Deep Neural Network Based Decoding of Short LDPC Codes

Information and coding theory

Skolkovo Institute of Science and Technology (Skoltech)

Skoltech
Skolkovo Institute of Science and Technology

Motivation

The main goal is to suggest an application of deep learning to channel decoding in order to improve performance or reduce complexity/latency.

State-of-the-art

There were already attempts to construct neural network (NN) learning-based decoders in literature, here we face with so-called curse of dimensionality problem: even for a short code of length N = 100 bits and rate R = 0.5, 2^{50} different codewords exist, which are far too many to fully train any NN in practice.

Possible suggestions

In our opinion, the only way deal with the problem is to combine deep learning methods with existing decoding methods.

Learning methods to investigate

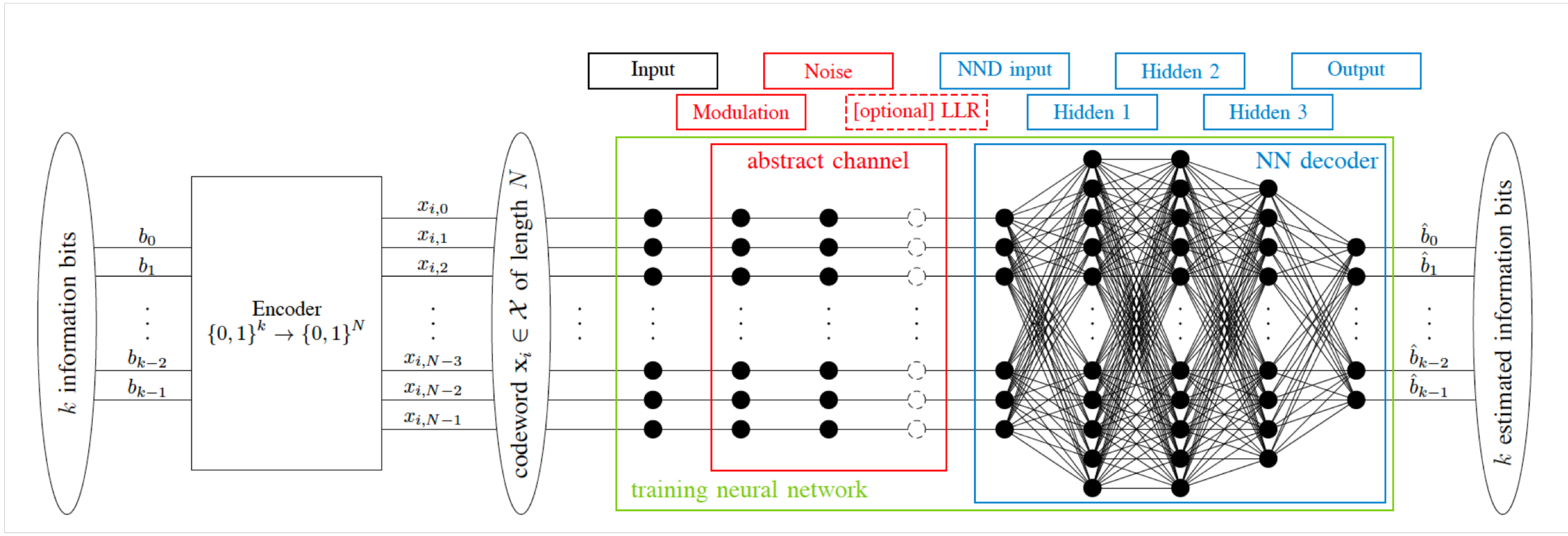
- **Method 1.** General purpose NN

  Tobias Gruber, Sebastian Cammerer, Jakob Hoydis, and Stephan ten Brink, On Deep Learning-Based Channel Decoding // arXiv:1701.07738

- **Method 2.** NN with special structure

  Eliya Nachmani, Elad Marciano, Loren Lugosch, Warren J. Gross, David Burshtein, Senior and Yair Be'ery, Deep Learning Methods for Improved Decoding of Linear Codes // arXiv:1706.07043

Method 1. NN architecture



On Deep Learning-Based Channel Decoding // arXiv:1701.07738

Toy example

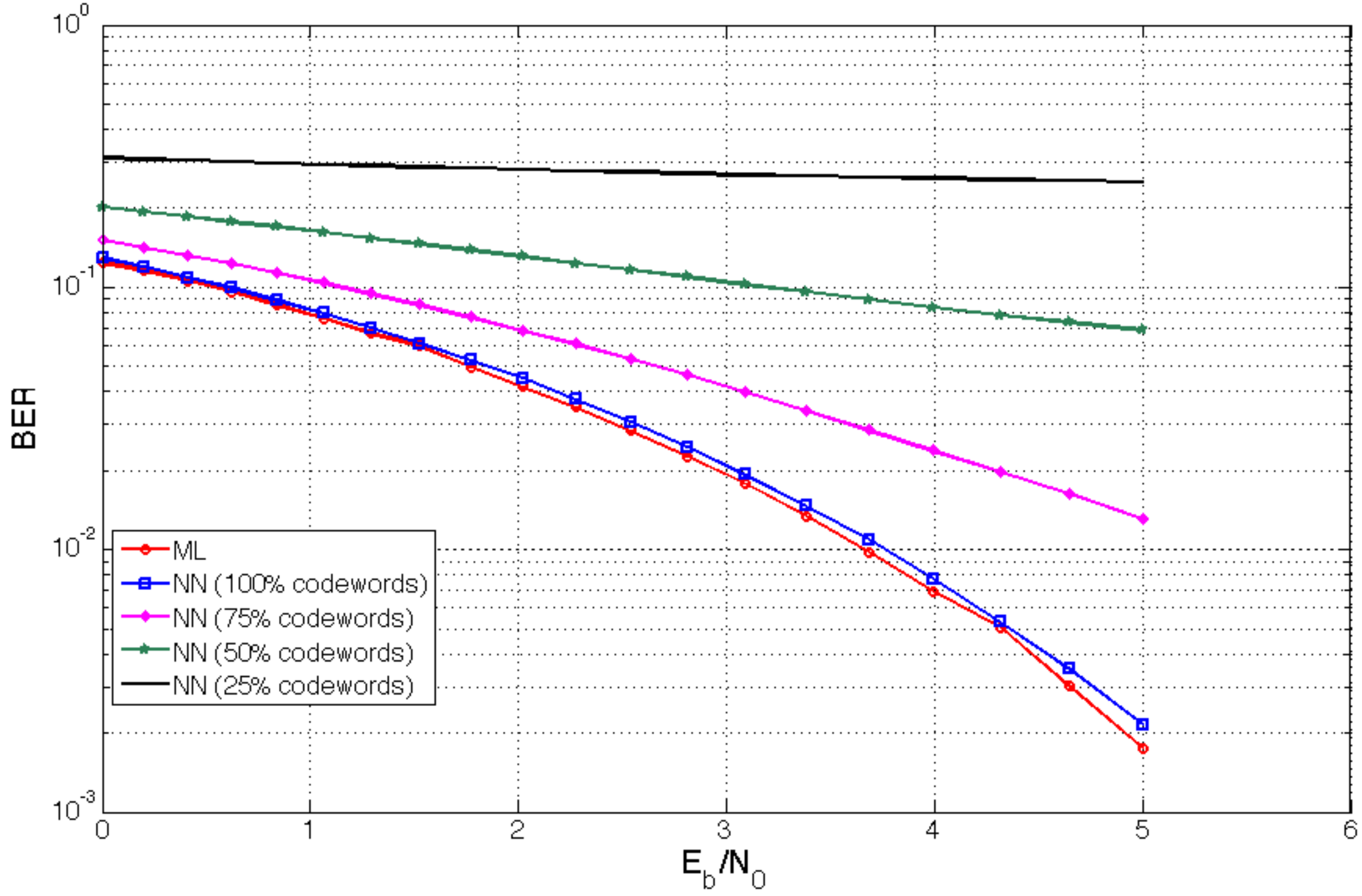
- [16, 8, 4] binary Reed-Muller subcode;
- To increase the minimum distance we choose the rows with the largest weight from the matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}^{\otimes 4}$$

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

Skoltech
Skolkovo Institute of Science and Technology

Method 1. Simulation results



Skoltech
Skolkovo Institute of Science and Technology

Method 2. Sum-Product/Min-Sum decoders

- Messages are log-likelihood ratios:

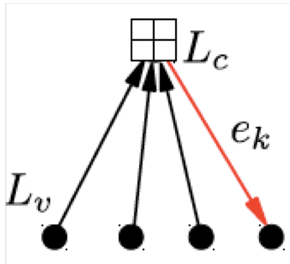
$$L_{\text{ch}} = \log \frac{\Pr(r|v=0)}{\Pr(r|v=1)}$$

- Check node update
  - Sum-Product

$$L_{e_k} = 2\, atanh \left(\prod_{k \neq k'} tanh \left(\frac{L_v(e_{k'})}{2} \right) \right)$$

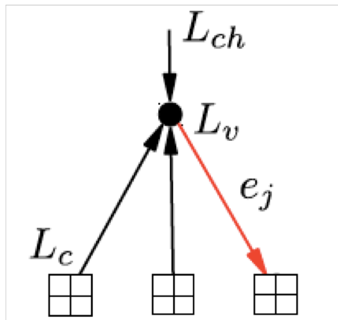
  - Min-Sum

$$L_{e_k} = \prod_{k \neq k'} sign\left(L_v(e_{k'})\right) \min_{k \neq k'} |L_v(e_{k'})|$$



- Variable node update

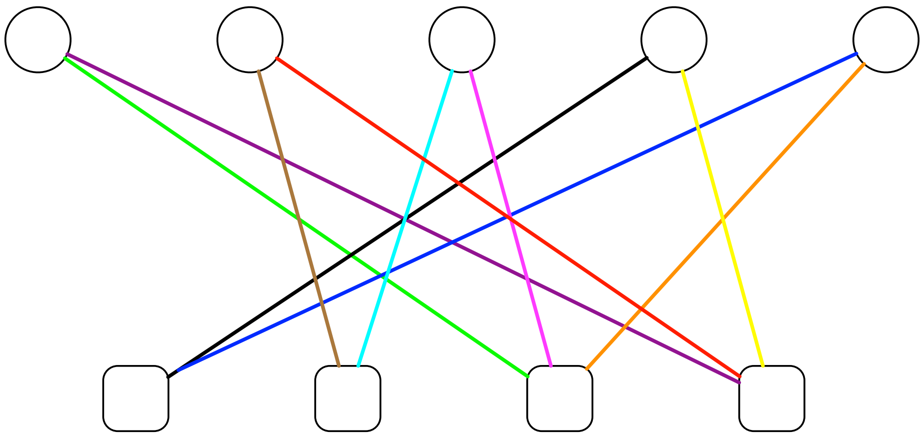
$$L_v(e_j) = L_{\text{ch}} + \sum_{j' \neq j} L_c(e_{j'})$$



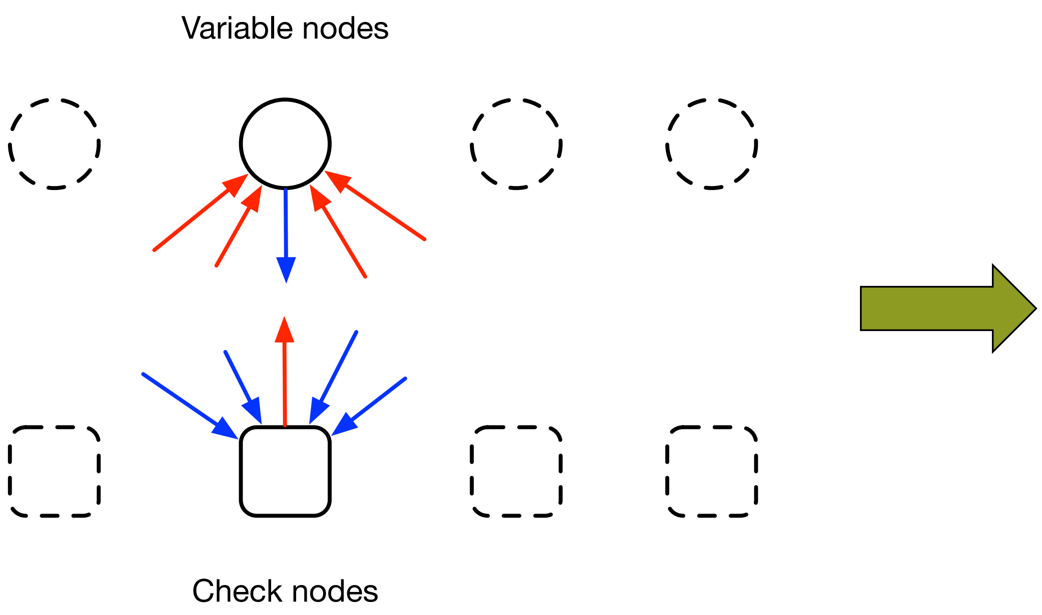
Skoltech
Skolkovo Institute of Science and Technology

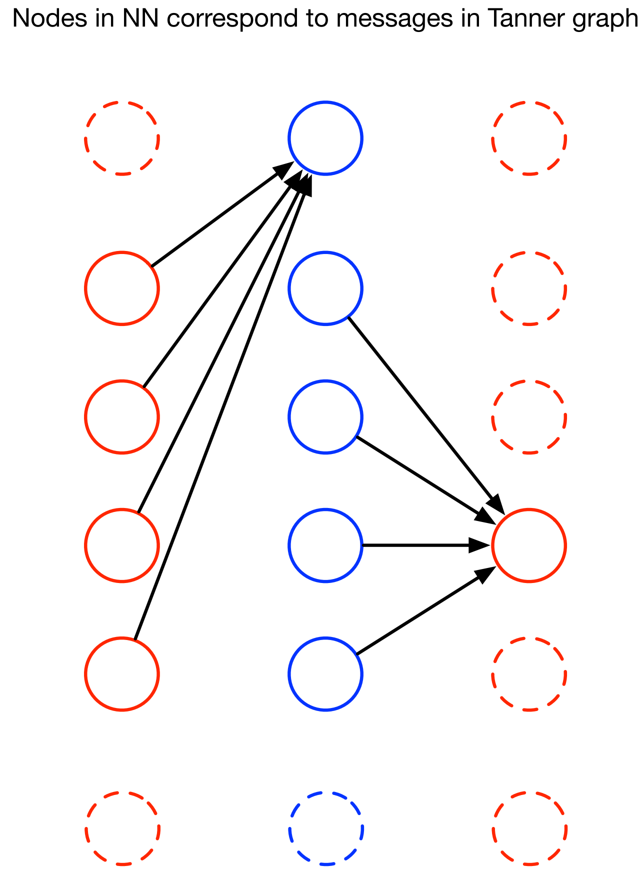
Method 2. Tanner graph

$$\mathbf{H} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix} \Longleftrightarrow$$



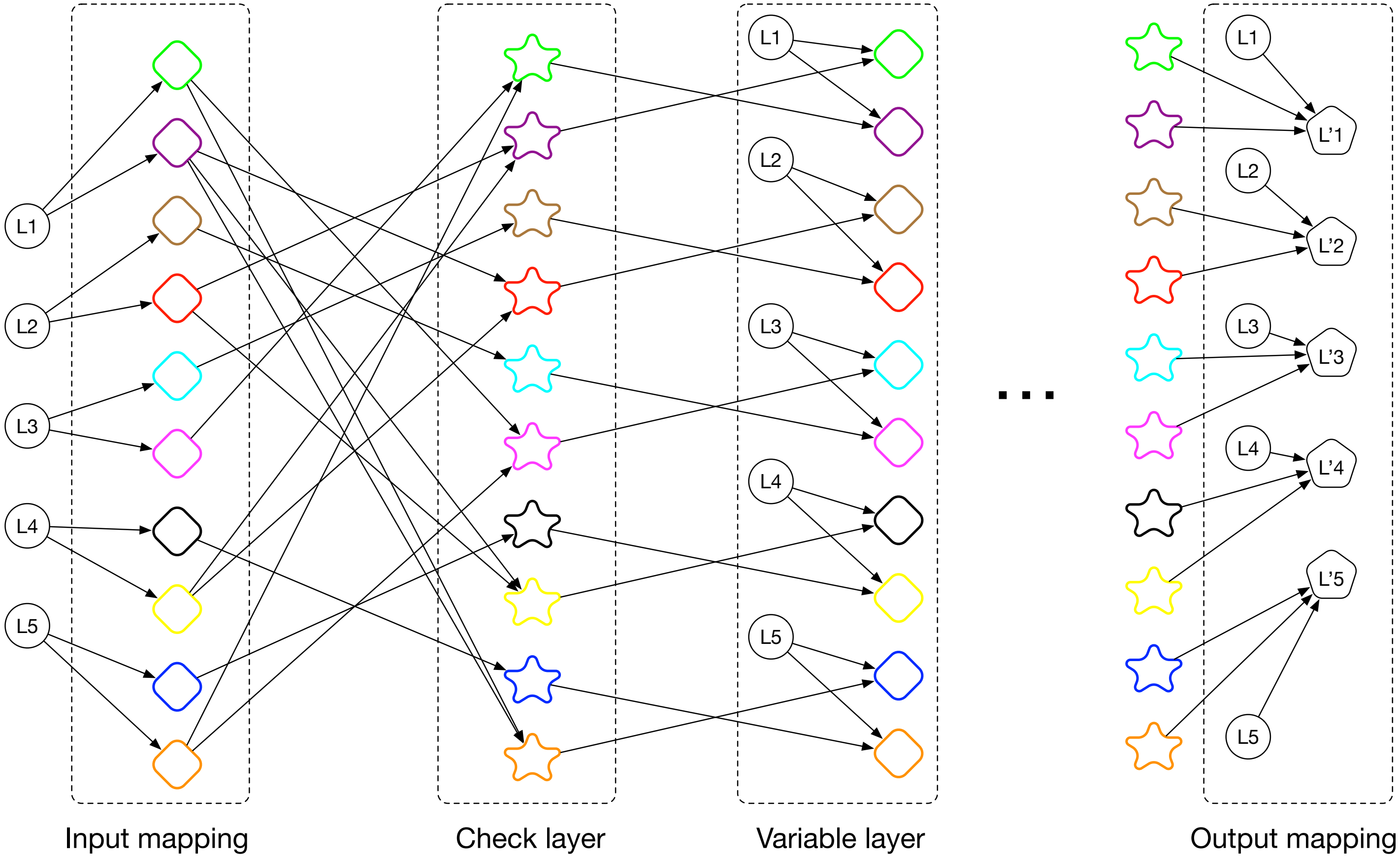
Method 2. NN from Tanner graph





Skoltech
Skolkovo Institute of Science and Technology

Method 2. NN from Tanner graph



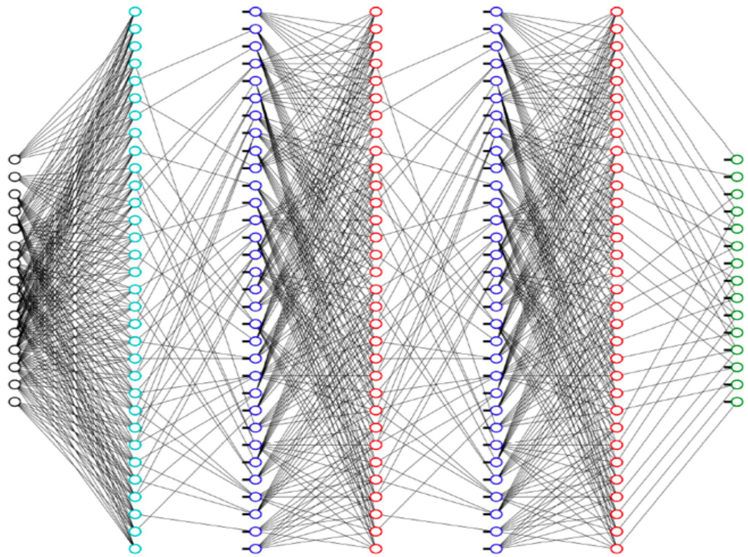
Method 2. NN architecture

<u>Features:</u>

- Use Tanner graph to construct NN;
- Sparse connectivity;
- Special activation functions (Min-Sum rules);

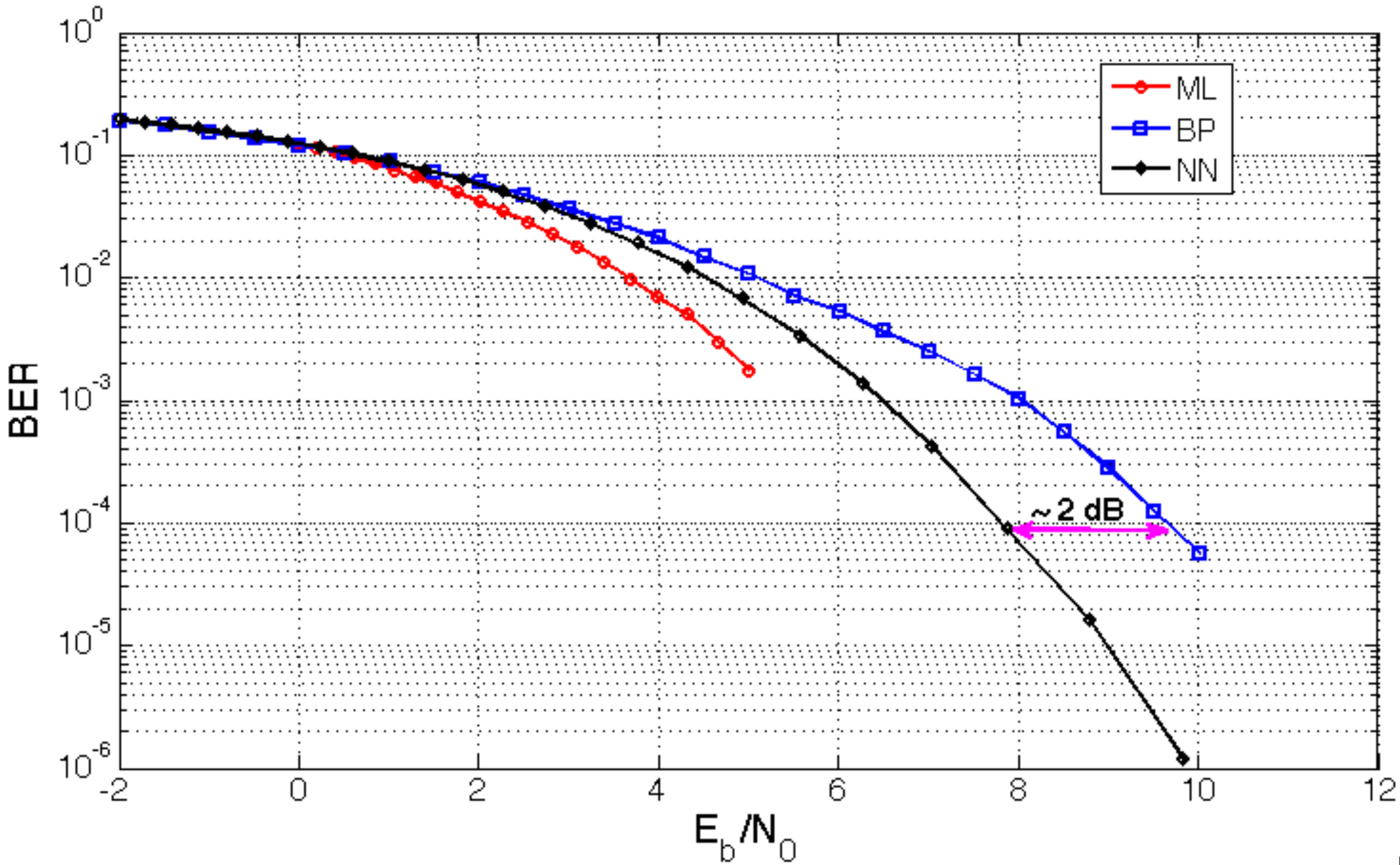
<u>Advantages:</u>

- Small number of trainable parameters (scales linearly with the code length and the number of layers);
- Training on zero codeword only;
- Training on single SNR (in what follows ***design SNR***)



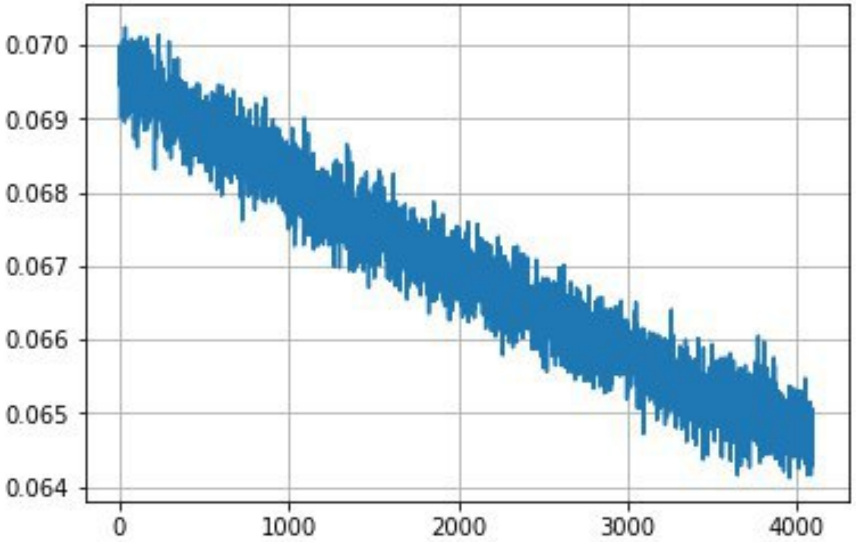
Deep Learning Methods for Improved Decoding of Linear Codes // arXiv:1706.07043

Method 2. Simulation results



Skoltech
Skolkovo Institute of Science and Technology

Method 2. Training details



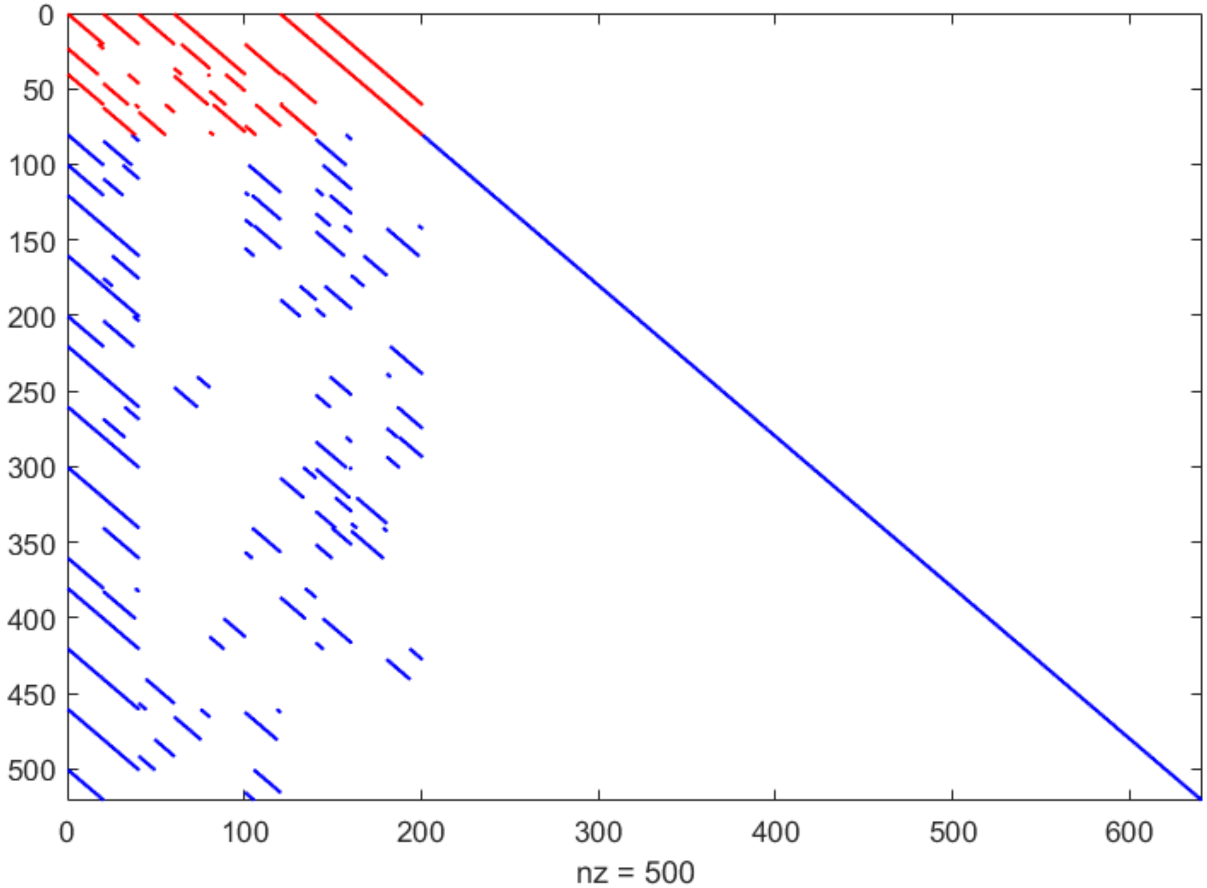
Training iterations

- Training can be continued, training loss still did not converge;
- NN is not a subject for overfitting;
- Batch size = 65K words;
- Total training iterations = 12K;
- 5 decoding iterations;
- design E_bN_0 = 1 dB, but still stable for different EbN0 values!

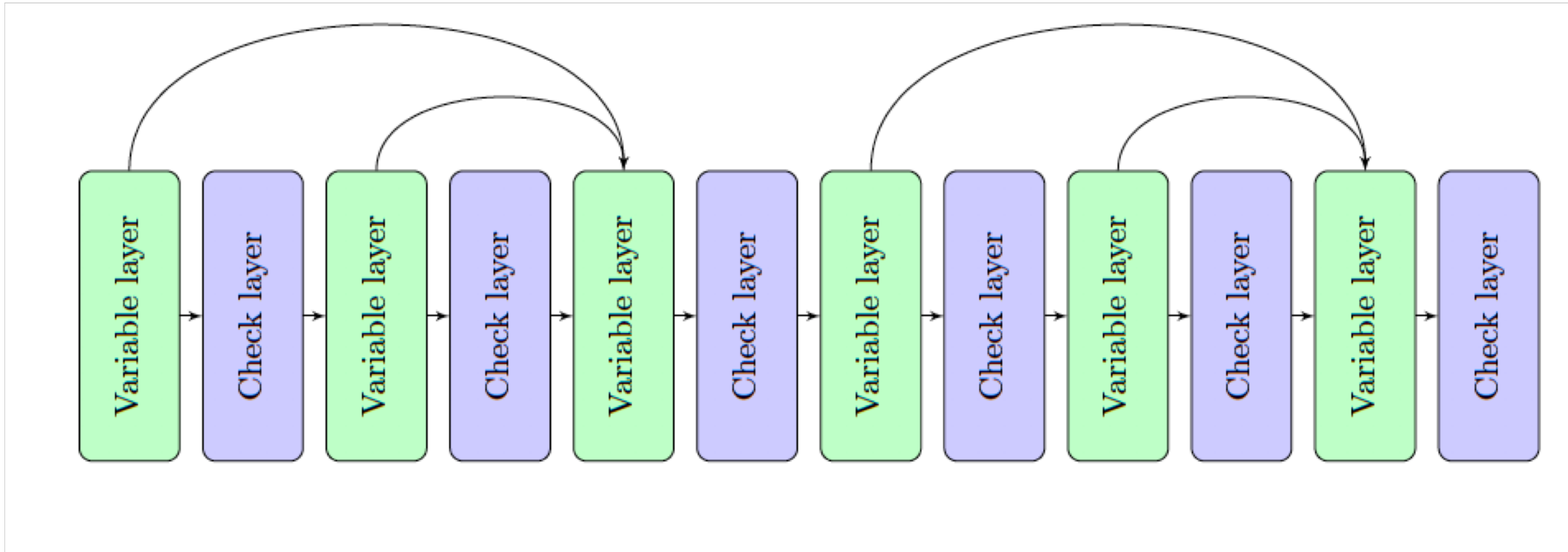
Skoltech
Skolkovo Institute of Science and Technology

What about longer codes?

5G LDPC codes (k=120 bits)

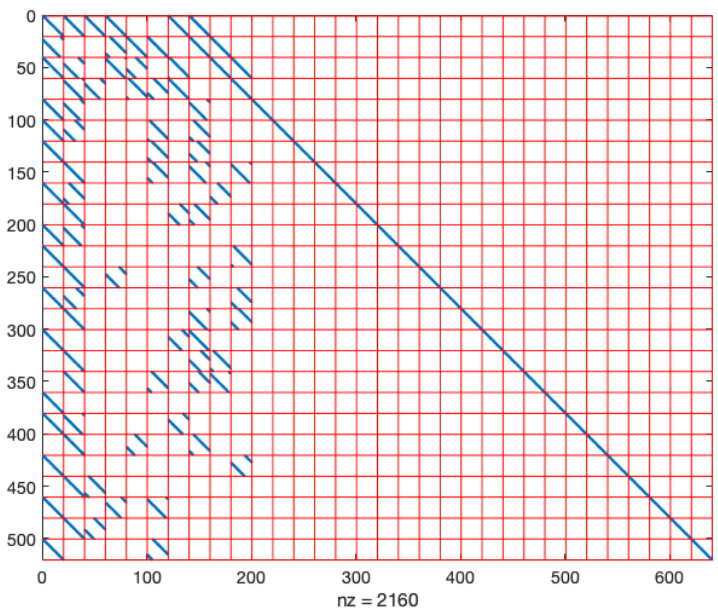


Residual connections



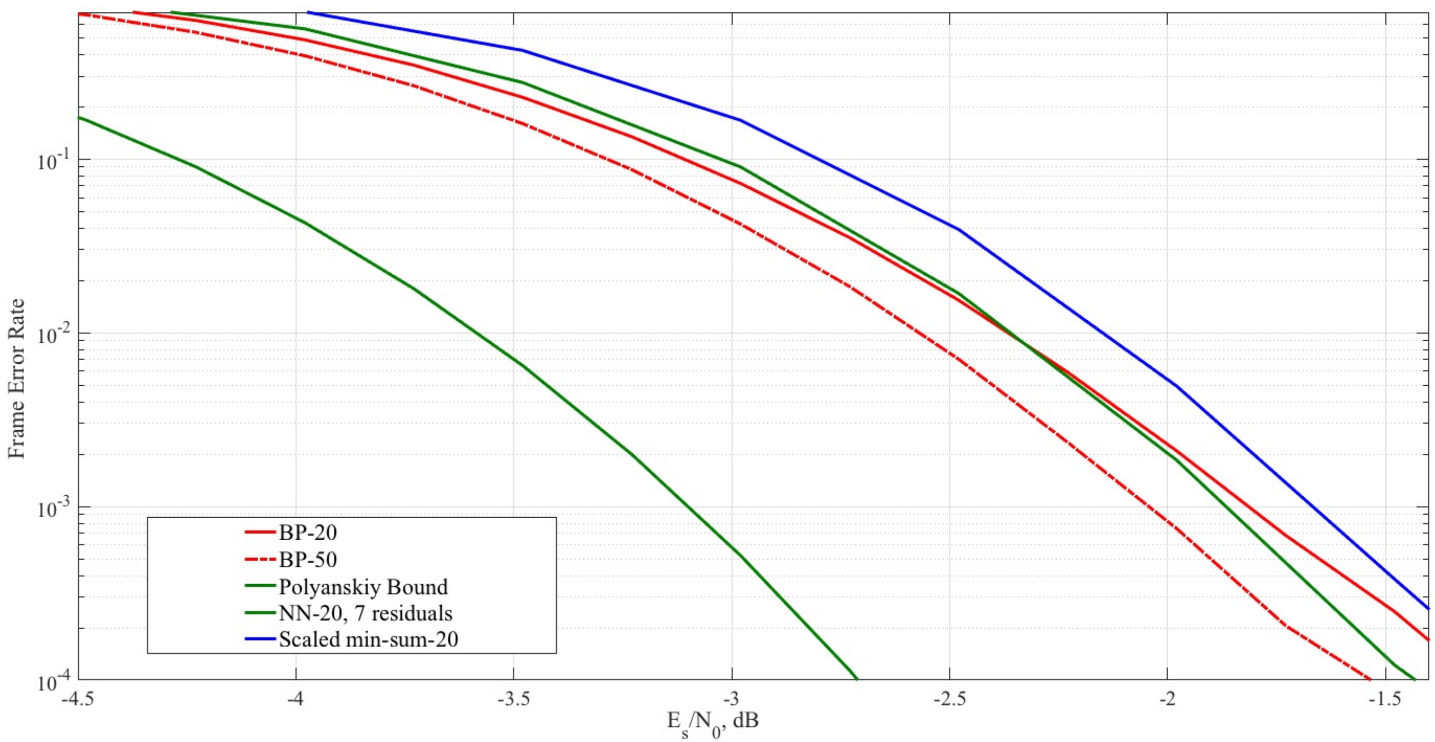
Weight sharing

5G LDPC codes are quasi-cyclic ones. This means that the code parity check matrix H consists of circular matrices (circulants). This property can be used in multiple forms. The first option is to decode the same codeword multiple types with different shifts. This kind of diversity can improve decoder performance. Another option is to embed this diversity into the neural network. This leads to weight sharing. The trainable vector size can be reduced and the training procedure becomes more efficient without any performance loss.

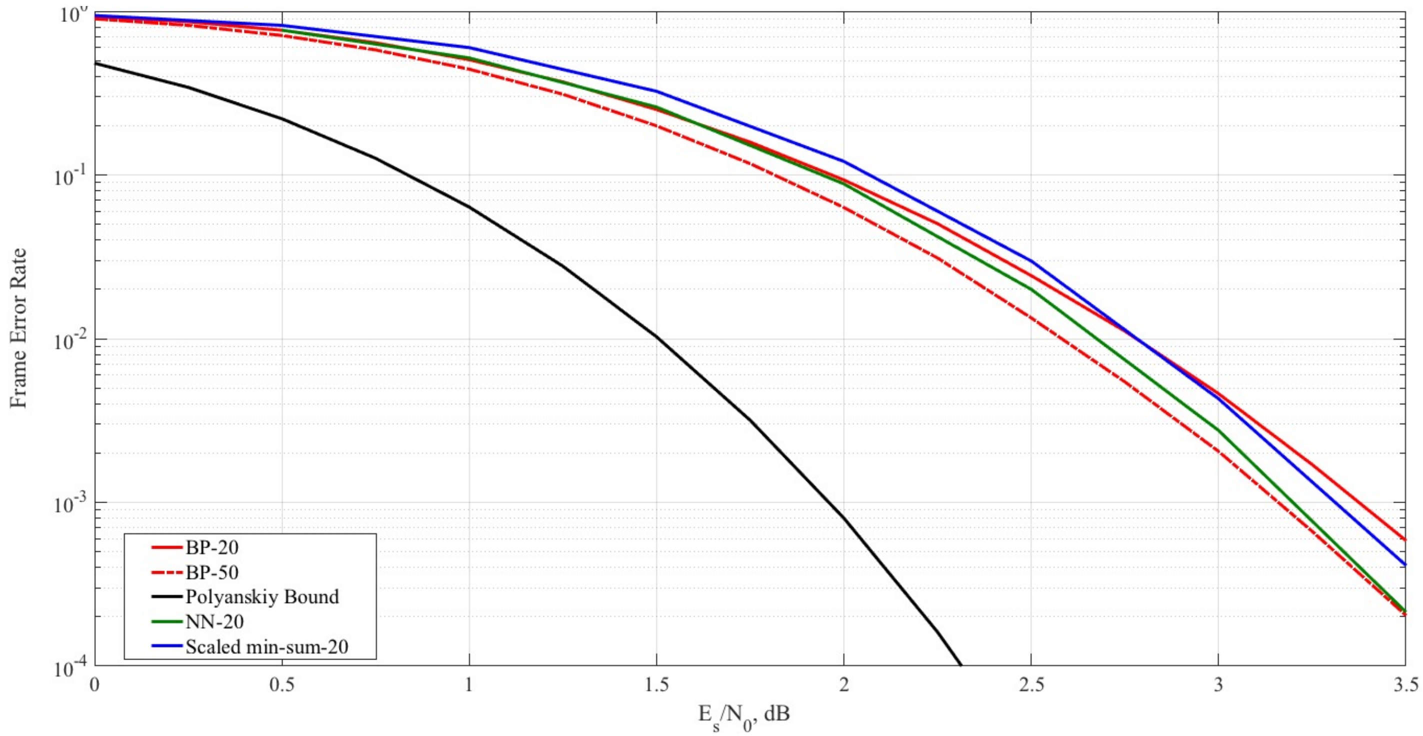


Skoltech
Skolkovo Institute of Science and Technology

Simulation results, 5G LDPC codes, BG2, K=120, R=0.2

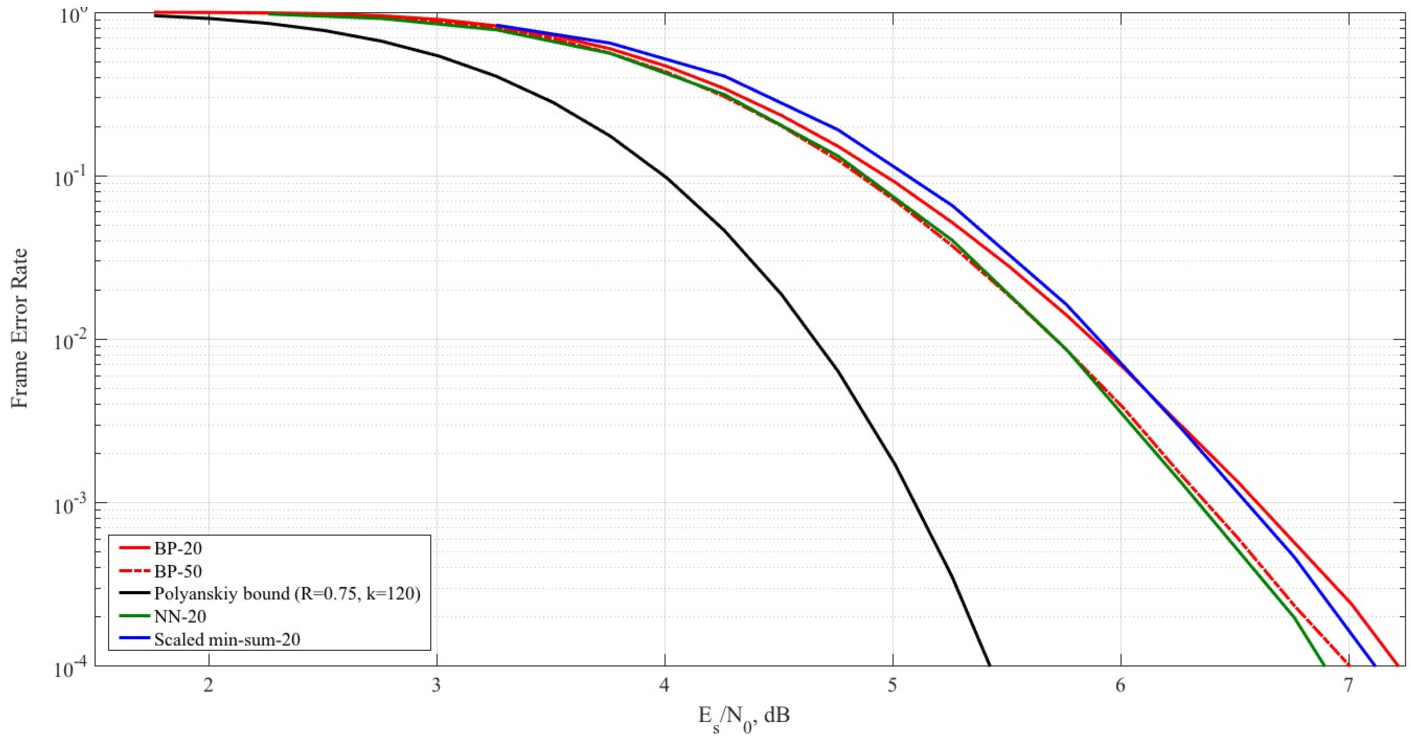


Simulation results, 5G LDPC codes, BG2, K=120, R=0.5



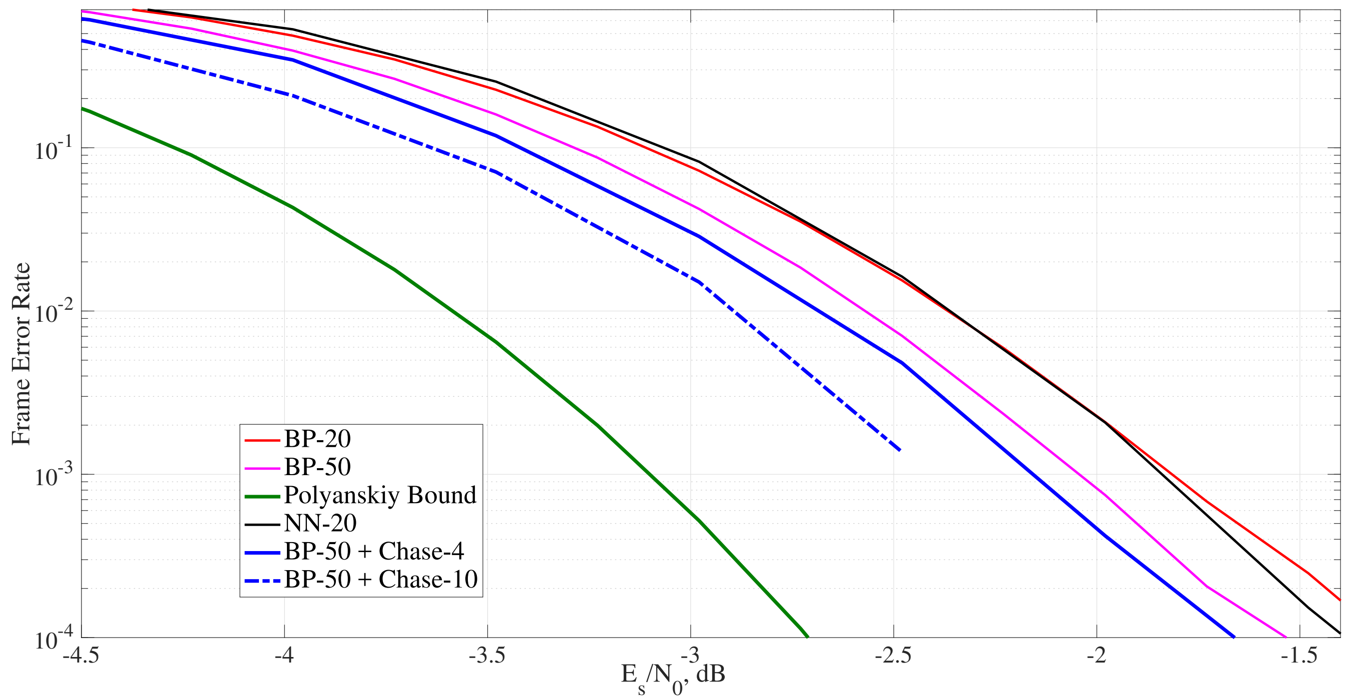
Skoltech
Skolkovo Institute of Science and Technology

Simulation results, 5G LDPC codes, BG2, K=120, R=0.75



Possible ways to improve the performance

- → OSD decoding
- → Chase decoding

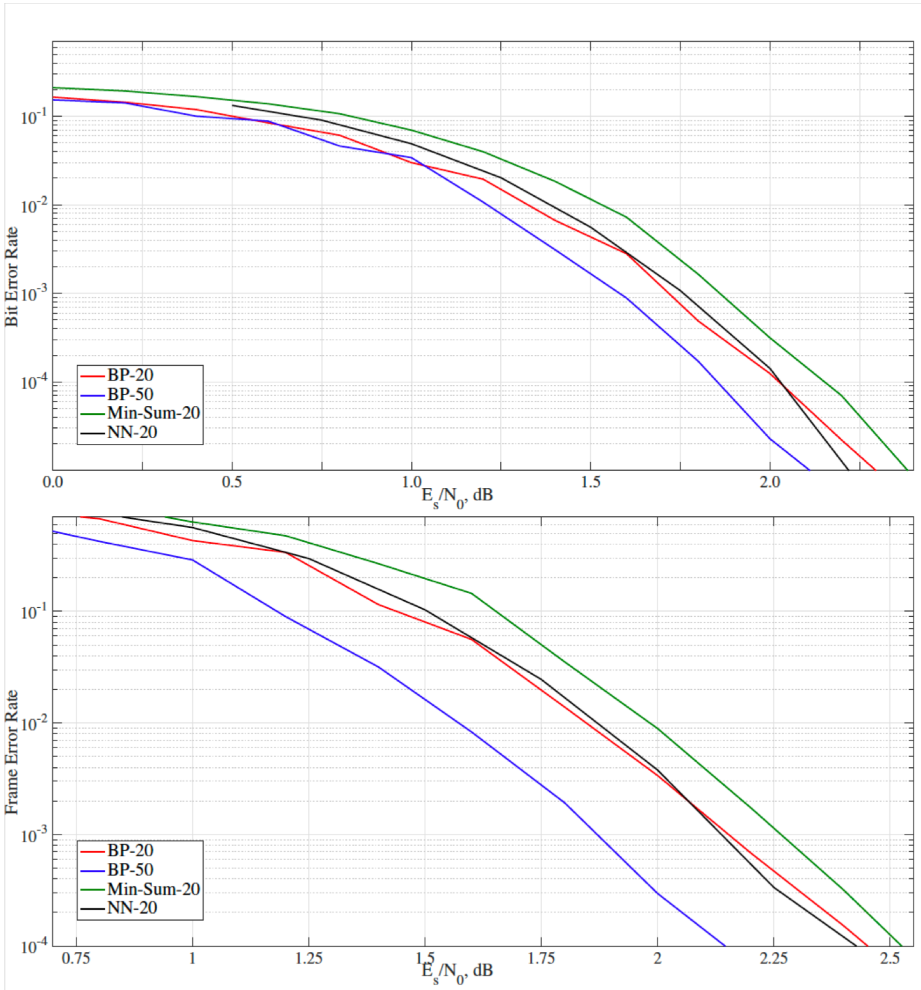


Skoltech
Skolkovo Institute of Science and Technology

What about medium length codes?

Skoltech
Skolkovo Institute of Science and Technology

Simulation results, 5G LDPC codes, BG2, K=512, R=0.5



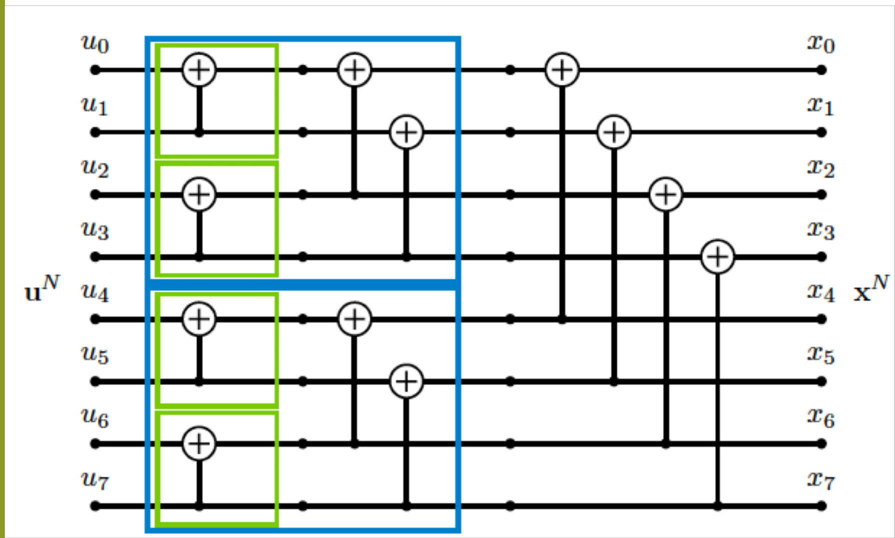
Skoltech
Skolkovo Institute of Science and Technology

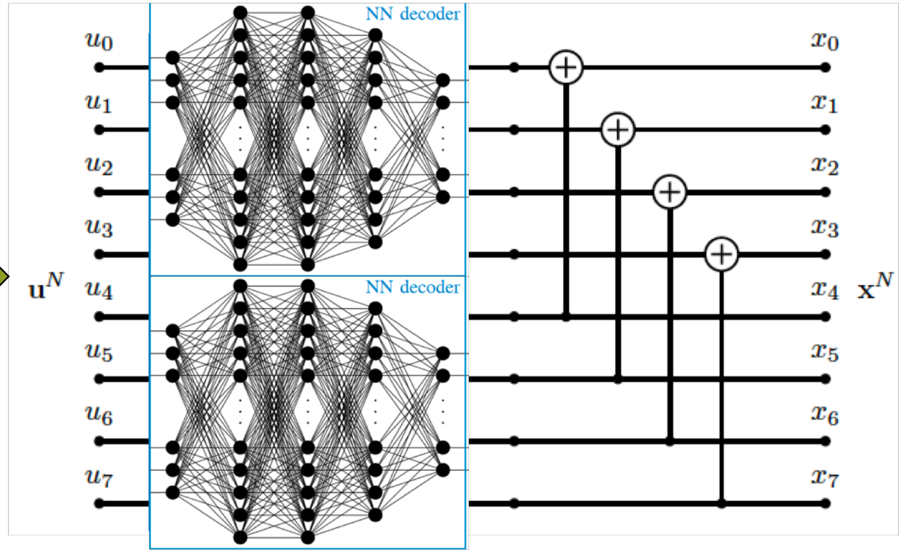
LDPC codes, summary

- NN decoder outperforms Sum-Product decoder in high SNR region as the weights help the decoder to deal with trapping sets;
- NN decoding is a promising method for decoding of short and moderate length LDPC codes;

Polar and Reed-Muller codes (N=128, K=64)

- Tanner graph method (Method 2) applied to the whole code gives bad results. The reason is simple: Min-Sum rules are very suboptimal in this case. We start with a very bad decoder and the weight optimization cannot help;
- Proposed solution (we know how to decode shorter codes – let’s use partitioning)

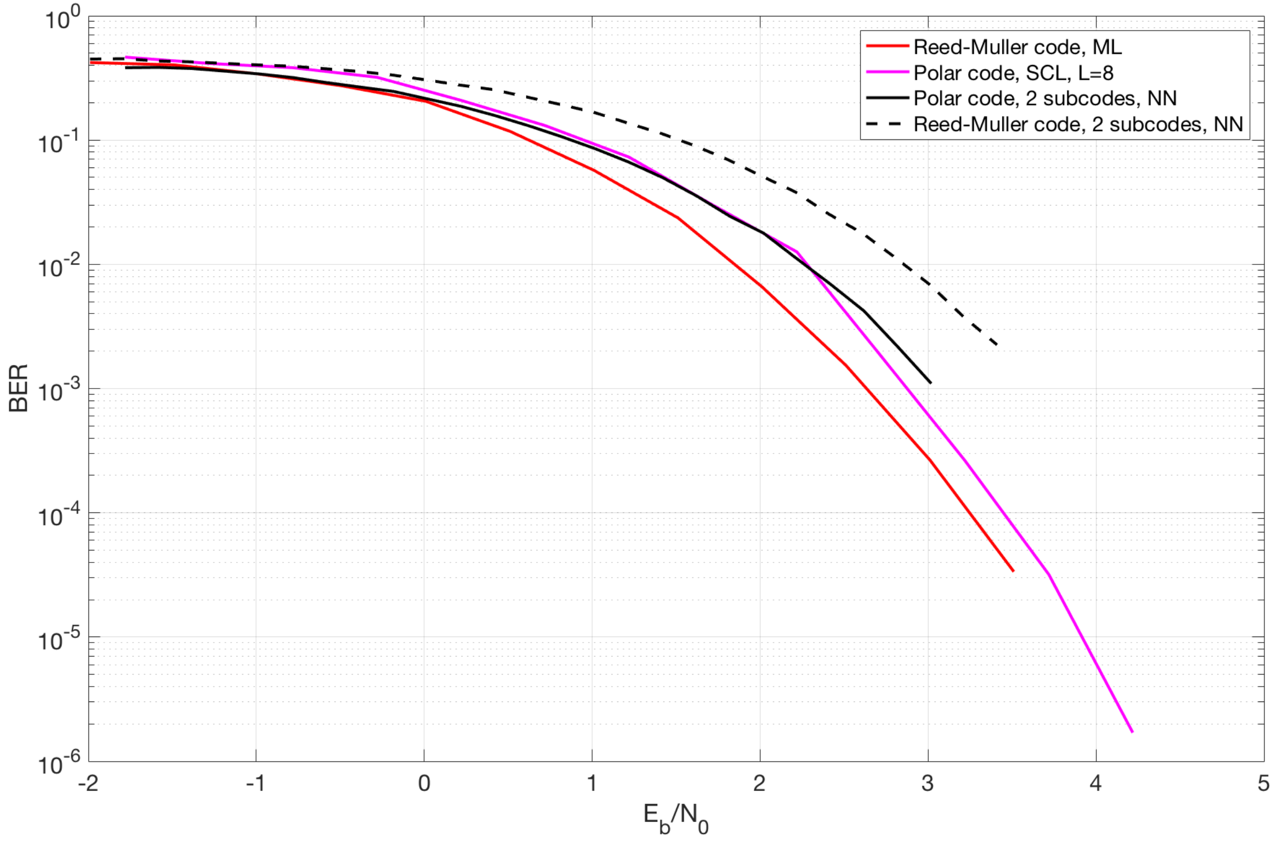




Skoltech
Skolkovo Institute of Science and Technology

Polar and Reed-Muller codes (N=128, K=64)

- We used general NN method (Method 1) to train subcodes;
- 3 hidden layers (128, 64, 32);
- The performance is slightly worse, that the performance of SCL;
- Significant latency reduction (NN decoder is parallel);

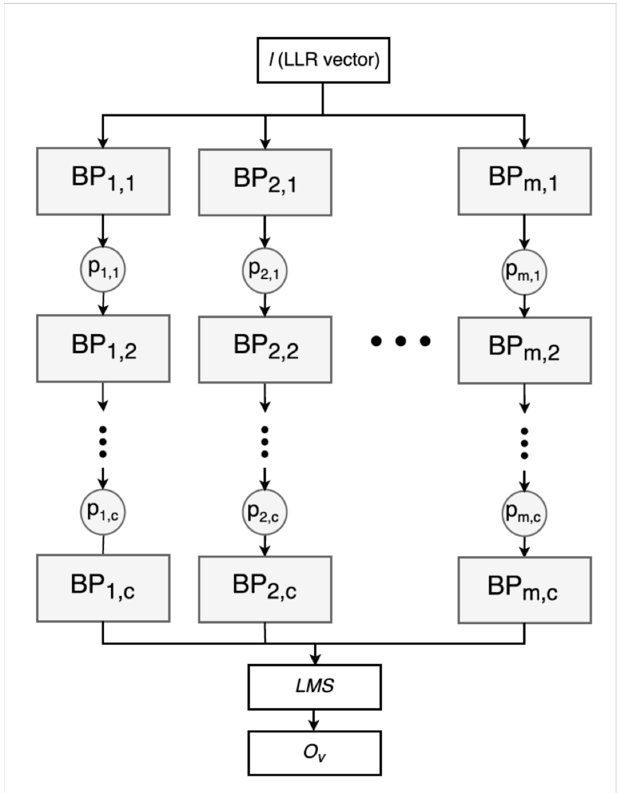


Polar and Reed-Muller codes, summary

- Reed-Muller codes have better ML performance, but NN decoder performance is worse for them;
- The performance of SCL decoder for Polar codes is very close to ML decoder, so there is no room for improving the performance with NN decoder;
- At the same time the NN decoder is parallel and helps to reduce latency and memory consumption (we do not have a list any more);
- NN decoder is suitable for moderate length Polar codes;
- Need to optimize the Polar code construction method for NN decoder;

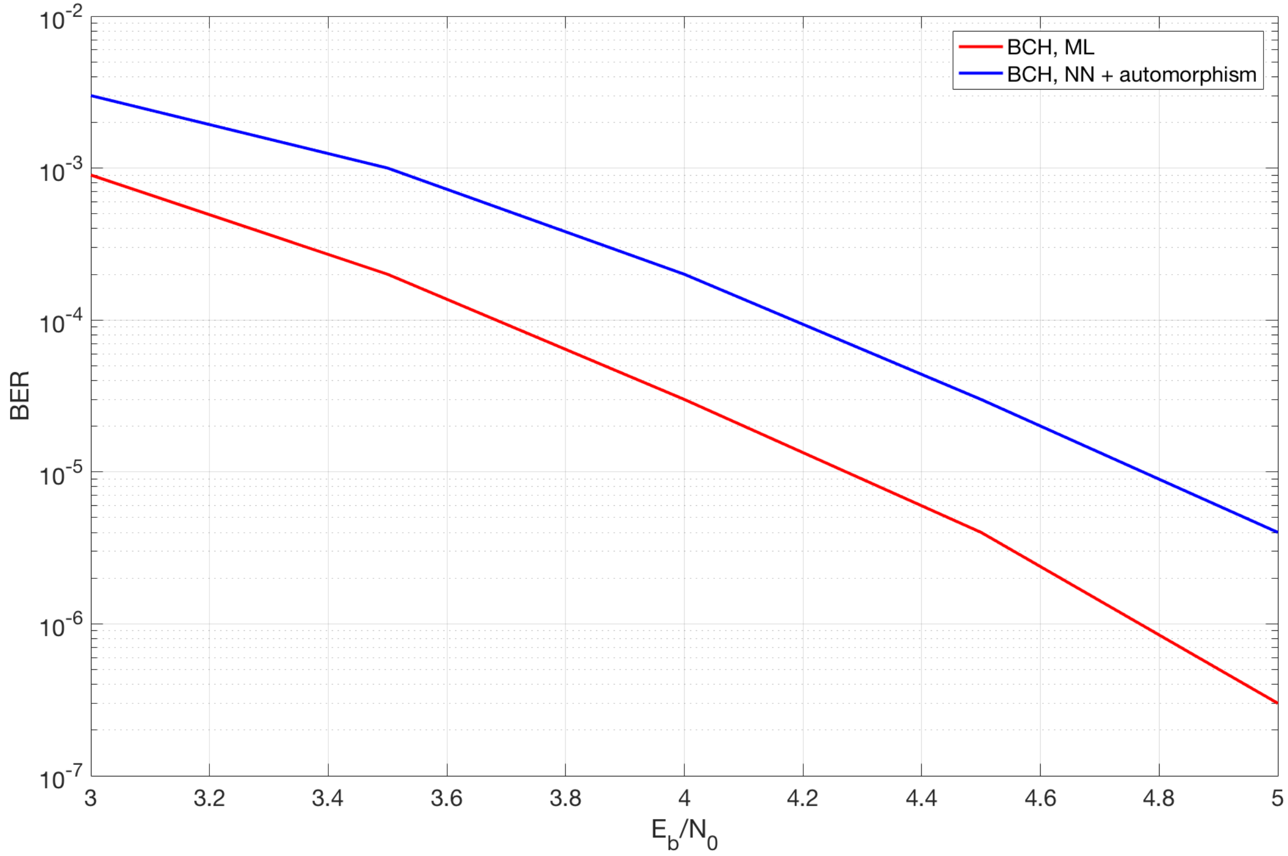
BCH code (N=63, K=36)

- Tannen graph based decoder leads to bad performance, but can be improved is we use special permutations from the code automorphism group;
- Automorphism group of BCH (and any cyclic code) includes cyclic shifts;
- The decoder was modified as shown in the figure;

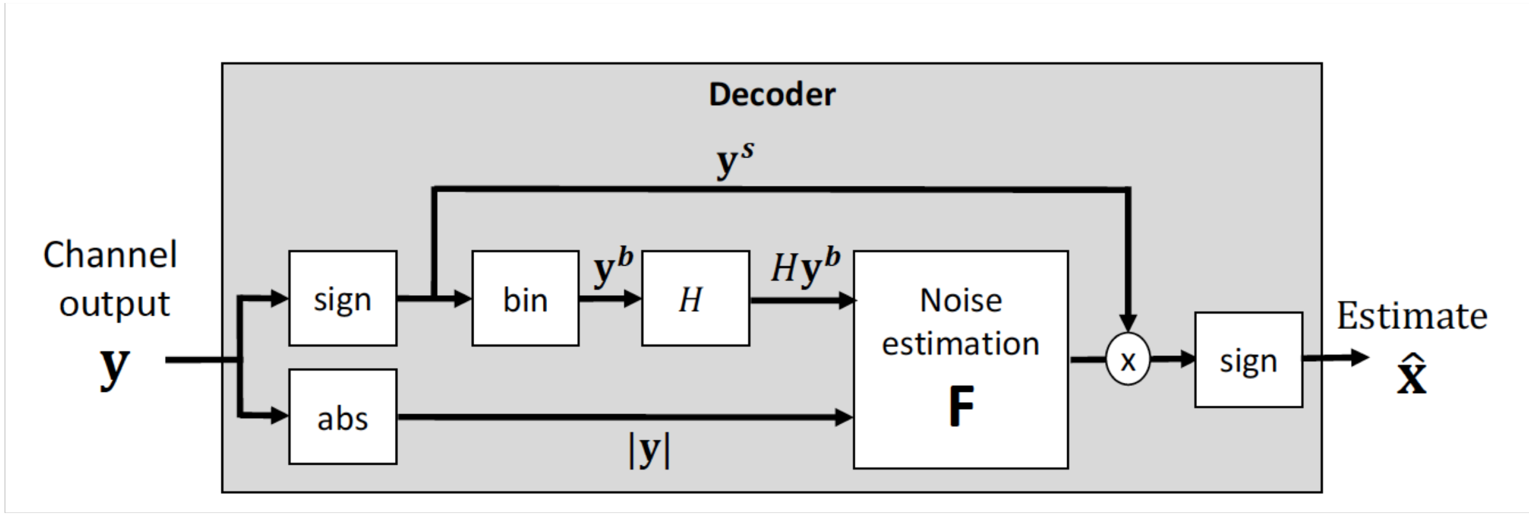


BCH code (N=63, K=36)

- We see, that the performance is very close to ML;
- The complexity is big, need to simplify the method;



Syndrome based decoding with neural networks



Amir Bennatan, Yoni Choukroun, Pavel Kisilev, Deep Learning for Decoding of Linear Codes - A Syndrome-Based Approach, https://arxiv.org/abs/1802.04741.

Thank you for your attention!

Skoltech

Skolkovo Institute of Science and Technology