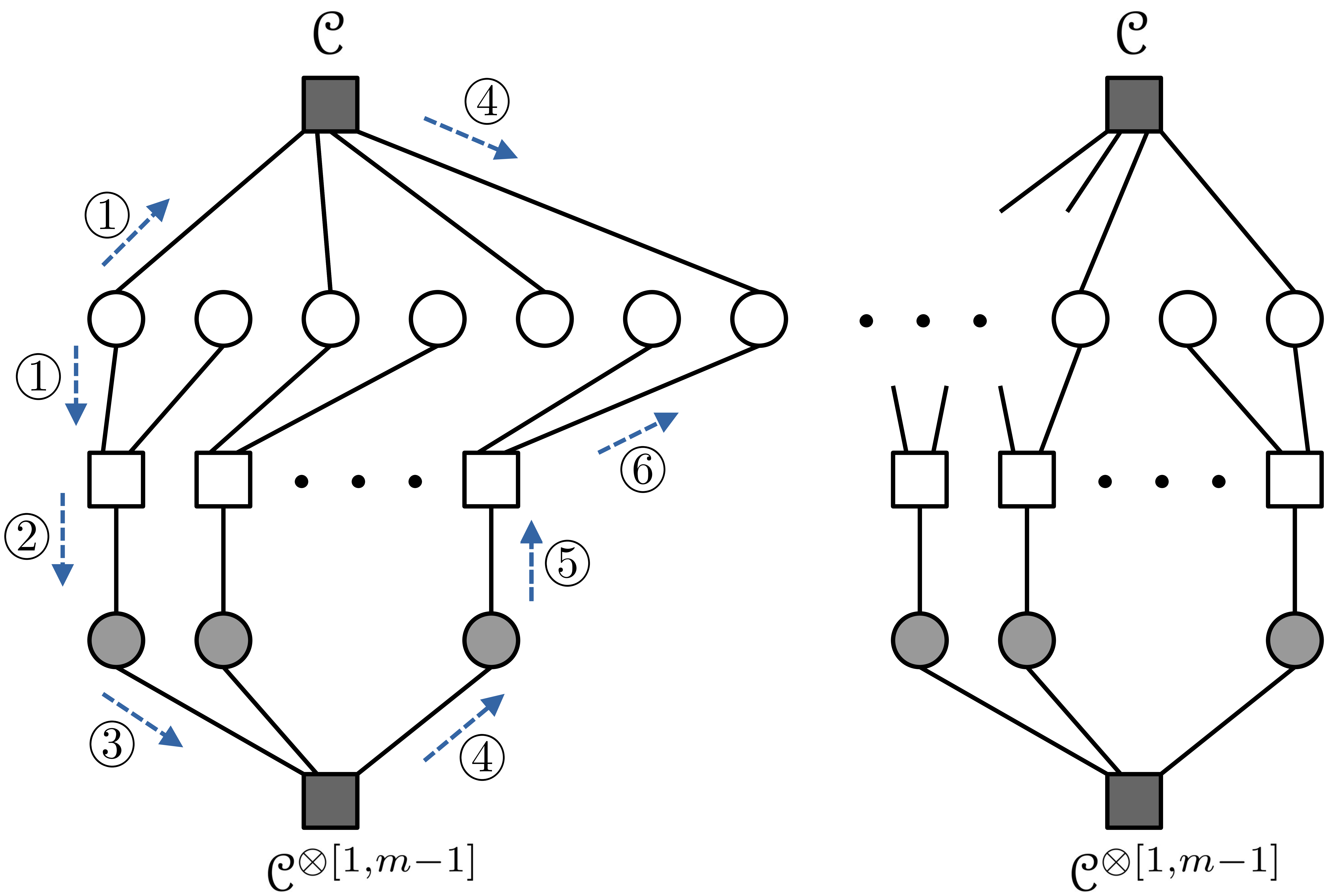
$$\mathscr{P}_{\mathscr{H}}(\mathbf{c}) + \mathscr{P}_{\mathscr{H}'}(\mathbf{c}) \in \mathscr{C}^{\otimes[r-1,m-1]} \quad \text{for every } \mathbf{c} \in \mathscr{C}^{\otimes[r,m]}$$

BELIEF PROPAGATION DECODER FOR SECOND-ORDER CODES

- Projections of second-order codewords yields first-order codewords
    - These can be ‘soft-in soft-out’ decoded efficiently
    - Can be used as generalised check nodes in a factor graph

- Since $\mathscr{C}^{\otimes[2,m]} \subset \mathscr{C}^{\otimes m}$
    - Every second-order codeword is also a codeword in the product code
    - When viewed as $n \times \cdots \times n$ array,
      length-n vectors along each dimension belong to the base code $\mathscr{C}$
    - These criteria can also be used as generalised check nodes

BELIEF PROPAGATION DECODER FOR SECOND-ORDER CODES



BP DECODER FOR SECOND-ORDER CODES

- We use all possible projections (to $\mathscr{C}^{\otimes[1,m-1]}$) and puncturing (to $\mathscr{C}$) in the factor graph

- Puncturings to $\mathscr{C}$ are useful only when $\mathscr{C}$ is non-trivial ($\mathscr{C} \neq \mathbb{F}_2^n$)

- Complexity order per BP iteration

$$\max\{N, N^\alpha\}\log N \ \text{ if } \ \alpha \neq 1 \quad \text{and} \quad N \log^2 N \ \text{ if } \ \alpha = 1$$

IMPROVING DECODER PERFORMANCE VIA LOCAL GRAPH SEARCH

LOCAL GRAPH SEARCH [KAMENEV, T-COM '22]

- Consider a graph $\mathcal{G}$ with
  - Vertex set: all codewords in $\mathcal{C}^{\otimes[r,m]}$
  - Edge set: two codewords are neighbours if their Hamming distance is $d^r n^{m-r}$
- Degree of each node is small: $\mathcal{O}\left(\log^r N\right)$ if $n \neq 2d$
- Starting from the output of BP decoder trace a path in $\mathcal{G}$ of some length, say P
  - At each step, pick the neighbor with the largest likelihood
- Among all P codewords visited in the path, choose the one with largest likelihood

LOCAL GRAPH SEARCH [KAMENEV, T-COM '22]

- Complexity order for second-order codes

$$P \log^2 N \max \left\{N,\ \log^2 N \log P\right\}$$

- If we use CRC for the recursive subproduct codes,
  we can ignore the codewords in the path that do not satisfy the CRC

MAIN REFERENCES

- G. Reeves and H. D. Pfister, "Reed–Muller Codes on BMS Channels Achieve Vanishing Bit-Error Probability for all Rates Below Capacity," in IEEE Transactions on Information Theory, vol. 70, no. 2, pp. 920-949, Feb. 2024, doi: 10.1109/TIT.2023.3286452.
- L. P. Natarajan and P. Krishnan, "Berman Codes: A Generalization of Reed–Muller Codes That Achieve BEC Capacity," in IEEE Transactions on Information Theory, vol. 69, no. 11, pp. 6956-6980, Nov. 2023, doi: 10.1109/TIT.2023.3299287.
- M. Lian, C. Häger and H. D. Pfister, "Decoding Reed–Muller Codes Using Redundant Code Constraints," 2020 IEEE International Symposium on Information Theory (ISIT), Los Angeles, CA, USA, 2020, pp. 42-47, doi: 10.1109/ISIT44484.2020.9174087.
- M. Kamenev, "On Decoding of Reed-Muller Codes Using a Local Graph Search," in IEEE Transactions on Communications, vol. 70, no. 2, pp. 739-748, Feb. 2022, doi: 10.1109/TCOMM.2021.3128541.
- A. Ashikhmin and S. Litsyn, "Simple MAP decoding of first-order Reed-Muller and Hamming codes," in IEEE Transactions on Information Theory, vol. 50, no. 8, pp. 1812-1818, Aug. 2004, doi: 10.1109/TIT.2004.831835.