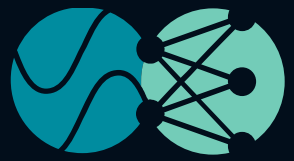


FunKI

Artificial Intelligence for Mobile Communications

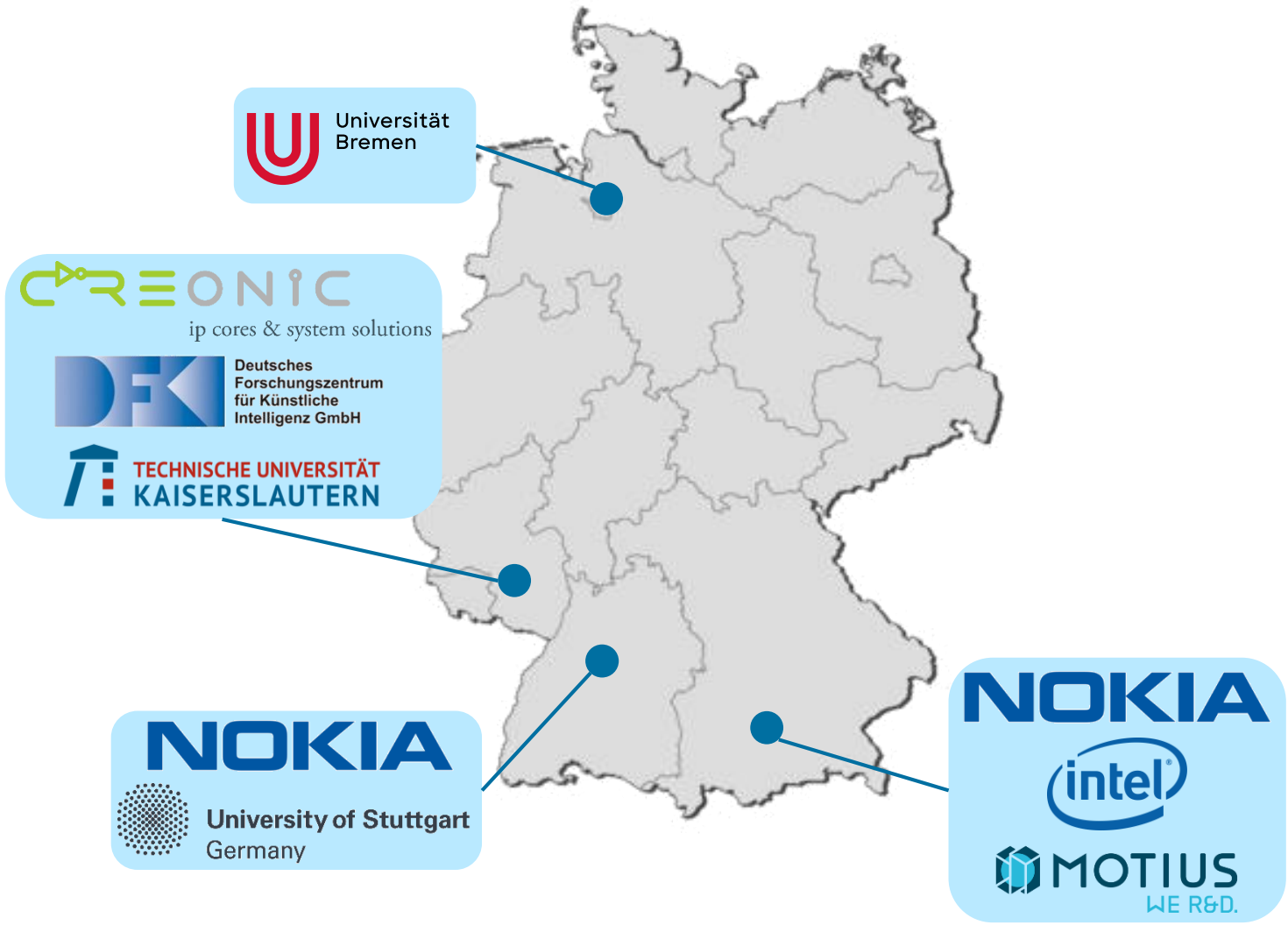
Dr. Dirk Wübben

25. Fachtagung Mobilkommunikation
3.-4.11.2021

CREONIC ip cores & system solutions
Deutsches Forschungszentrum für Künstliche Intelligenz GmbH
intel
MOTIUS WE R&D.
GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung
NOKIA
TECHNISCHE UNIVERSITÄT KAISERSLAUTERN
Universität Bremen
University of Stuttgart Germany

Funkkommunikation mit Künstlicher Intelligenz (FunKI)

Radio Communication with Artificial Intelligence



- Volume
  - 6.28 M€
- Duration
  - 05/2020 – 05/2023
- Coordinator
  - University of Bremen
    Prof. Dr. Armin Dekorsy
    Dr. Dirk Wübben
- Internet
  - www.funki.tech

GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung

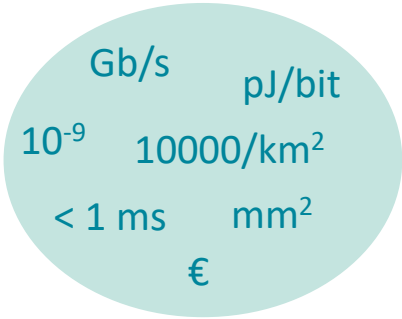
Problem Statement

Source: http://www.emfexplained.info

- Modern wireless communication systems are required that can support massive increase of devices while being both powerful and resource-efficient

- Higher protocol layers have successfully applied methods of **artificial intelligence** (**AI**)
- Model-driven approach for PHY and MAC are common
  - Good and practicable models used under idealized assumptions
  - Models enable the development of efficient procedures and algorithms

- **Challenge: Implementations with traditional methods become increasingly difficult**
  - Conflicting requirements in terms of reliability, transmission rate, number of terminals per area, latency, available radio resources, energy efficiency, complexity, and hardware costs
  - Mathematical modelling of the problem is no longer possible (model deficit) or leads to very complex models (algorithm deficit)



- **Data-driven machine learning (ML) methods** represent a promising approach for gaining an understanding of the model or for system and technology design

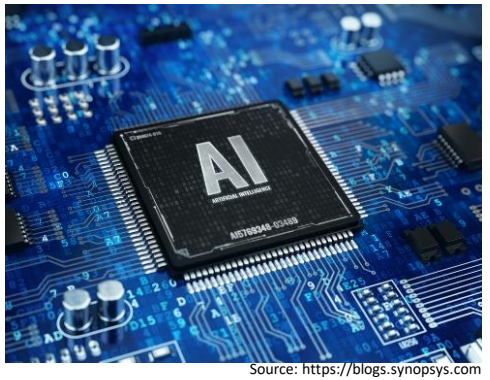

Source: https://www.nicepng.com

Scopes of Learning

- **Offline Learning**
  - Replacement of *classical* signal processing tasks by pre-trained ML-blocks (e.g. NNs)
  - Training is done offline and resulting ML-block is finally used in the actual system
  - Properties: May reduce latency and/or improves the resulting BER performance

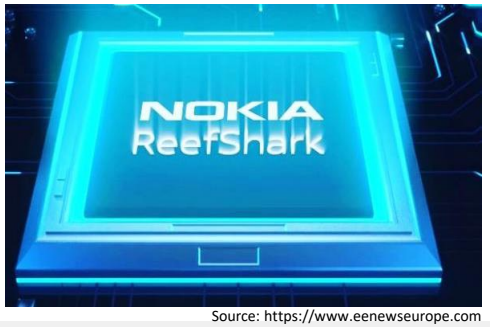

Source: https://macropolo.org

- **Online Learning**
  - Implementing AI at run-time to adapt/refine a communication system to a specific channel or to hardware impairment
  - On the fly adaption to effects that are unfeasible to model in classical channel models
  - Example: Adaptive end-to-end training of the autoencoder with continuously re-training



Source: https://blogs.synopsys.com

- **Design-time Learning**
  - Transfer insights from the learning process back to classical signal processing algorithms by applying the outcome in classical receiver structures
  - Leverage existing transceivers by means of machine learning and by new insights from learning algorithms


+


Source: https://www.eenewseurope.com

Main topics

Conceptual design, optimization and evaluation of essential components

- Transmitter, channel, receiver
- Channel estimation, prediction, feedback

FPGA hardware implementation / ASIC design

- Implementation of NN-based transceiver components by hardware accelerators

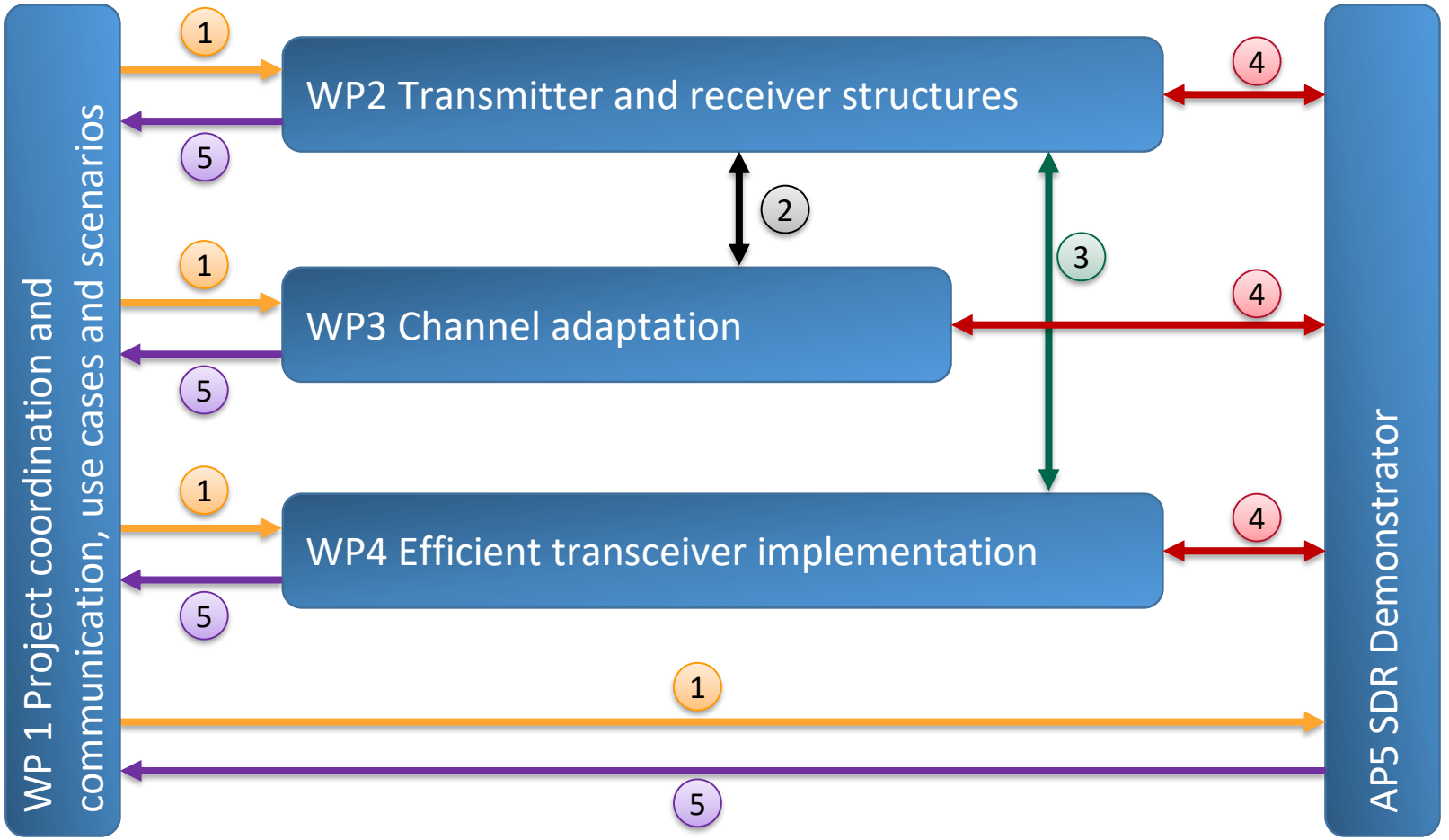
Training data generation

- Measurement prototype for multi-antenna/multi-carrier channel measurements
- Measured/real channel data for massive MIMO systems

Testing and demonstration of the technologies in selected fields of application

- SDR-based demonstrator implementing selected AI components

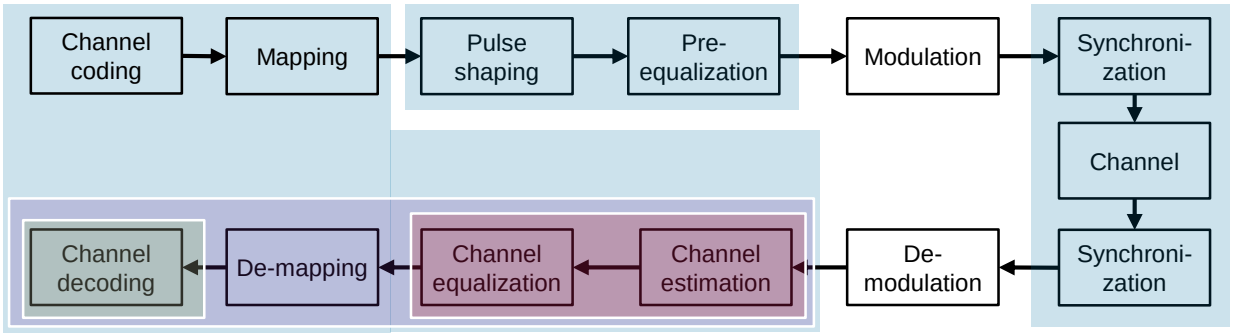
Workpackages



1. Scenarios, KPIs
2. Exchange of trained transceivers and of channel measurements
3. Iterative workflow for the transceiver implementation („bidirectional research approach“)
4. Bidirectional exchange „SDR Demo“
5. Exchange of knowledge and results

WP2 Transmitter and receiver structures

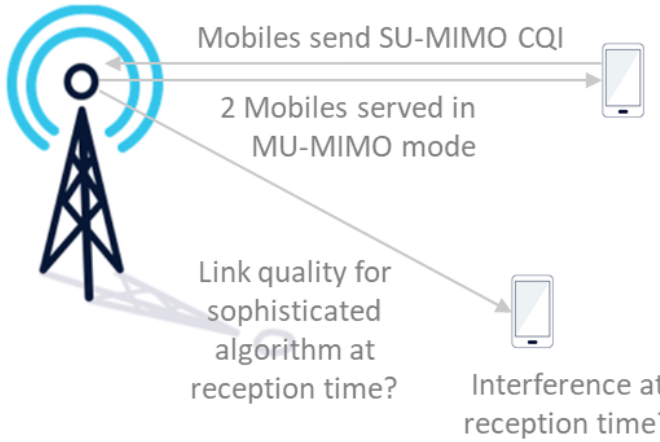
- **Motivation**
  - SotA: Handcrafted processing blocks derived from channel models, e.g., coding, pulse shaping, channel estimation
  - Those models can only generalize or approximate the real channel
    → mismatch between processing block and the real channel possible
  - Dividing a system into single and separate blocks may not be optimal
    - How to combine several communication blocks?
    - No single model can cover all realizations
- **Goal**
  - Optimization of single components of as well as the whole communication system by application of deep learning
- **Approaches**
  - Measurements and a following data-driven optimization of single blocks
  - Redesign of a communication system to allow end-to-end training (e.g. autoencoder)



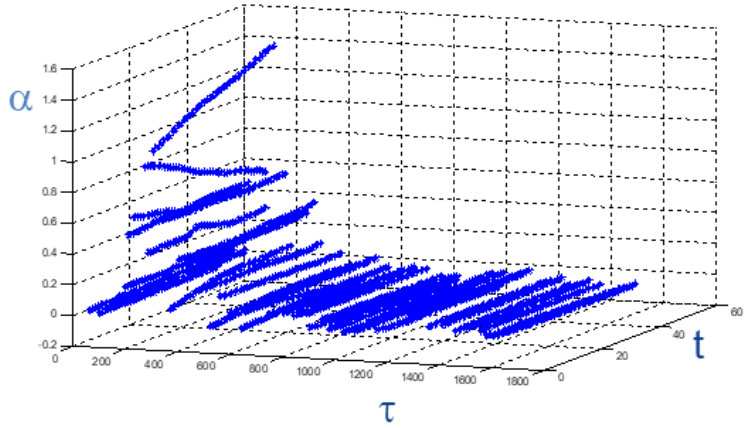
WP3 Channel adaptation

- **Motivation**
  - A very well tuned channel adaptation helps to realize the gains promised by enhanced algorithms (MU-MIMO, Beamforming, CoMP, CoSch, etc.)
  - Parameter estimation and CSI prediction is the basis for channel adaptation
  - PRACH measures about frequency and timing offset is key for further parameter estimation and prediction
- **Goal**
  - Develop AI based algorithms that enhance system behavior by improvement of PRACH measurements, channel estimation & prediction, and channel adaptation
- **Approaches**
  - Channel measurements in relevant scenarios and according channel modelling
  - AI based profiling of multipath components for channel estimation & prediction
  - AI based PRACH detection and parameter estimation with expert knowledge
  - AI based channel adaptation that adapts to different situations and environments

challenges of link adaptation



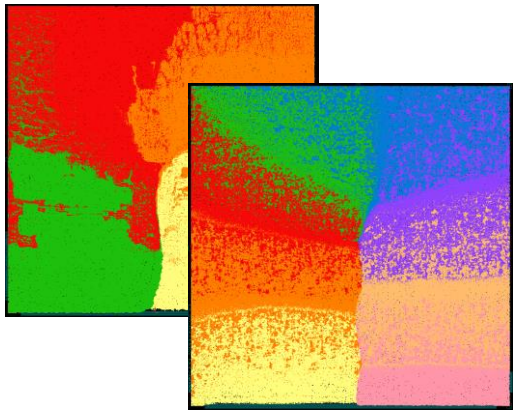
multipath components evolving over time



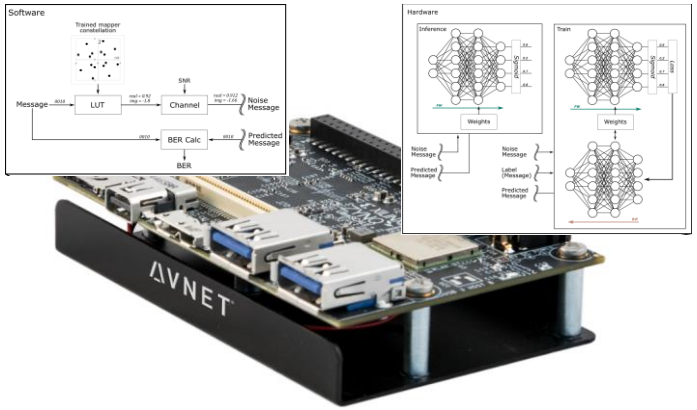
PRACH: **P**hysical **R**andom **A**ccess **Ch**annel

WP4 Efficient transceiver implementation

- **Motivation**
  - Baseband signal processing components, e.g. demodulation, equalization, channel decoding, are extremely critical building blocks w.r.t. throughput, latency and energy efficiency in the overall system and must thus be implemented as dedicated hardware accelerators
  - Large gap between “high-level” complexity analyses and actual implementation complexity
- **Goal**
  - Implementation and evaluation of AI processing elements as dedicated hardware accelerators in ASIC and FPGA technologies and fair comparison with state-of-the-art solutions
- **Approaches**
  - Implementation/simulation frameworks for fast design space exploration
  - Hardware/algorithm co-design
  - Investigate trade-off between implementation cost and performance



ASIC implementation 22nm FD-SOI:
4 and 8 iteration LDPC channel decoder



FPGA implementation: Autoencoder with trainable demapper

WP5 Testbed implementations

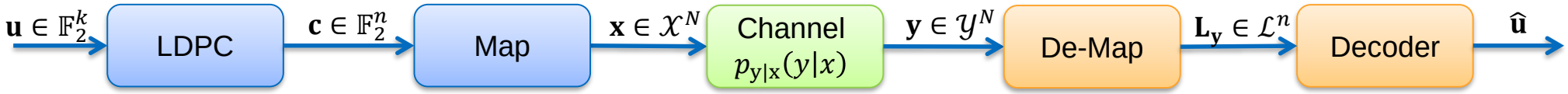
- **Motivation**
  - Various ideas and scenarios are introduced in other WP. But are they doable in existing hardware?

- **Goal**
  - SDR based Testbed for implementation and evaluation of ML-based baseband processing
  - Show the feasibility to implement the various algorithms in specific scenarios
  - Comparison with existing solutions to demonstrate possible advantages of KI in communications

- **Approach**
  - Mixed Architecture: selected KI component on FPGA (for example Xilinx Alveo), remaining test chain in GNU-Radio
  - 5G Open-Source Software

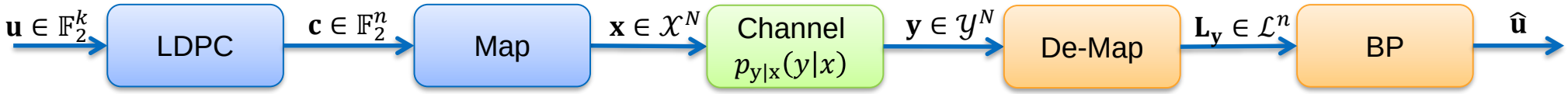
ML-based Decoder Implementations

Forward Error Correction



- Forward Error Correction (FEC)
  - Adding redundancy to improve reliability of message exchange
- Problem Statement
  - Decoder is the most complex part of the signal processing chain
  - High energy consumption / chip area by decoder implementation, iterative decoder introduces additional latency
- Goal
  - Decoder implementations meeting trade-off wrt. energy-efficiency, low-complexity, e2e performance, and latency
- Approaches: Designing decoder schemes using machine-learning driven approach
  - **MLS-BP**: Adapt common decoder implementation for short block length
  - **Discrete Decoder**: Implement decoder with very low bit resolution
  - **NN-FoC**: Execute decoder only if success is likely

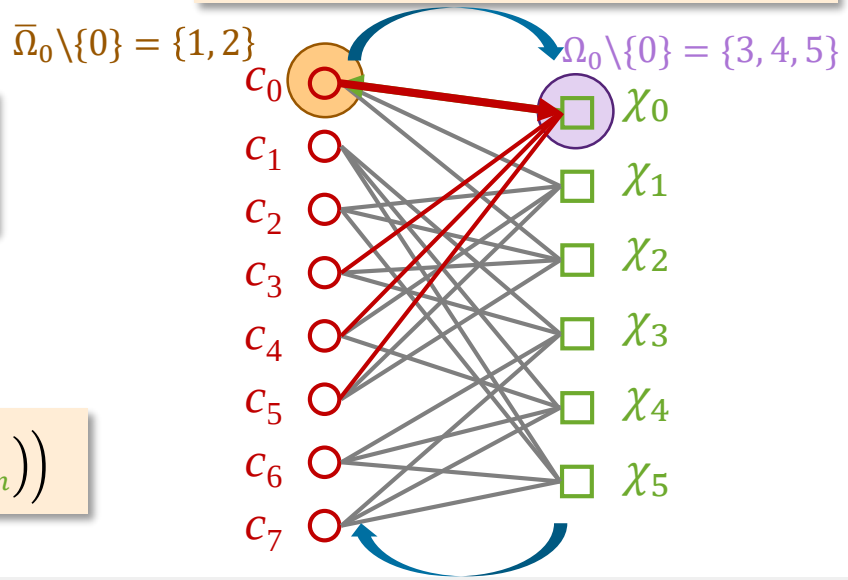
Decoding of LDPC-Codes via Belief Propagation (BP)



- **Message Passing (MP) / Belief Propagation (BP) [KFL01]**
  - Decoding by exchanging messages between variable nodes (VN) and check nodes (CN)

Initialize VN messages

$$L_{j\to m}^{(0)} = L(c_j) \quad j = 0, \dots, 7 \quad m \in \bar{\Omega}_j$$



VN update

$$L_{j\to m}^{(i)} = L_{j\to m}^{(0)} + \sum_{k \in \bar{\Omega}_j \backslash \{m\}} L_{j\leftarrow k}^{(i)}$$

Decision

$$\hat{c}_j^{(i)} = \frac{1}{2}\left(1 - \operatorname{sign}\left(L_{j\to m}^{(i)} + L_{j\leftarrow m}^{(i)}\right)\right)$$

CN update (Boxplus)

$$L_{j\leftarrow m}^{(i)} = 2 \cdot \operatorname{artanh}\left[\prod_{k \in \Omega_m \backslash \{j\}} \tanh\left(\frac{L_{k\to m}^{(i-1)}}{2}\right)\right]$$

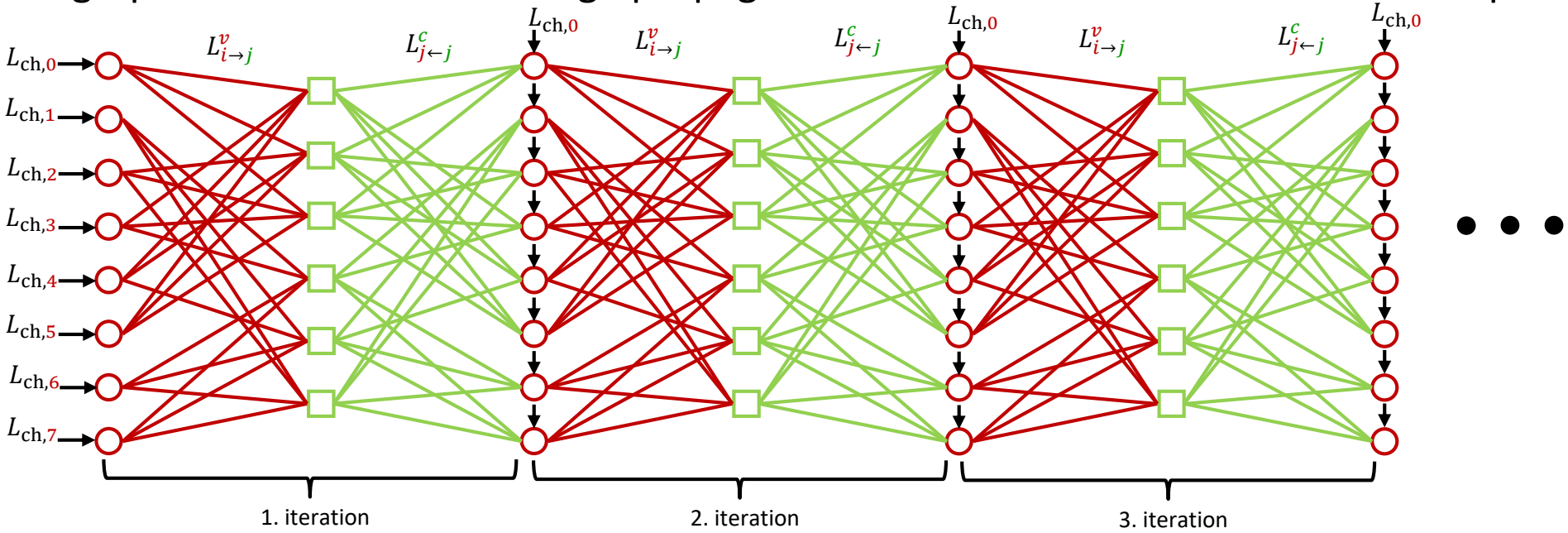
CN update (Min-Sum Approximation)

$$L_{j\leftarrow m}^{(i)} \approx \left(\prod_{k \in \Omega_m \backslash \{j\}} \operatorname{sign}\left(L_{k\to m}^{(i-1)}\right)\right) \min_{k \in \Omega_m \backslash \{j\}} \left|L_{k\to m}^{(i-1)}\right|$$

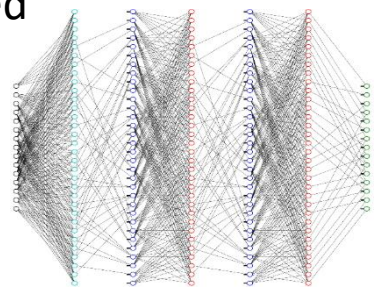
[KFL07] Kschischang, Frey, Loeliger: Factor graphs and the sum-product algorithm, IEEE Transactions on Information Theory, vol. 47, pp. 498–519, February 2001

Unrolled Factor Graph

- Unrolling factor graph to visualize the message propagation over iterations → favorable HW implement.



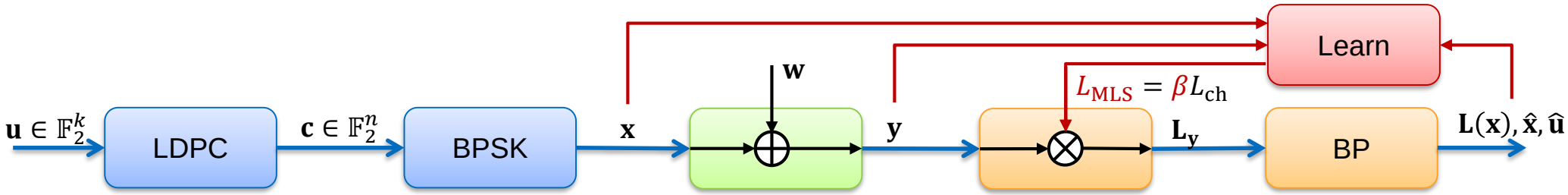
- **Problem**: BP decoder is suboptimal for short block length
  - Performance degrades due to loops in the factor graph → reliability of messages are overestimated
- **Neuronal BP (N-BP)** interprets graph as Neural Network with adaptable weights **[NBB18]**
  - Training of N-BP starts as standard BP decoder
  - Drawbacks: training for fixed number of iterations, huge training complexity (increase in k), number of weights scale with iterations



[NBB18] Nachmani, Marciano, Lugosch, Gross, Burshtein, Be'ery: Deep Learning Methods for Improved Decoding of Linear Codes, IEEE Journal Sel. Topics in Signal Processing, vol. 12, pp. 119–131, 2018

ML-Scaled Belief Propagation for Short Codes

ML-Scaled Belief Propagation (MLS-BP)



- **Alternative Approach**
  - Use single scalar factor $\beta \leq 1$ for received LLRs and common BP→ equivalent channel reliability factor $L_{\mathrm{MLS}} = \beta L_{\mathrm{ch}}$
- **Bruteforce (BF)** approach to determine L_{MLS}
  - Per SNR loop over scaling factors, simulate BER and select L_{BF}^* leading to minimum BER
- **Machine Learning Scaled Belief Propagation (MLS-BP) [HWD21]**
  - Adaptation of L_{MLS} via simple ML approach
  - **Training:** Generate training dataset $(\boldsymbol{y}, \boldsymbol{x})$ and minimize loss function

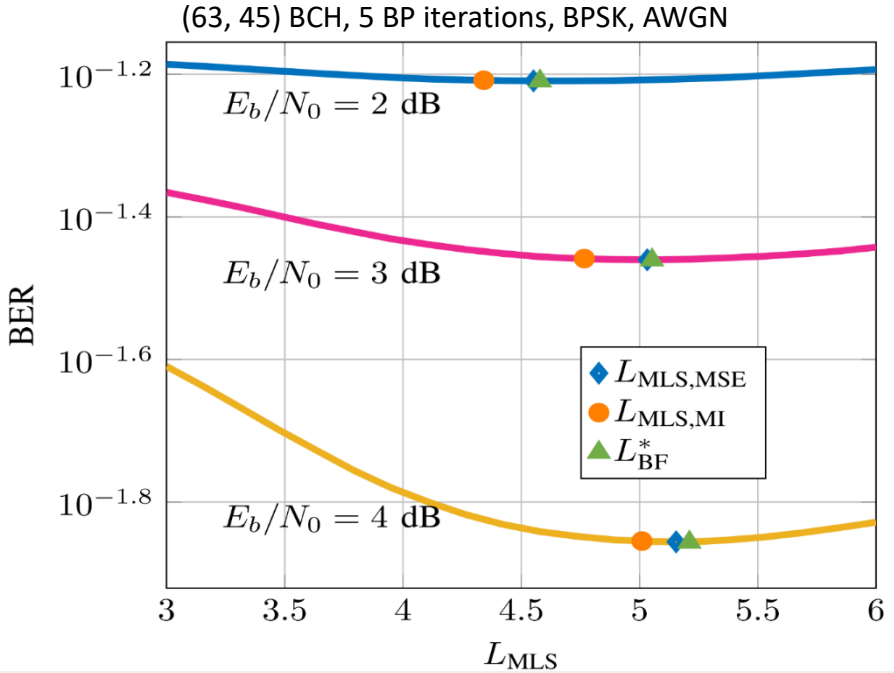
MSE loss

$$\mathcal{L}_{\mathrm{MSE}} = \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2$$

MI loss

$$\mathcal{L}_{\mathrm{MI}} = -\left(1 - \frac{1}{n}\sum_{i=1}^{n} \log_2\left(1 + e^{-x_i \cdot L(x_i)}\right)\right)$$

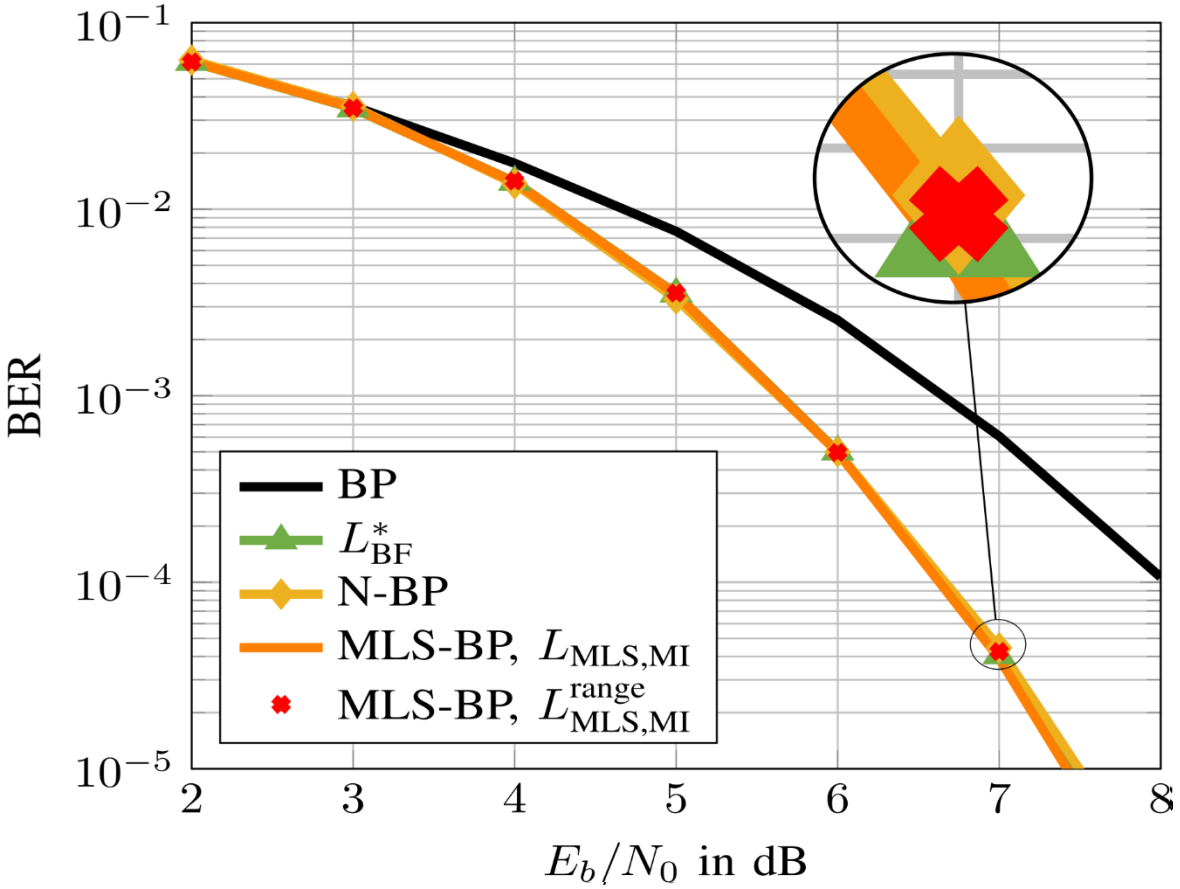
  - **Inference**: No extra overhead during execution



[HWD21] Hummert, Wübben, Dekorsy: Machine Learning Scaled Belief Propagation for Short Codes, VTC-Fall 2021

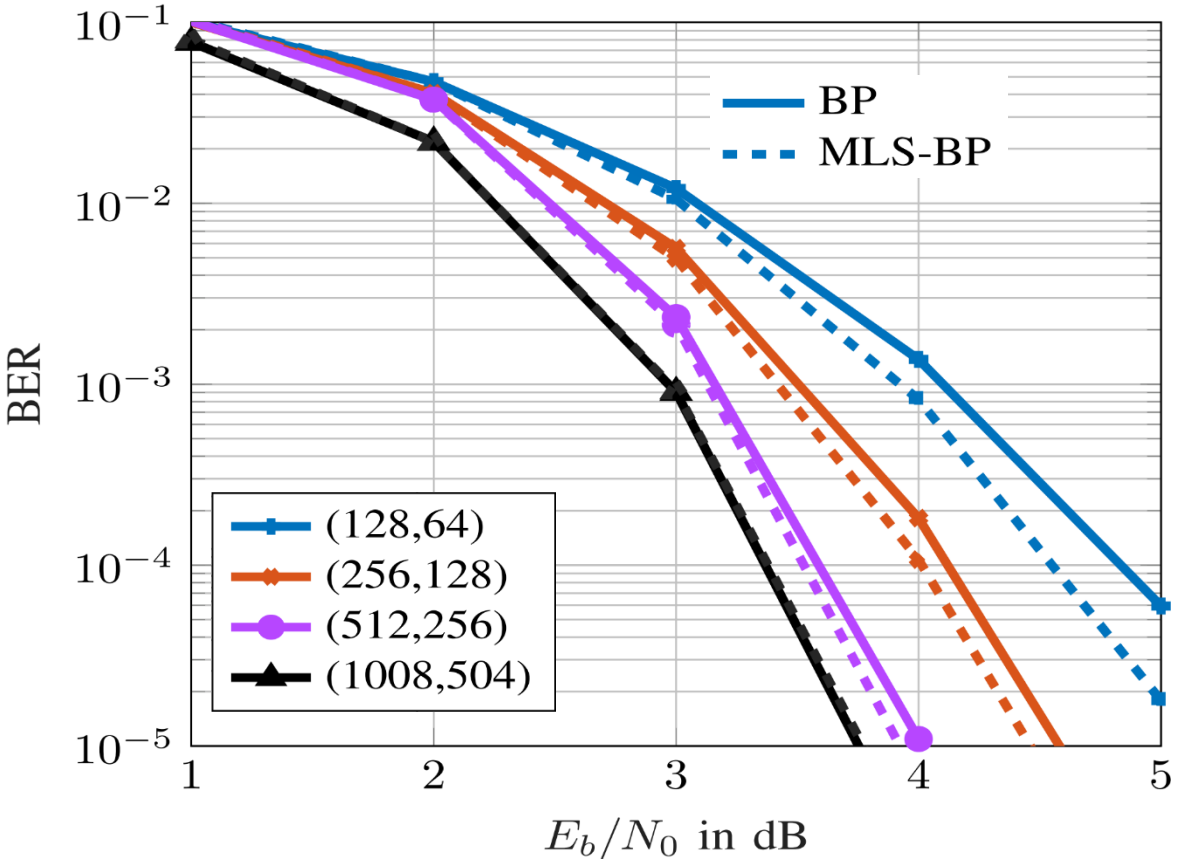
Performance for (63, 45) BCH code

- **Scenario**
  - (63, 45) BCH code, 5 BP iterations, BPSK, AWGN
- **Observations**
  - N-BP shows substantial gains to standard BP (loss due to loops in the graph)
  - MLS-BP is nearly indistinguishable to N-BP
  - Range-learned constant $L_{\mathrm{MLS}}^{\mathrm{range}}$: Use training set containing a range of SNR to learn scaling factor → nearly same performance as SNR-specific L_{MLS}
  - As expected, L_{BF}^{*} shows same performance
- **Conclusion**
  - Range-learned constant is sufficient to improve the performance of BP decoder for short codes



Performance for longer LDPC codes

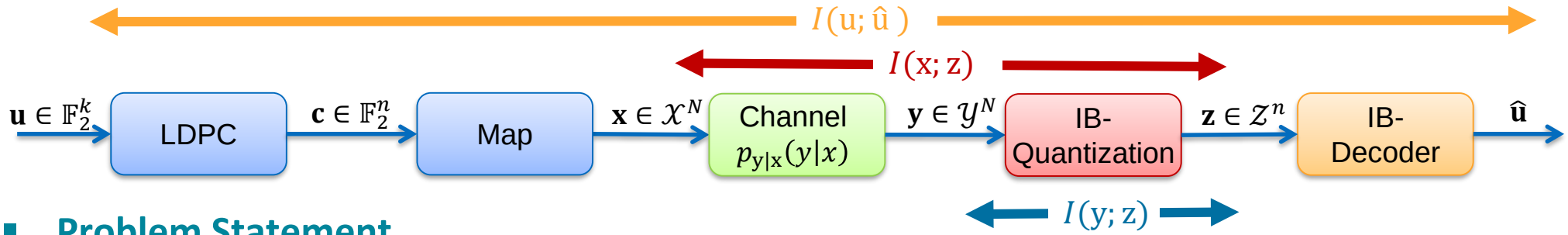
- **Scenario**
  - LDPC codes for different length, 5 BP iterations and range-learned constant $L_{\mathrm{MLS}}^{\mathrm{range}}$
- **Observations**
  - With increasing codeword length n, suboptimality of BP reduces
  - Possible gains of MLS-BP reduce as well
- **Conclusion**
  - MLS-BP introduces trainable scaling factor for LLRs
  - Instead of SNR-specific scaling factor one ranged-learned factor is sufficient
  - No adaption of BP implementation necessary
  - For short codes performance improvements are achieved *for free*



[HWD21] Hummert, Wübben, Dekorsy: Machine Learning Scaled Belief Propagation for Short Codes, VTC-Fall 2021

Discrete Decoder Implementation

Information Bottleneck Method for Discrete Decoder Design

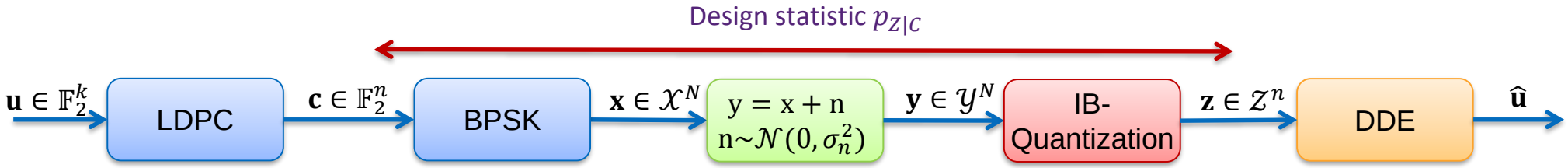


- **Problem Statement**
  - Design of efficient decoding algorithms with low bit-resolution

- **Approach**
  - Learn information processing chain by focusing only on relevant information while maximizing transmission rate
  - Mutual information to formulate trade-off between complexity and relevant information $I(\mathrm{x};\mathrm{z})$

- **Information Bottleneck Method (IBM)**
  - Relevant information processing: interest is on information of source signal x or u
  - Quantization: Trade-off between compression rate $I(\mathrm{y};\mathrm{z})$ and relevant information $I(\mathrm{x};\mathrm{z})$
  - IB-based decoder: Low internal variable resolution (e.g. 3-4 bit) and simple discrete operations

LUT-MP: Decoder Learning



- LUT-MP decoder algorithm is learned via IB ($\beta \to \infty$) in **Discrete Density Evolution (DDE)** [KYK08, KY14]

Initial VN distribution

$$p_{C,\bar{T}^{(0)}} \leftarrow p_{Z|C}\, p_C$$

CN LUT

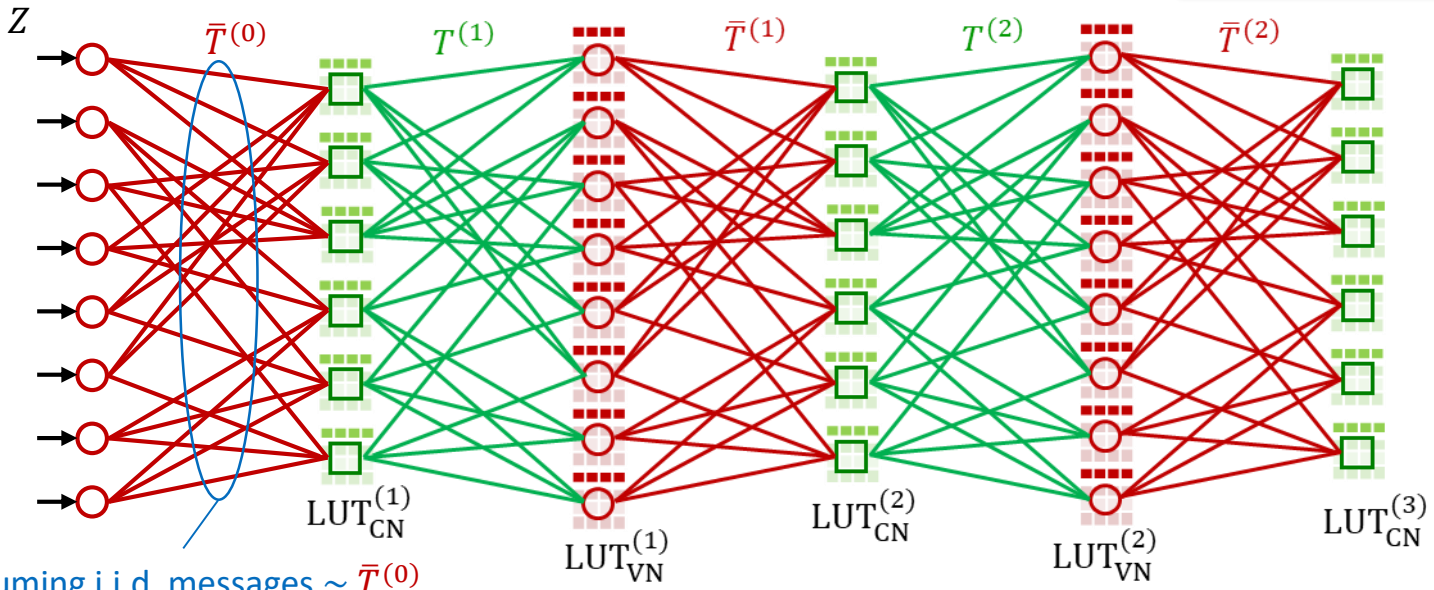
$$\mathrm{LUT}_{\mathrm{CN}}^{(i)} = \underset{\mathrm{Q}}{\operatorname{argmax}}\, I\left(C;\, T^{(i)}\right)$$

VN LUT

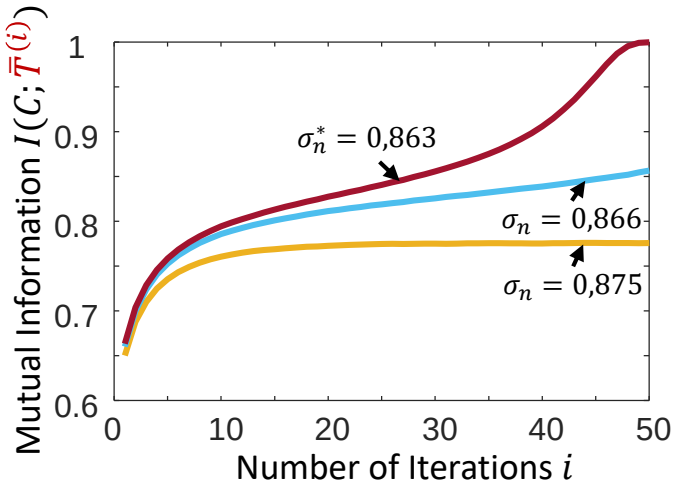
$$\mathrm{LUT}_{\mathrm{VN}}^{(i)} = \underset{\mathrm{Q}}{\operatorname{argmax}}\, I\left(C;\, \bar{T}^{(i)}\right)$$

Decision LUT

$$\mathrm{LUT}_{\mathrm{DC}}^{(i)} = \underset{\mathrm{Q}}{\operatorname{argmax}}\, I\left(C;\, \hat{C}^{(i)}\right)$$



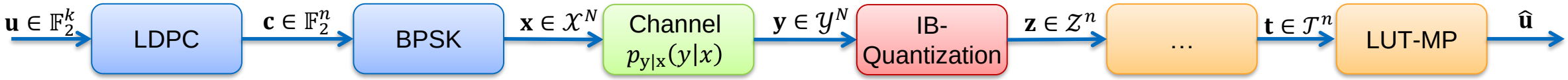
Assuming i.i.d. messages $\sim \bar{T}^{(0)}$



LUT-MP decoder is optimized for noise threshold σ_N^* of code

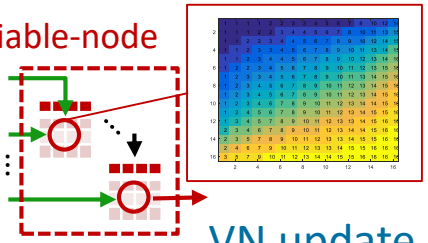
[KYK08] Kurkoski, Yamaguchi, Kobayashi: Noise thresholds for discrete LDPC decoding mappings, IEEE Globecom, 2008
[KY14] Kurkoski, Yagi: Quantization of Binary-Input Discrete Memoryless Channels, in IEEE Transactions on Information Theory, Aug. 2014

LUT-MP: Decoder Execution



- **Lookup-Table based Message Passing (LUT-MP)** [RK16]
  - Node operations in variable nodes (VN) and check nodes (CN) are implemented by **fixed LUTs**

Variable-node



VN update

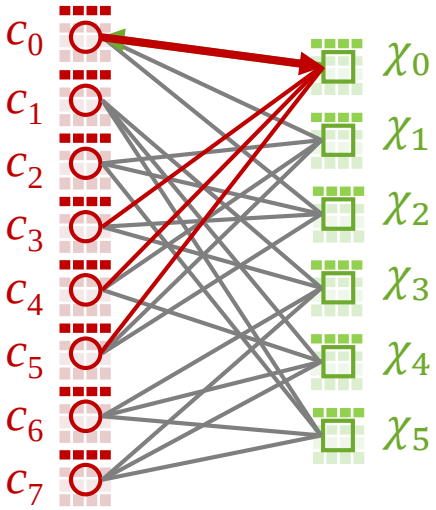
$$\bar{t}_{j\to m}^{(i)} = \mathrm{LUT}_{\mathrm{VN}}^{(i)}\left(\bar{t}_{j\to m}^{(0)}, \boldsymbol{t}_{j,m}^{(i-1)}\right)$$

Decision

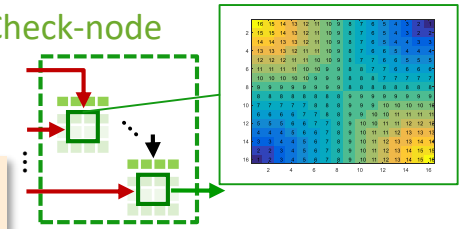
$$\hat{c}_j^{(i)} = \mathrm{LUT}_{\mathrm{DC}}^{(i)}\left(\bar{t}_{j\to m}^{(i)}, t_{m\to j}^{(i)}\right)$$

Initialize VN messages

$$\bar{t}_{j\to m}^{(0)} = \bar{z}_j \quad j = 1, \dots, 8 \quad m \in \bar{\Omega}_j$$



Check-node



CN update

$$t_{m\to j}^{(i)} = \mathrm{LUT}_{\mathrm{CN}}^{(i)}\left(\bar{\boldsymbol{t}}_{j,m}^{(i-1)}\right)$$

Min-LUT [MM20]: minimum of integers

$$\mathrm{LUT}_{\mathrm{MS}}(\bar{\boldsymbol{t}}) = \begin{cases} \min_k |\bar{t}_k| & \text{if} \quad \prod_k \mathrm{sign}(\bar{t}_k) = 1 \\ \neg \min_k |\bar{t}_k| & \text{if} \quad \prod_k \mathrm{sign}(\bar{t}_k) = -1 \end{cases}$$

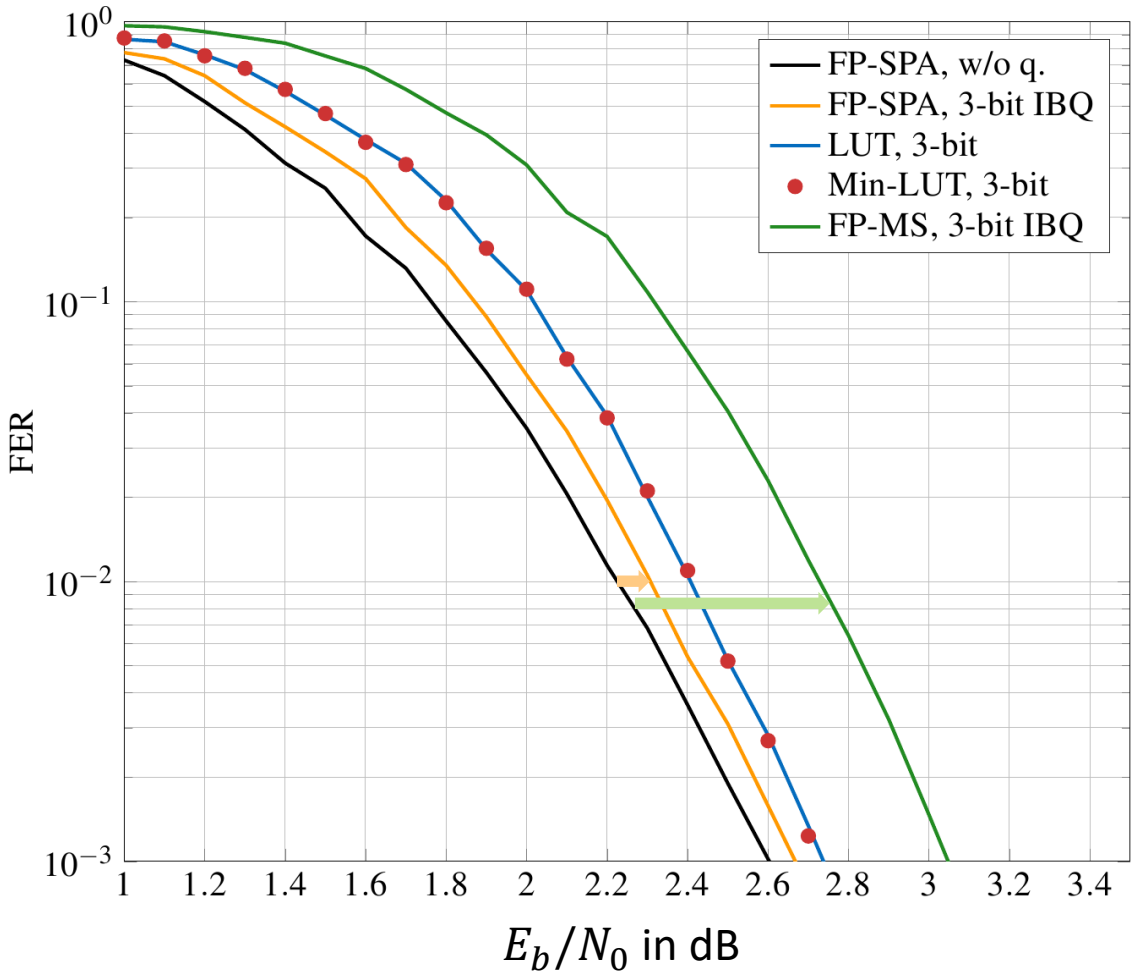
[RK16] Romero, Kurkoski: LDPC Decoding Mappings That Maximize Mutual Information, IEEE JSAC, 2016
[MM20] Meidlinger, Matz, Burg: Design and Decoding of Irregular LDPC Codes Based on Discrete Message Passing, IEEE TCOM, 2020

Performance Evaluation

- **Simulation setup**
  - (3,6)-regular LDPC Code, $n = 816$, $i_{\max} = 40$
  - Benchmark: **FP-SPA** with double precision

- **Observations**
  - **FP-SPA with 3-bit IB ADC** → 0.1 dB loss
  - **FP-MS: Min-Sum with 3-bit IB ADC** → 0.45 dB loss
  - **3-bit LUT-MP** → 0.15 dB loss
  - **3-bit Min-LUT** → almost no difference

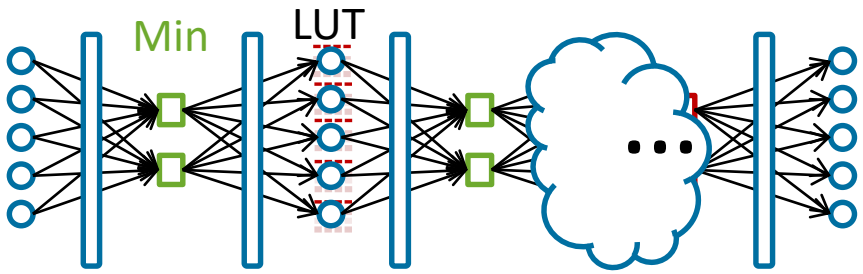
- **Conclusion**
  - Discrete decoder with 3-bit resolution performs close to SPA with double precision
  - 4-bit LUT-MP would reduce gap to **FP-SPA**



LDPC Decoder – First synthesis results

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

- Example code: $n = 24$, $k = 12$, WiMAX-like code, $d_v \in \{2, 3, 6\}$ and $d_c = 6$
- Synthesis in 28nm FD-SOI technology



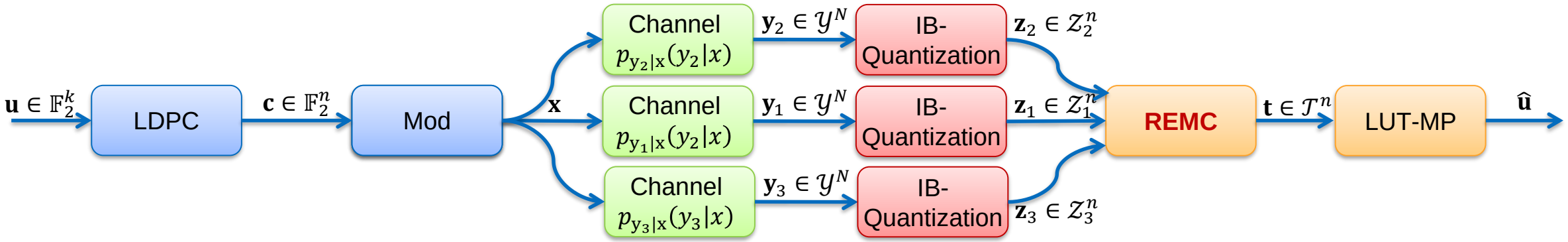
- Comparison 6-bit Min-Sum vs. 3-bit LUT-MP
  - Reduction of cell area by 14%
  - 50% reduced node interconnections due to reduction from 6 to 3 bit
- Comparison 6-bit Min-Sum vs. 3-bit Min-LUT
  - Reduction of cell area by 38%
- Comparison 6-bit Min-Sum vs. 4-bit LUT-MP
  - Larger total cell area, but 33% reduced node interconnections (6 to 4 bit)

Architecture	MS 6bit	LUT-MP 4bit	Min-LUT 4bit	LUT-MP 3bit	Min-LUT 3bit
Codeblock Size	24	24	24	24	24
Pipeline Stages	10	10	10	10	10
Target (ns)	2.00	2.00	2.00	2.00	2.00
Critial Path (ns)	1.75	1.79	1.49	1.49	1.51
Frequency (MHz)	571	559	500	671	662
Throughput (Gbps)	13.7	13.4	12.0	16.1	15.9
Total Cell Area (umm^2)	55711.9	143553.9	95262.9	47908.6	34588.6
Area wrt. MS 6bit	100%	257%	171%	86%	62%

FD-SOI: **F**ully **D**epleted **S**ilicon **O**n **I**nsulator

FunKI

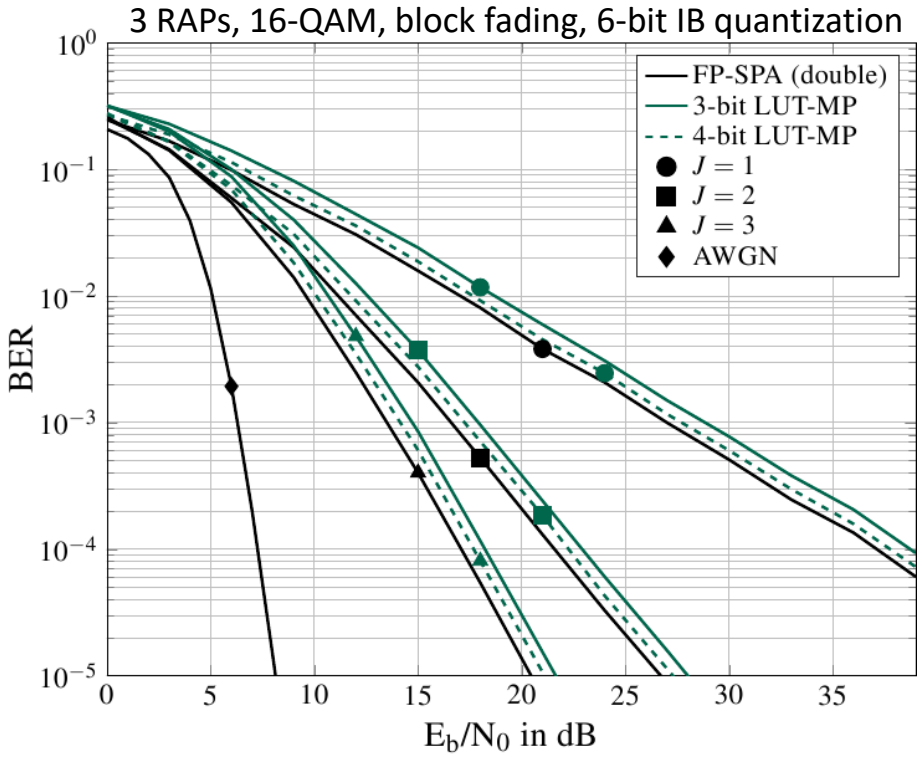
Relative Entropy based Message Combining (REMC)



- **Problem**: LUT-MP has been designed for specific e2e distribution
- **Challenge**: How to apply given LUT-MP implementation for other scenarios, e.g. higher modulation order, diversity, discrete signals with different cardinality
- **Relative Entropy based Message Combining (REMC) [MWD21]**
  - Combine messages with similar meaning $p(c_\nu|z_1, \dots, z_J)$ wrt. given decoder design distribution $p^*(c|t)$

$$t_\nu = \mathrm{Q}_{C,\nu}(z_1, z_2, \dots, z_J) = \operatorname{argmin}_{t \in \mathcal{T}} D_{\mathrm{KL}}\left(p(c_\nu|z_1, z_2, \dots, z_J) \,||\, p^*(c|t)\right)$$

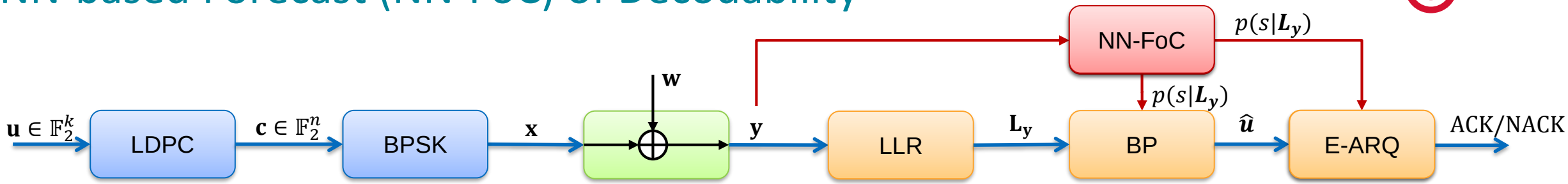
  - Forwarding 3-bit messages + LUT-MP decoder close to optimum



[MWD21] Monsees, Wübben, Dekorsy: Relative Entropy based Message Combining for Exploiting Diversity in Information Optimized Processing, WSA 2021

NN-based Forecasting of Decodability for Early ARQ

NN-based Forecast (NN-FoC) of Decodability



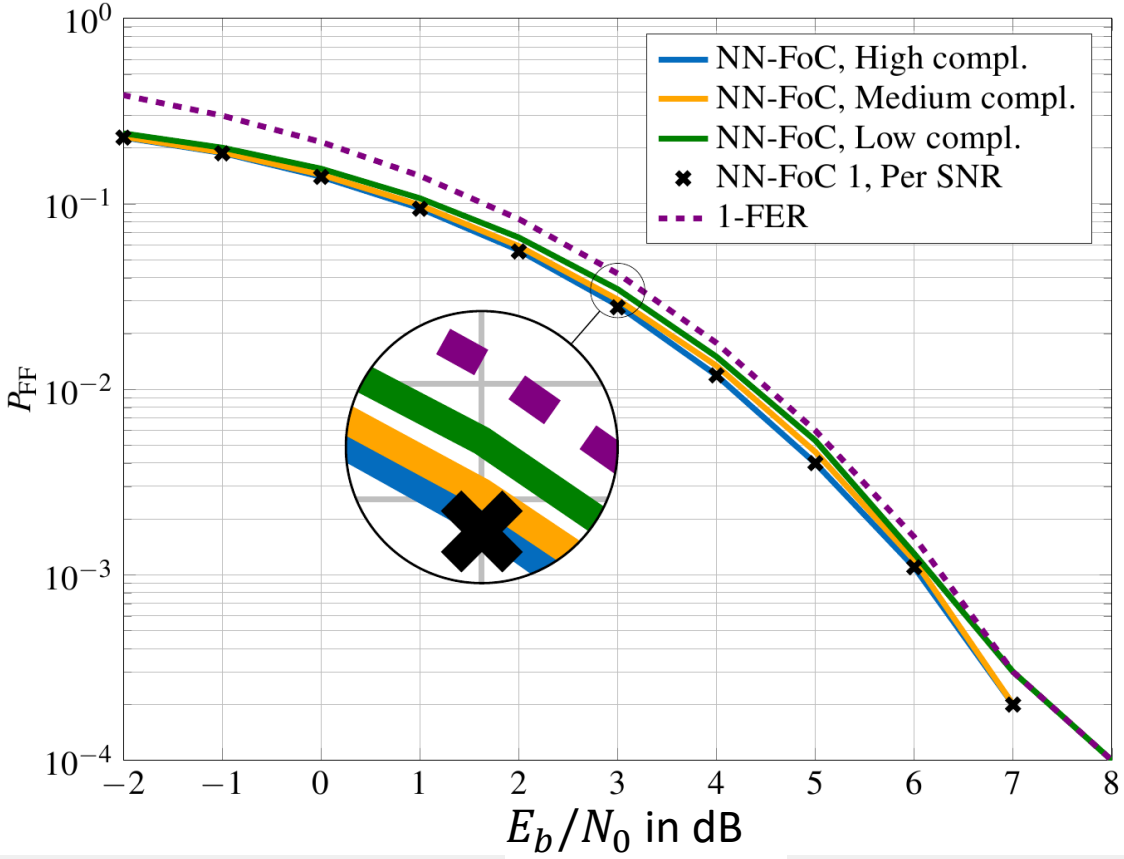
- **Problem**: Iterative decoder introduces complexity and additional latency
  - ARQ asks for re-transmissions in case of decoder failure
- **Idea**: Forecast decoder success based on received LLR $\mathbf{L_y}$ without executing the decoder
  - Save computing resources by not running complex decoder if not promising
  - **Early ARQ**: Feedback of (forecasted) NACK immediately initiates re-transmission → reducing latency (e.g., URLLC)
- **NN-based Forecast (NN-FoC) of Decodability [HWD21]**
  - Use NN to determine probability $p(s|\boldsymbol{L_y})$ of decoding packet correctly ($s = 1$) for given receive LLR $\boldsymbol{L_y}$
  - Hard decision with threshold α to decide $\hat{s} = Q_\alpha\left(p(s|\boldsymbol{L_y})\right)$
  - Structure: Feedforward NN with input layer of length n, hidden layers with ReLu activation, scalar sigmoid output
  - Training dataset with features $\boldsymbol{L_y}$ and labels $s \in \{0,1\}$ → $(\boldsymbol{L_y}, s)$

[HWD21] Hummert, Wübben, Dekorsy: Neural Network-based Forecasting of Decodability for Early ARQ, ISWCS 2021

Performance Evaluation for different NN-FoC

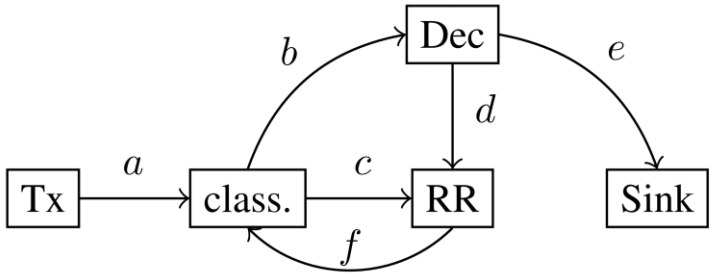
- **System**
  - **F**alse **F**orecasts $P_{\mathrm{FF}} = \frac{\#\text{ of false forecasts}}{\text{all forecasts}}$ vs SNR
  - (7, 4) Hamming code and BP decoder
  - NN-FoC with high, medium, and low complexity
  - SNR-based classifier that forecast all packets to be decodable → $P_{\mathrm{FF}} = 1 - \mathrm{FER}$
  - Benchmark: High complex NN-FoC trained per SNR
- **Observations**
  - NN-FoC trained per SNR outperforms FER-classifier
  - All NN-FoC with range training perform almost as good as the NN-FoC with training per SNR
- **Conclusion**
  - Low complex NN-FoC with training over SNR range is sufficient

NN-FoC	High compl.	Mid compl.	Low compl.
Hidden layer	4	4	2
Width of layer	200, 100, 500, 200	50, 50, 50, 20	50,20
# of weights	2.617.901	9.091	1.441



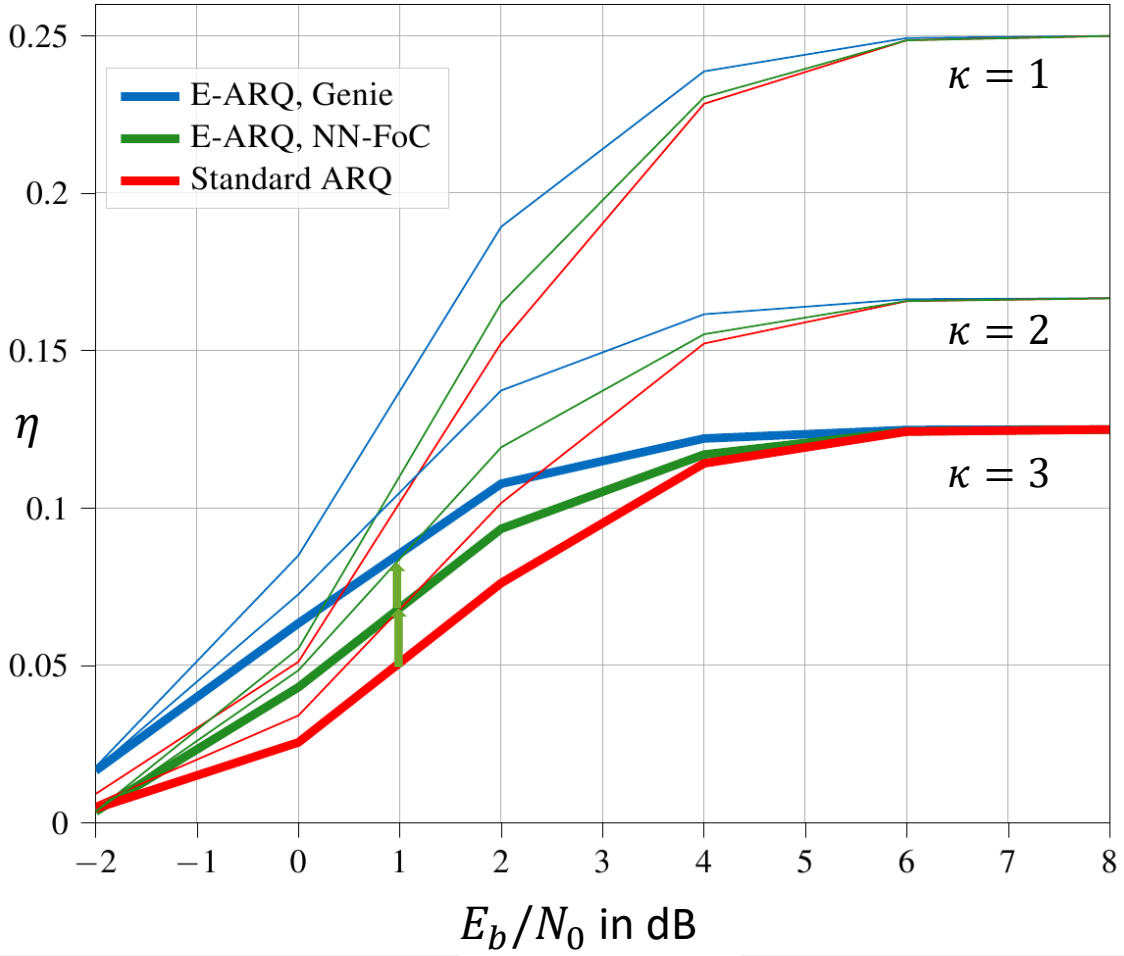
Efficiency Analysis

- Finite state diagram to analyze efficiency η of NN-FoC with E-ARQ and different decoder delays $\kappa = T_{Dec}/T_B$



$$\eta = R_c \frac{T_B}{T_{avg}} = R_c \frac{(1-(1-P_s)-P_{\mathrm{FP}}P_s)^2}{P_{\mathrm{TP}}P_s\big((\kappa+1)P_s+(1-P_s)\big)}$$

- **System**
  - For (32, 16) LDPC code comparison of common ARQ and E-ARQ with genie classifier and NN-FoC
- **Conclusion**
  - Efficiency improvements by NN-FoC and E-ARQ in comparison to standard ARQ, gains increase for higher κ



Conclusion

- **Challenge:** FEC decoder is most complex component of receiver chain

- **Machine Learning Scaled Belief Propagation (MLS-BP)**
  - Limit impact of overestimating LLRs for short block length by one trainable scaling factor
  - For short codes performance improvements are achieved for free

- **Discrete Decoder Design**
  - Information Bottleneck Methods provides approach to learn information processing with minimum complexity
  - **LUT-MP** decoder with 3-bit achieves performance close to optimum
  - **REMQ** combines messages with similar meaning, e.g. to apply fixed decoder for general systems

- **NN-based Forecast (NN-FoC)**
  - Forecasting of decodability to save decoding complexity and latency