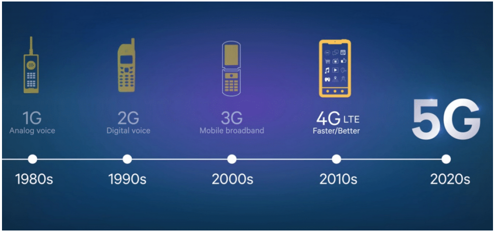
Deep-Learning-Aided Successive-Cancellation Decoding of Polar Codes

Seyyed Ali Hashemi[1], Nghia Doan[2], Thibaud Tonnellier[2], Warren Gross[2]

[1]Stanford University, USA
[2]McGill University, Canada

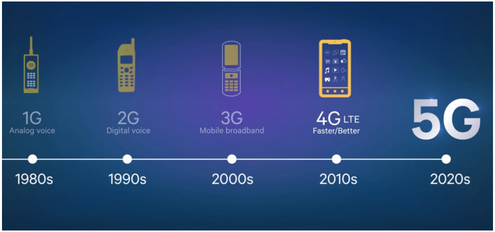
Asilomar Conference on Signals, Systems, and Computers
Pacific Grove, USA
November 5, 2019

Motivation



- Polar codes: selected for the eMBB control channel in 5G
- 5G has stringent requirements:
  - Low implementation complexity
- Successive-cancellation (SC) list (SCL) decoding:
  - Good error-correction performance for large list sizes
  - Complexity increases with list size

Motivation



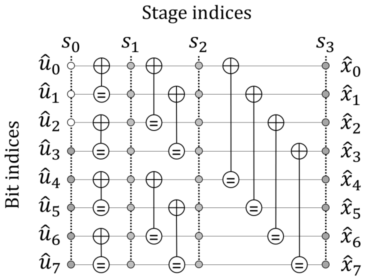
- Polar codes: selected for the eMBB control channel in 5G
- 5G has stringent requirements:
  - Low implementation complexity
- Dynamic SC flip (DSCF) decoding:
  - Comparable error-correction performance with SCL
  - Complexity close to SC at practical SNR
  - **Costly computations for a bit-flipping metric**

This talk

- A bit-flipping metric based on correlations between the bits
- A training framework to learn the correlations between the bits

Polar codes

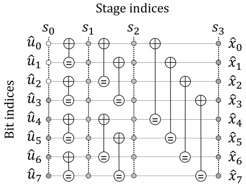
- $\mathcal{P}(N, K)$, N: code length, K: message length
- Code construction: based on polarization phenomenon
  - K most reliable channels: information bits $\mathcal{A}$
  - $N - K$ least reliable channels: frozen bits $\mathcal{A}^c$

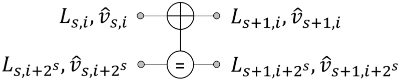


$\mathcal{P}(8, 5)$ with u_0, u_1, and u_2 as frozen bits

Arıkan'09.

SC Decoding





Right-to-left: soft values (LLRs)

$$L_{s,i} = f(L_{s+1,i}, L_{s+1,i+2^s})$$
$$L_{s,i+2^s} = g(L_{s+1,i}, L_{s+1,i+2^s}, \hat{\nu}_{s,i})$$

where

$$f(a,b) = \min(|a|,|b|)\,\mathrm{sgn}(a)\,\mathrm{sgn}(b)$$
$$g(a,b,c) = b + (1-2c)a,$$

Left-to-right: hard values (bits)

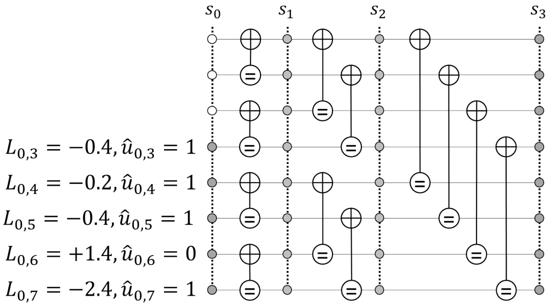
$$\hat{\nu}_{s+1,i} = \hat{\nu}_{s,i} \oplus \hat{\nu}_{s,i+2^s}$$
$$\hat{\nu}_{s+1,i+2^s} = \hat{\nu}_{s,i+2^s}$$

where

$$\hat{u}_i = \hat{\nu}_{0,i} = \begin{cases} 0 & \text{if } u_i \text{ is frozen,} \\ \frac{1-\mathrm{sgn}(L_{0,i})}{2} & \text{otherwise.} \end{cases}$$

Successive Cancellation Flip (SCF) Decoding

Example: all-zero codeword

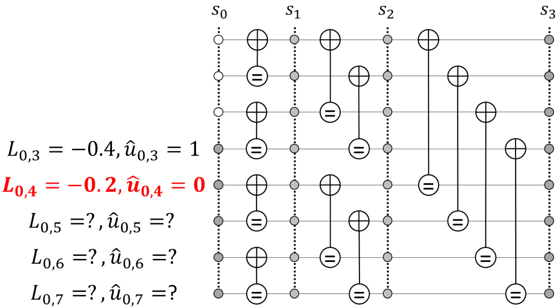


- Flipping position: $\underset{\forall i \in \mathcal{A}}{\arg\min} \left| L_{0,i} \right|$

Afisiadis'14

Successive Cancellation Flip (SCF) Decoding

Example: all-zero codeword



- Flipping position: $\arg\min\limits_{\forall i \in \mathcal{A}} |L_{0,i}| = 4$

Afisiadis'14

Dynamic SC-Flip (DSCF) Decoding

- Introduced a bit-flipping metric for high-order errors
- Bit-flipping probability at the i-th information bit:
  - $\mathcal{E}_\omega$: set of bit-flipping position at error order ω
  - $p_i^* = Pr(\hat{u}_i = u_i | \hat{\boldsymbol{u}}_0^{i-1} = \boldsymbol{u}_0^{i-1}) \approx \frac{1}{1+\exp(-\alpha|L_{0,i}|)}$
  - $\alpha > 0$ is a perturbation parameter

$$Pr(\mathrm{BF}_i) = \prod_{\forall j \in \mathcal{E}_\omega,\, j \leq i} \left(1 - p_j^*\right) \prod_{\forall j \in \mathcal{A} \setminus \mathcal{E}_\omega,\, j < i} p_j^*$$

- Flipping position: $\arg\max_{\forall i \in \mathcal{A}} Pr(\mathrm{BF}_i)$

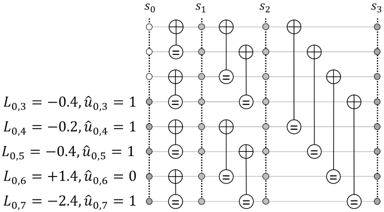
Chandesris'18

Dynamic SC-Flip (DSCF) Decoding

Example: all-zero codeword, $p_i^* = \frac{1}{1+\exp(-\alpha|L_{0,i}|)}$, $\alpha = 0.3$

- $\omega = 1$:

i	p_i^*	$1 - p_i^*$	$Pr(\mathrm{BF}_i)$
3	0.53	0.47	-
4	0.51	0.49	-
5	0.53	0.47	-
6	0.60	0.40	-
7	0.67	0.33	-



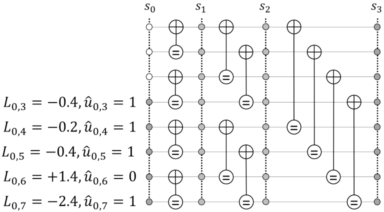
-

Dynamic SC-Flip (DSCF) Decoding

Example: all-zero codeword, $p_i^* = \frac{1}{1+\exp(-\alpha|L_{0,i}|)}$, $\alpha = 0.3$

- $\omega = 1$:

i	p_i^*	$1-p_i^*$	$Pr(\mathrm{BF}_i)$
3	0.53	0.47	0.47
4	0.51	0.49	-
5	0.53	0.47	-
6	0.60	0.40	-
7	0.67	0.33	-



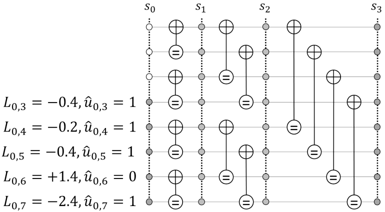
- $Pr(\mathrm{BF}_3) = 1 - p_3^*$

Dynamic SC-Flip (DSCF) Decoding

Example: all-zero codeword, $p_i^* = \frac{1}{1+\exp(-\alpha|L_{0,i}|)}$, $\alpha = 0.3$

- $\omega = 1$:

i	p_i^*	$1 - p_i^*$	$Pr(\mathrm{BF}_i)$
3	0.53	0.47	0.47
4	0.51	0.49	0.26
5	0.53	0.47	-
6	0.60	0.40	-
7	0.67	0.33	-



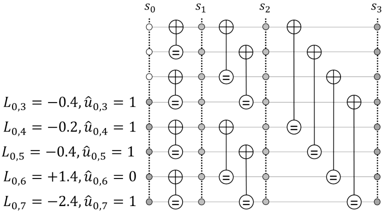
- $Pr(\mathrm{BF}_4) = p_3^* \times (1 - p_4^*)$

Dynamic SC-Flip (DSCF) Decoding

Example: all-zero codeword, $p_i^* = \frac{1}{1+\exp(-\alpha|L_{0,i}|)}$, $\alpha = 0.3$

- $\omega = 1$:

i	p_i^*	$1 - p_i^*$	$Pr(\mathrm{BF}_i)$
3	0.53	0.47	0.47
4	0.51	0.49	0.26
5	0.53	0.47	0.13
6	0.60	0.40	-
7	0.67	0.33	-



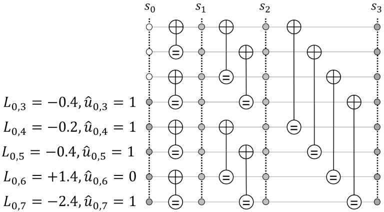
- $Pr(\mathrm{BF}_5) = p_3^* \times p_4^* \times (1 - p_5^*)$

Dynamic SC-Flip (DSCF) Decoding

Example: all-zero codeword, $p_i^* = \frac{1}{1+\exp(-\alpha|L_{0,i}|)}$, $\alpha = 0.3$

- $\omega = 1$:

i	p_i^*	$1 - p_i^*$	$Pr(\text{BF}_i)$
3	0.53	0.47	0.47
4	0.51	0.49	0.26
5	0.53	0.47	0.13
6	0.60	0.40	0.06
7	0.67	0.33	-



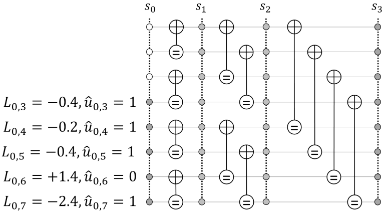
- $Pr(\text{BF}_6) = p_3^* \times p_4^* \times p_5^* \times (1 - p_6^*)$

Dynamic SC-Flip (DSCF) Decoding

Example: all-zero codeword, $p_i^* = \frac{1}{1+\exp(-\alpha|L_{0,i}|)}$, $\alpha = 0.3$

- $\omega = 1$:

i	p_i^*	$1 - p_i^*$	$Pr(\text{BF}_i)$
3	0.53	0.47	0.47
4	0.51	0.49	0.26
5	0.53	0.47	0.13
6	0.60	0.40	0.06
7	0.67	0.33	0.03



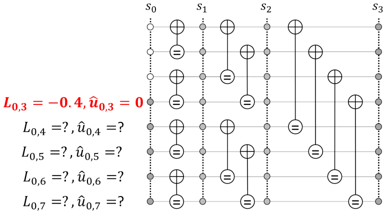
- $Pr(\text{BF}_7) = p_3^* \times p_4^* \times p_5^* \times p_6^* \times (1 - p_7^*)$

Dynamic SC-Flip (DSCF) Decoding

Example: all-zero codeword, $p_i^* = \frac{1}{1+\exp(-\alpha|L_{0,i}|)}$, $\alpha = 0.3$

- $\omega = 1$:

i	p_i^*	$1 - p_i^*$	$Pr(\mathrm{BF}_i)$
3	0.53	0.47	**0.47**
4	0.51	0.49	0.26
5	0.53	0.47	0.13
6	0.60	0.40	0.06
7	0.67	0.33	0.03



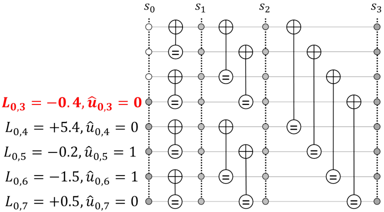
- $\arg\max_{\forall i \in \mathcal{A}} Pr(\mathrm{BF}_i) = 3$

Dynamic SC-Flip (DSCF) Decoding

Example: all-zero codeword, $p_i^* = \frac{1}{1+\exp(-\alpha|L_{0,i}|)}$, $\alpha = 0.3$

- $\omega = 2$:

i	p_i^*	$1 - p_i^*$	$Pr(\mathrm{BF}_i)$
3	0.53	0.47	**0.47**
4	0.83	0.17	-
5	0.51	0.49	-
6	0.61	0.39	-
7	0.46	0.54	-



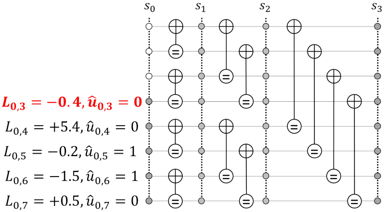
-

Dynamic SC-Flip (DSCF) Decoding

Example: all-zero codeword, $p_i^* = \frac{1}{1+\exp(-\alpha|L_{0,i}|)}$, $\alpha = 0.3$

- $\omega = 2$:

i	p_i^*	$1-p_i^*$	$Pr(\text{BF}_i)$
3	0.53	0.47	**0.47**
4	0.83	0.17	0.08
5	0.51	0.49	-
6	0.61	0.39	-
7	0.46	0.54	-



- $Pr(\text{BF}_4) = (1-p_3^*) \times (1-p_4^*)$

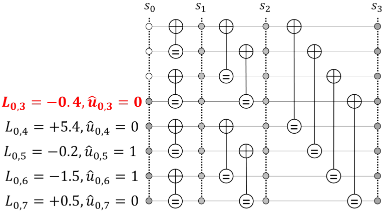
Dynamic SC-Flip (DSCF) Decoding

Example: all-zero codeword, $p_i^* = \frac{1}{1+\exp(-\alpha|L_{0,i}|)}$, $\alpha = 0.3$

- $\omega = 2$:

i	p_i^*	$1 - p_i^*$	$Pr(\mathrm{BF}_i)$
3	0.53	0.47	**0.47**
4	0.83	0.17	0.08
5	0.51	0.49	0.19
6	0.61	0.39	-
7	0.46	0.54	-

- $Pr(\mathrm{BF}_5) = (1 - p_3^*) \times p_4^* \times (1 - p_5^*)$

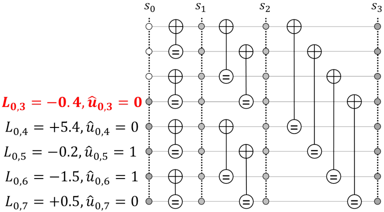


Dynamic SC-Flip (DSCF) Decoding

Example: all-zero codeword, $p_i^* = \frac{1}{1+\exp(-\alpha|L_{0,i}|)}$, $\alpha = 0.3$

- $\omega = 2$:

i	p_i^*	$1 - p_i^*$	$Pr(\mathrm{BF}_i)$
3	0.53	0.47	**0.47**
4	0.83	0.17	0.08
5	0.51	0.49	0.19
6	0.61	0.39	0.07
7	0.46	0.54	-



- $Pr(\mathrm{BF}_6) = (1 - p_3^*) \times p_4^* \times p_5^* \times (1 - p_6^*)$

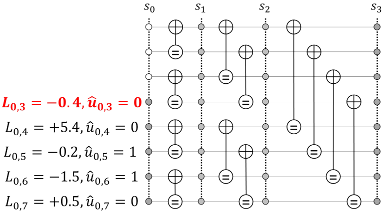
Dynamic SC-Flip (DSCF) Decoding

Example: all-zero codeword, $p_i^* = \frac{1}{1+\exp(-\alpha|L_{0,i}|)}$, $\alpha = 0.3$

- $\omega = 2$:

i	p_i^*	$1 - p_i^*$	$Pr(\mathrm{BF}_i)$
3	0.53	0.47	**0.47**
4	0.83	0.17	0.08
5	0.51	0.49	0.19
6	0.61	0.39	0.07
7	0.46	0.54	0.06

- $Pr(\mathrm{BF}_7) = (1 - p_3^*) \times p_4^* \times p_5^* \times p_6^* \times (1 - p_7^*)$



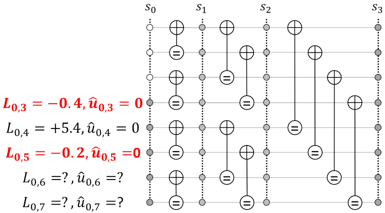
Dynamic SC-Flip (DSCF) Decoding

Example: all-zero codeword, $p_i^* = \frac{1}{1+\exp(-\alpha|L_{0,i}|)}$, $\alpha = 0.3$

- $\omega = 2$:

i	p_i^*	$1 - p_i^*$	$Pr(\text{BF}_i)$
3	0.53	0.47	**0.47**
4	0.83	0.17	0.08
5	0.51	0.49	**0.19**
6	0.61	0.39	0.07
7	0.46	0.54	0.06

- $\arg\max_{\forall i \in \mathcal{A}} Pr(\text{BF}_i) = 5$



DSCF Decoding Issues

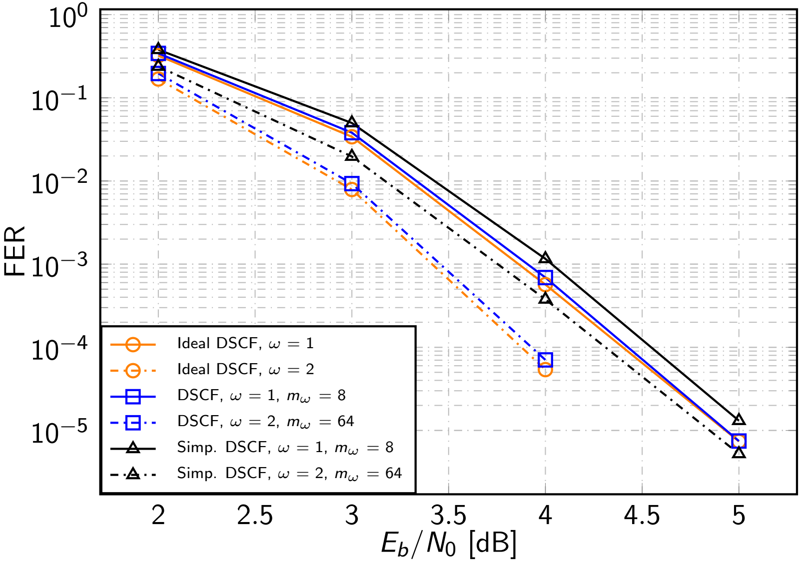
- Numerical stability: $Q_i = -\frac{1}{\alpha}\ln(Pr(\mathrm{BF}_i))$

$$Q_i = \sum_{\substack{\forall j \in \mathcal{A} \\ j \leq i}} \frac{1}{\alpha} \ln\left(1 + \exp\left(-\alpha |L_{0,j}|\right)\right) + \sum_{\substack{\forall j \in \mathcal{E}_\omega \\ j \leq i}} |L_{0,j}|$$

- **Computing Q_i requires costly** ln **and** exp **functions!**
- Simplifying Q_i: $\ln(1 + \exp(x)) \approx \mathrm{ReLU}(x)$

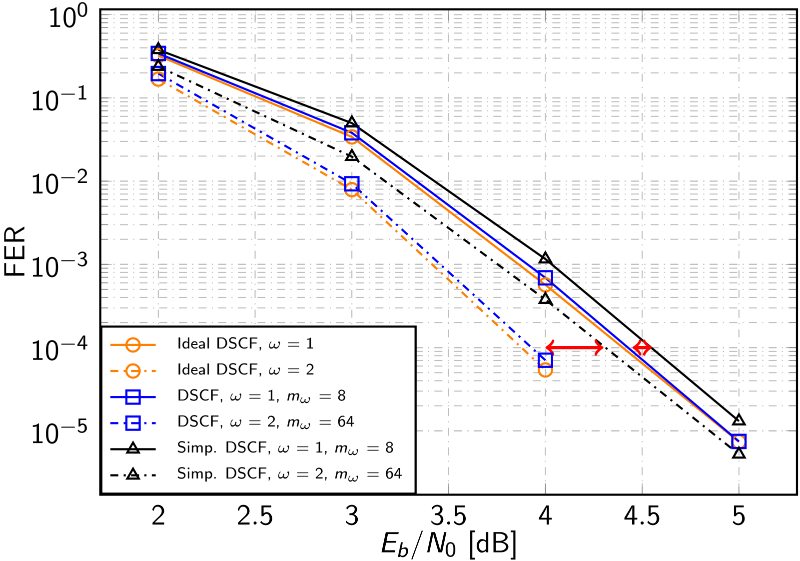
$$\tilde{Q}_i = \sum_{\substack{\forall j \in \mathcal{A} \\ j \leq i}} \frac{1}{\alpha} \mathrm{ReLU}\left(-\alpha |L_{0,j}|\right) + \sum_{\substack{\forall j \in \mathcal{E}_\omega \\ j \leq i}} |L_{0,j}| = \sum_{\substack{\forall j \in \mathcal{E}_\omega \\ j \leq i}} |L_{0,j}|$$

DSCF Decoding Issues



$\mathcal{P}(256, 128) - C24$ used in 5G.

DSCF Decoding Issues



$\mathcal{P}(256, 128) - C24$ used in 5G.

Proposed Method

- Use the likelihood ratios directly:

$$l^*_{i_\omega} = \max\left\{ \frac{\Pr(\hat{u}_{i_\omega} = 0 | \boldsymbol{y}, \boldsymbol{u})}{\Pr(\hat{u}_{i_\omega} = 1 | \boldsymbol{y}, \boldsymbol{u})}, \frac{\Pr(\hat{u}_{i_\omega} = 1 | \boldsymbol{y}, \boldsymbol{u})}{\Pr(\hat{u}_{i_\omega} = 0 | \boldsymbol{y}, \boldsymbol{u})} \right\}$$

  - Indicates how likely $\hat{u}_{i_\omega}$ is decoded correctly given $\boldsymbol{y}$ and $\boldsymbol{u}$
- First-order erroneous bit is

$$i^*_\omega = \underset{\forall i_\omega \in \mathcal{A}}{\arg\min}\, l^*_{i_\omega}.$$

Proposed Method

- Use the likelihood ratios directly:

$$l^*_{i_\omega} = \max\left\{ \frac{\Pr(\hat{u}_{i_\omega} = 0|\boldsymbol{y}, \boldsymbol{u})}{\Pr(\hat{u}_{i_\omega} = 1|\boldsymbol{y}, \boldsymbol{u})}, \frac{\Pr(\hat{u}_{i_\omega} = 1|\boldsymbol{y}, \boldsymbol{u})}{\Pr(\hat{u}_{i_\omega} = 0|\boldsymbol{y}, \boldsymbol{u})} \right\}$$

  - Indicates how likely $\hat{u}_{i_\omega}$ is decoded correctly given $\boldsymbol{y}$ and $\boldsymbol{u}$
- First-order erroneous bit is

$$i^*_\omega = \arg\min_{\forall i_\omega \in \mathcal{A}} l^*_{i_\omega}.$$

- But we don't have $\boldsymbol{u}$!

Proposed Method

- We need to estimate $l^*_{i_\omega}$
- Let's model it as a function of what we have:

$$l^*_{i_\omega} \approx \prod_{\forall i \in \mathcal{A}} l_i^{\beta_{i_\omega,i}}$$

where

$$\begin{aligned} l_i &= \max\left\{ \frac{\Pr(\hat{u}_i = 0 | \boldsymbol{y}, \hat{\boldsymbol{u}}_0^{i-1})}{\Pr(\hat{u}_i = 1 | \boldsymbol{y}, \hat{\boldsymbol{u}}_0^{i-1})}, \frac{\Pr(\hat{u}_i = 1 | \boldsymbol{y}, \hat{\boldsymbol{u}}_0^{i-1})}{\Pr(\hat{u}_i = 0 | \boldsymbol{y}, \hat{\boldsymbol{u}}_0^{i-1})} \right\} \\ &= \exp\left(|L_{0,i}|\right) \end{aligned}$$

and $\beta_{i_\omega,i} \in \mathbb{R}$ are perturbation parameters:

- $\beta_{i_\omega,i} = \beta_{i,i_\omega}$
- $\beta_{i_\omega,i_\omega} = 1$

Proposed Method

- Numerical stability:

$$Q_{i_\omega} = \ln(l^*_{i_\omega}) \approx \ln\left(\prod_{\forall i \in \mathcal{A}} \exp\left(\beta_{i_\omega,i}|L_{0,i}|\right)\right)$$
$$= \sum_{\forall i \in \mathcal{A}} \beta_{i_\omega,i}|L_{0,i}|$$

- In matrix form:

$$\boldsymbol{Q} = |\boldsymbol{L}_0| \cdot \boldsymbol{\beta}$$

- $i^*_\omega = \underset{\forall i_\omega \in \mathcal{A}}{\arg\min}\, Q_{i_\omega}$

Proposed Method

- Numerical stability:

$$Q_{i_\omega} = \ln(l^*_{i_\omega}) \approx \ln\left(\prod_{\forall i \in \mathcal{A}} \exp\left(\beta_{i_\omega,i}|L_{0,i}|\right)\right)$$
$$= \sum_{\forall i \in \mathcal{A}} \beta_{i_\omega,i}|L_{0,i}|$$

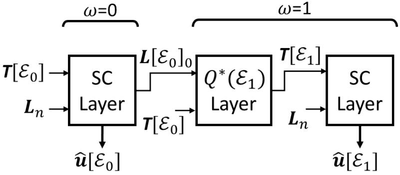
- In matrix form:

$$\boldsymbol{Q} = |\boldsymbol{L}_0| \cdot \boldsymbol{\beta}$$

- $i^*_\omega = \underset{\forall i_\omega \in \mathcal{A}}{\arg\min}\, Q_{i_\omega}$

We call $\boldsymbol{\beta}$ the correlation matrix!

Designing β

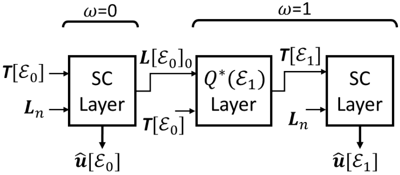


- Deep learning:
  - Each element in β is a trainable parameter
  - Stochastic gradient-descent (SGD) is used to train β
  - All-zero codeword dataset
- The goal is to have a bit-flipping vector $\hat{\boldsymbol{T}}$:

$$\hat{T}_i = \begin{cases} -1 & \text{if } i = i^*_\omega \\ +1 & \text{if } i \neq i^*_\omega \end{cases}$$

  - Not differentiable with respect to Q_{i_ω}!

Designing β



- We use a soft estimate for $\hat{T}_i$

$$\tilde{T}_i = \tanh(Q_i - \tau)$$

  τ is the average of the first and the second minima of Q_i
  - Only one bit is flipped!
- Loss function:

$$\frac{1}{K} \sum_{i=0}^{K-1} \underbrace{\mathcal{L}\left(\frac{1-\tilde{T}_i}{2}, \frac{1-T_i}{2}\right)}_{\text{Binary Cross-Entropy}} + \lambda \underbrace{\sum_{i_\omega=0}^{K-1} \sum_{i=i_\omega+1}^{K-1} (\beta_{i_\omega,i})^2}_{\text{L2 Regularization}}$$

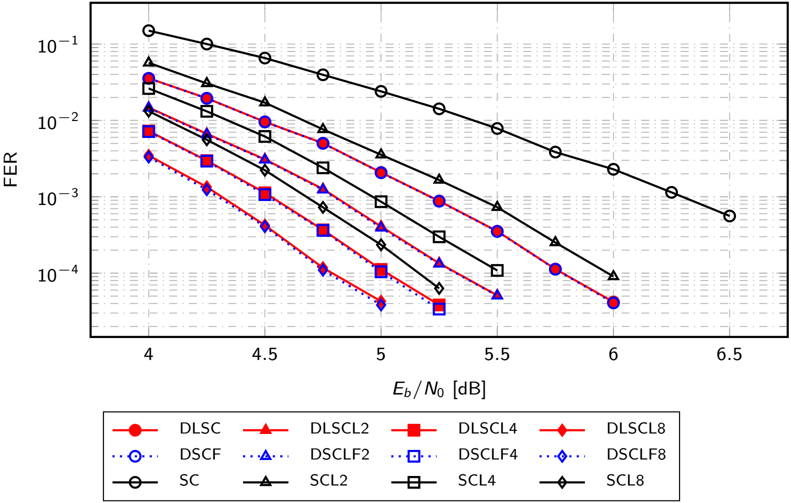
Training Setup

minibatch size	128
learning rate	10^{-4}
dataset size	2^{18}
E_b/N_0	5 dB
λ	0.25

All the elements in $\boldsymbol{\beta}$ that are in $[-10^{-4}, 10^{-4}]$ are set to zero!

Results

$\mathcal{P}(128, 64)$ with CRC of length 24 used in 5G.

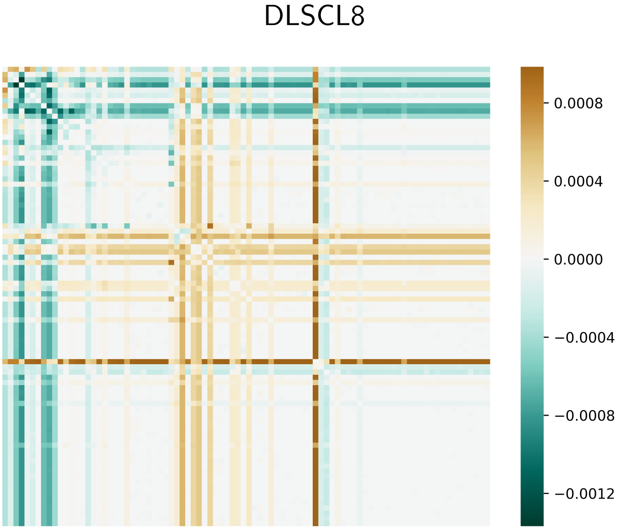


Results

Number of operations for $\mathcal{P}(128, 64)$

Decoders	$\times$	$+$	ln/exp
DSCF	7832	4004	7832
DLSC	2652 (66% $\downarrow$)	2564 (36% $\downarrow$)	0
DLSCL2	3116 (60% $\downarrow$)	3028 (24% $\downarrow$)	0
DLSCL4	3238 (59% $\downarrow$)	3150 (21% $\downarrow$)	0
DLSCL8	3176 (59% $\downarrow$)	3088 (23% $\downarrow$)	0

How Does β Look Like?



Conclusion

- A new bit-flipping metric is proposed for DSCF decoding
  - Based on a *correlation matrix*
  - No computationally expensive transcendental functions
- A training framework is introduced to design the correlation matrix
- Compared to DSCF decoding:
  - Almost no error-correction performance loss
  - Up to 66% and 36% savings in the number of $\times$'s and $+$'s respectively
- Future works:
  - Generalize it for all code rates (and lengths)
  - Use the correlation matrix for belief propagation decoding

Thank You!