Backtracking and Look-Ahead Decoding Algorithms for Improved Successive Cancellation Decoding Performance of Polar Codes

Myna Vajha, V. S. Chaitanya Mukka, and P. Vijay Kumar†

Department of Electrical Communication Engineering,
Indian Institute of Science, Bangalore

2019 IEEE International Symposium on Information Theory
Paris, France

MO1.R2: Polar and RM Codes
July 8, 2019

† also Visiting Professor, USC

Two Decoding Algorithms for Polar Codes

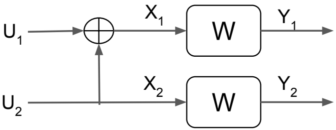
- Both the algorithms use $O(N)$ memory.

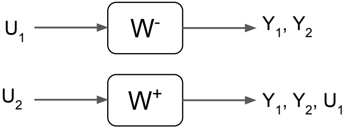
- Successive Cancellation Back-Tracking (SC-BT)
  - Performs close to ML for short block lengths $N = 32, 64$.
  - Improved performance over SCD for larger block lengths $N = 128, 256$.
  - Computation complexity varies with error patterns.
  - Small average complexity at large SNR and high complexity at small SNRs.

- Successive Cancellation Look-Ahead (SC-LA)
  - Tractable complexity. $O(\frac{2^D}{D} N \log N)$ where D is the parameter of the algorithm.
  - Improved performance over successive cancellation decoding.
  - Can be extended to list decoding.

Polarization of 2-channels (Arıkan 2008)

W is binary DMC

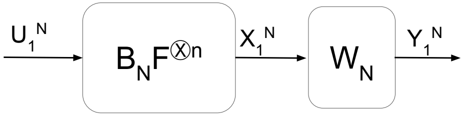




$$\begin{bmatrix} X_1 & X_2 \end{bmatrix} = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \underbrace{\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}}_{F}$$

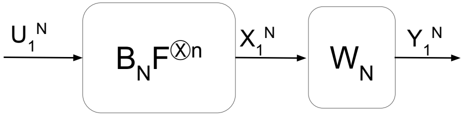
$$I(W^+) \geq I(W) \geq I(W^-)$$

Extension to Polarization of $N = 2^n$-channels (Arıkan 2008)

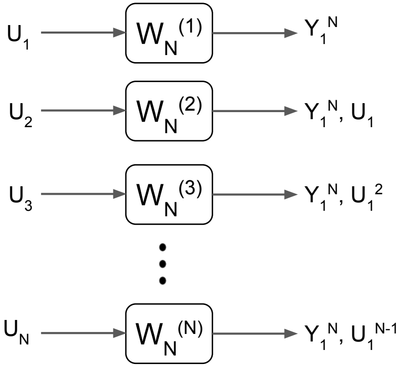


- B_N is a bit reversal permutation matrix.
- $F^{\otimes n}$ is n-time Kronecker product of F.
- W_N is N-independant uses of channel W.

Extension to Polarization of $N = 2^n$-channels (Arıkan 2008)

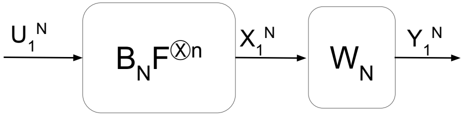


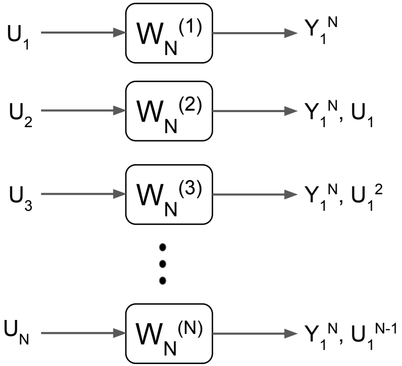
- B_N is a bit reversal permutation matrix.
- $F^{\otimes n}$ is n-time Kronecker product of F.
- W_N is N-independant uses of channel W.



N Bit Channels

Extension to Polarization of $N = 2^n$-channels (Arıkan 2008)





N Bit Channels

- B_N is a bit reversal permutation matrix.
- $F^{\otimes n}$ is n-time Kronecker product of F.
- W_N is N-independant uses of channel W.

- $\mathcal{F}$ is the set of frozen symbols, for any $i \in \mathcal{F}$, $U_i = 0$.
- $|\mathcal{F}^c| = K$, the dimension of the code.
- $\mathcal{F}$ carefully chosen such that $\sum_{i \in \mathcal{F}^c} P_e(W_N^{(i)}) \leq \epsilon_N$

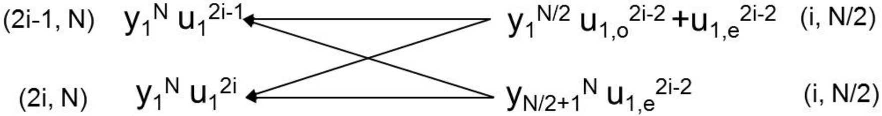
Successive Cancellation Decoding (SCD)

$$\hat{u}_i = \begin{cases} \arg\max_{u_i \in \{0,1\}} W_N^{(i)}(y_1^N, \hat{u}_1^{i-1} | u_i) & i \notin \mathcal{F} \\ 0 & i \in \mathcal{F} \end{cases}$$

Successive Cancellation Decoding (SCD)

$$\hat{u}_i = \begin{cases} \arg\max_{u_i \in \{0,1\}} W_N^{(i)}(y_1^N, \hat{u}_1^{i-1} | u_i) & i \notin \mathcal{F} \\ 0 & i \in \mathcal{F} \end{cases}$$

Recursive Metric Computation: For $i \in [\frac{N}{2}]$,

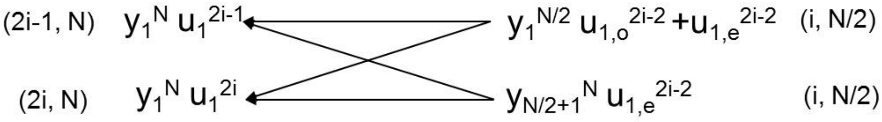


- Need two metrics from $\frac{N}{2}$ level with index i to compute metric at level N for indices $2i-1$ and $2i \implies O(N \log N)$ computation.

Successive Cancellation Decoding (SCD)

$$\hat{u}_i = \begin{cases} \arg\max_{u_i \in \{0,1\}} W_N^{(i)}(y_1^N, \hat{u}_1^{i-1} | u_i) & i \notin \mathcal{F} \\ 0 & i \in \mathcal{F} \end{cases}$$

Recursive Metric Computation: For $i \in [\frac{N}{2}]$,



- Need two metrics from $\frac{N}{2}$ level with index i to compute metric at level N for indices $2i-1$ and $2i \implies O(N \log N)$ computation.
- The recursive definition assumes that $U_1^N \sim \text{uniform}(\mathbb{F}_2^N)$. This is not true due to frozen symbols.

Impact of Uniformity Assumption (UA)

$$\underbrace{P(y_1^N u_1^{i-1} | u_i)}_{\text{UA metric}} = \sum_{u_{i+1}^N \in \mathbb{F}_2^{N-i}} 2P(y_1^N | u_1^N) P(u_1^N)$$

Impact of Uniformity Assumption (UA)

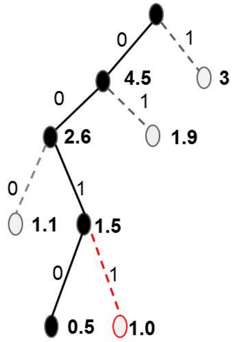
$$
\begin{aligned}
\underbrace{P(y_1^N u_1^{i-1} | u_i)}_{\text{UA metric}} &= \sum_{u_{i+1}^N \in \mathbb{F}_2^{N-i}} 2P(y_1^N | u_1^N) P(u_1^N) \\
&= \frac{1}{2^{N-1}} \sum_{u_{i+1}^N \in \mathbb{F}_2^{N-i}} P(y_1^N | u_1^N) \\
&= \underbrace{\frac{1}{2^{N-1}} \sum_{u_{i+1}^N \in \mathbb{F}_2^{N-i} : u_{i+1,\mathcal{F}}^N = 0} P(y_1^N | u_1^N)}_{\textit{actual}} + \text{noise}
\end{aligned}
$$

Decoding Algorithms as Search over Binary Tree

- Binary Tree of depth N with 2^N leaf nodes.
- Each node at depth i can be represented by u_1^i.
- Metric $M_i(u_1^i) = W_N^{(i)}(y_1^N, u_1^{i-1}|u_i)$

$$\begin{aligned} M_i(u_1^i) &= \frac{1}{2^{N-1}} \sum_{u_{i+1}^N \in \mathbb{F}_2^{N-i}} P(y_1^N | u_1^N) \\ &= M_{i+1}(u_1^i, 0) + M_{i+1}(u_1^i, 1) \quad \text{(Sum Metric)} \end{aligned}$$

- Metric at a node is sum of the child metrics.



$\hat{u}_1^4 = (0, 0, 1, 0)$

Decoding Algorithms: Search over Binary Tree

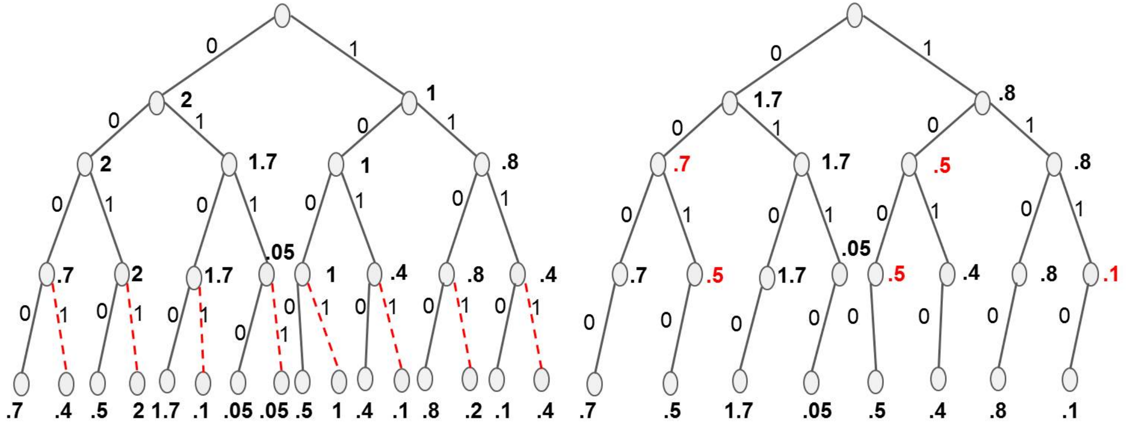
- Max Metric

$$M_i(u_1^i) \quad = \quad \max_{u_{i+1}^N \in \mathbb{F}_2^{N-i}} P(y_1^N | u_1^N) = \max\{M_{i+1}(u_1^i, 0), M_{i+1}(u_1^i, 1)\}$$

- At step i, we compute $M_i(\hat{u}_1^{i-1}, 0), M_i(\hat{u}_1^{i-1}, 1)$

$$\hat{u}_i = \arg \max_{u_i \in \{0,1\}} M_i(\hat{u}_1^{i-1}, u_i) \text{ for } i \notin \mathcal{F}$$

Decoding Algorithms: Search over Binary Tree

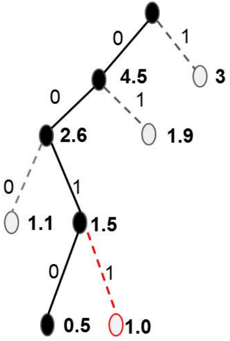


UA Metric vs Actual Metric (under Max Metric)

Decoding Algorithms: Search over Binary Tree

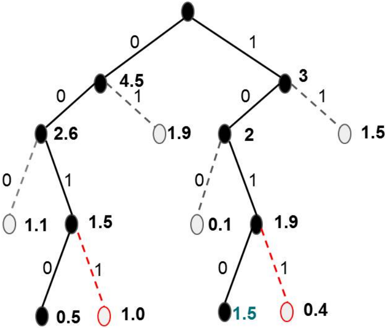
- 4-th symbol is assumed to be frozen

Successive Cancellation Decoding
(Arıkan, 2008)



$\hat{u}_1^4 = (0, 0, 1, 0)$

Succesive Cancellation List Decoding (SCLD)
(Tal & Vardy 2015)

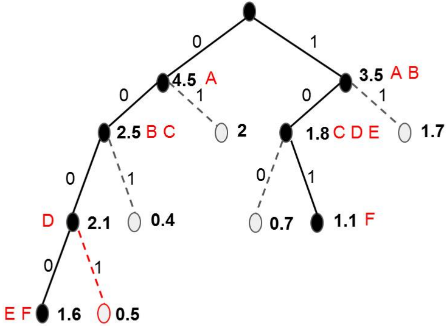


List size $L = 2$, $\hat{u}_1^4 = (1, 0, 1, 0)$

Output of SCLD is more likely (1.5) to be the message vector than the SCD output vector (0.5).

Decoding Algorithms: Search over Binary Tree

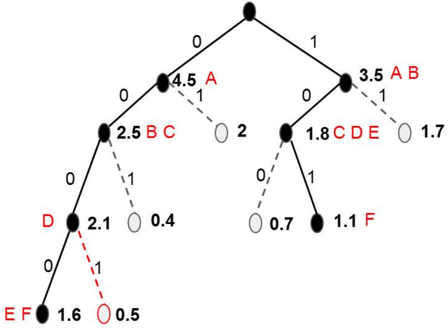
Stack Decoding
(Niu & Chen, 2012)



Stack size $D = 2$

Decoding Algorithms: Search over Binary Tree

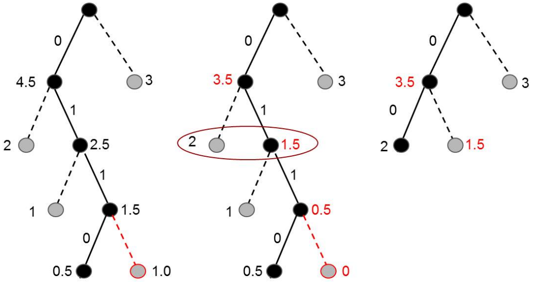
Stack Decoding
(Niu & Chen, 2012)



Stack size $D = 2$

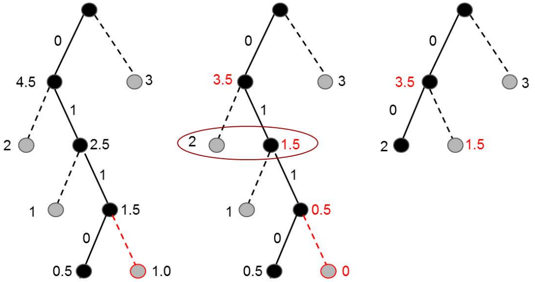
- SC Hybrid decoding (2013) by Chen, Liu, Lin that combines principles of stack decoding and SCLD.
- Sequential Decoding (2014) by Miloslavskaya & Trifonov and Score function (2018) by Trifonov introduce carefully designed bias functions to metrics.
- Partitioned SC List Decoding (2017) by Hashemi, Mondelli et. al. Trades memory for performance.

Successive Cancellation with Back-Tracking Algorithm (SC-BT)



- For $i \in \mathcal{F}$, set $M_i(\hat{u}_1^{i-1}, 1) = 0$.
- Backtrack the tree to update the metric at parent nodes. $M_{i-1}(\hat{u}_1^{i-1}) = M_i(\hat{u}_1^{i-1}, 0) + M_i(\hat{u}_1^{i-1}, 1)$
- Select new best path and continue the algorithm.
- Can select backtracking height D to be small in order to reduce computation at the cost of performance.

Successive Cancellation with Back-Tracking Algorithm (SC-BT)



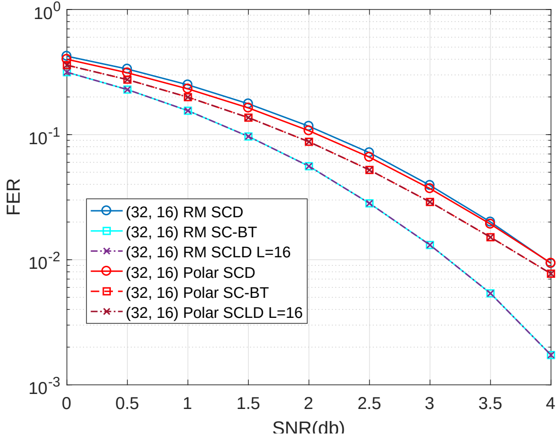
- For $i \in \mathcal{F}$, set $M_i(\hat{u}_1^{i-1}, 1) = 0$.
- Backtrack the tree to update the metric at parent nodes. $M_{i-1}(\hat{u}_1^{i-1}) = M_i(\hat{u}_1^{i-1}, 0) + M_i(\hat{u}_1^{i-1}, 1)$
- Select new best path and continue the algorithm.
- Can select backtracking height D to be small in order to reduce computation at the cost of performance.

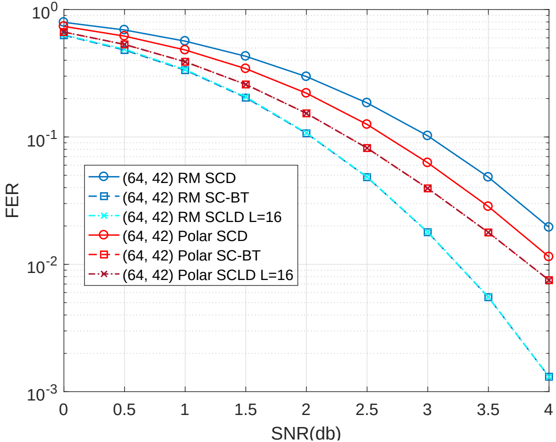
Complexity

	0 db		4 db	
D	max	mean	max	mean
4	105	$\simeq 71$	93	$\simeq 81$
8	1841	84	175	$\simeq 81$
10	5330	113	142	$\simeq 81$
16	10000	318	284	$\simeq 81$
32	10000	8496	322	$\simeq 81$

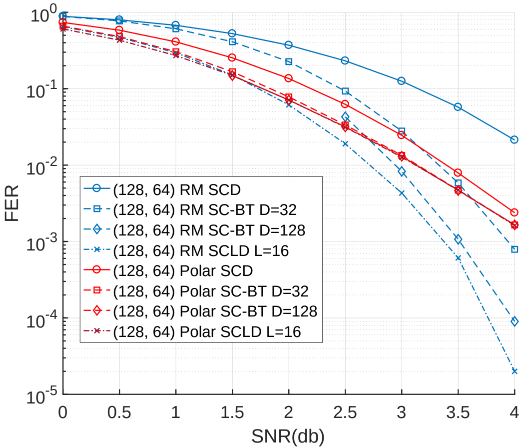
Table: Number of reflections observed for $(N = 256, K = 128)$, when backtracking is allowed to a height D.

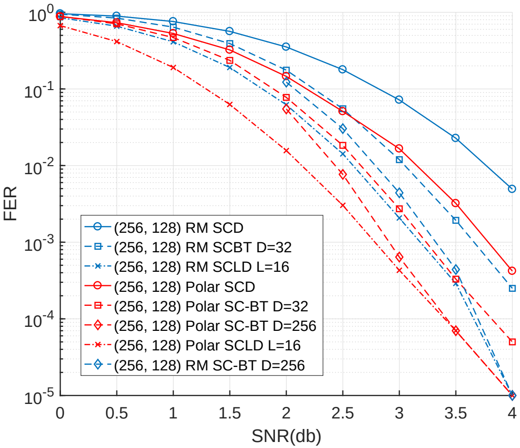
Evaluation of SC-BT Decoding



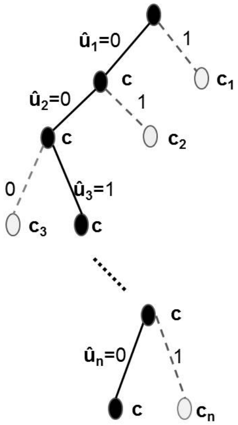


Evaluation of SC-BT Decoding





Successful SC-BT results in ML codeword (under Max Metric, $D = N$)



- It is clear that $c \geq c_i$ for all $i \in [N]$.
- Suppose $\exists \tilde{u}_1^N \neq \hat{u}_1^N$ such that

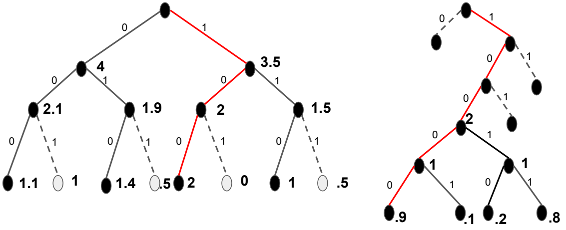
$$P(y_1^N|\tilde{u}_1^N) \quad > \quad P(y_1^N|\hat{u}_1^N)$$

- Let j be such that $\hat{u}_j \neq \tilde{u}_j$ and $\hat{u}_1^{j-1} = \tilde{u}_1^{j-1}$

$$\begin{aligned} c_j &\geq \max_{u_1^N \in \mathbb{F}_2^N : u_1^{j-1} = \hat{u}_1^{j-1}} P(y_1^N|u_1^N) \\ &\geq P(y_1^N|\tilde{u}_1^N) \\ &> P(y_1^N|\hat{u}_1^N) = c \quad \text{(contradiction)} \end{aligned}$$

- Can be extended to precoded polar codes with dynamic frozen symbols.
- Future Directions
  - Dynamics of this update process to be studied
  - Extension of the SC-BT algorithm to shortened, punctured polar codes

Successive Cancellation Look Ahead (SC-LA) Decoding



Look Ahead Depth $D = 2$.

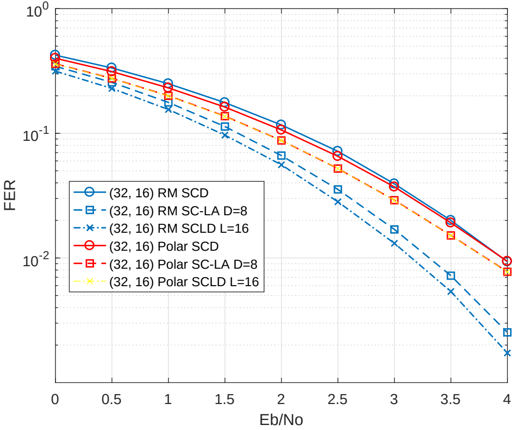
- When a frozen symbol is realized at depth D, the metrics computed at the 2^D leaf nodes is used to make decision on the D information bits.
- Can be extended to list decoding and dynamic frozen symbols.
- Complexity is tractable.

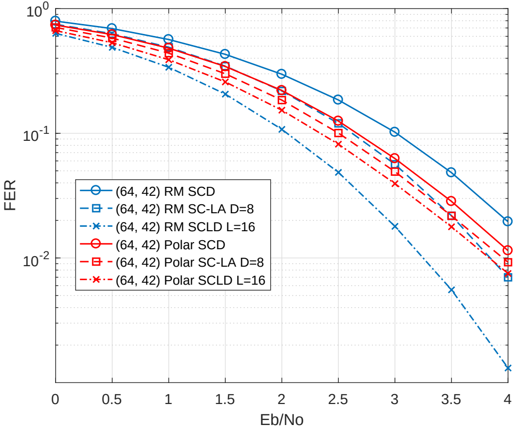
Complexity of traversing 2^D paths

$$
\begin{aligned}
C(i) &= \begin{cases} f(i) + 2C(i+1) & i \in \mathcal{F}^c \\ f(i) + C(i+1) & i \in F \end{cases} \\
C(j) &= \sum_{i=0}^{D-1} f(i+j)2^i + 2^D \sum_{i=D}^{D+f-1} f(i+j) \\
&\simeq O(2^D \log N)
\end{aligned}
$$

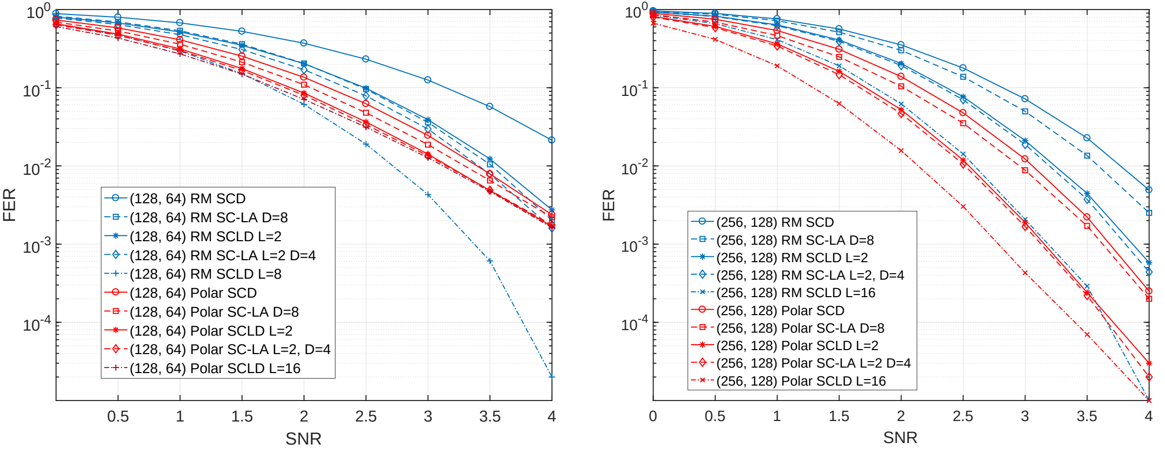
- Total complexity: $O(\frac{N}{D} 2^D \log N)$
- Future Work
  - Adapt the depth to the density of frozen bits

Evaluation of SC-LA Decoding





Evaluation of SC-LA Decoding



Thanks!