UNIVERSITÀ
DI PARMA

Department of Engineering and Architecture
Master of Science in Communication Engineering
(Laurea Magistrale)

Exploring Machine Learning Algorithms for Decoding Linear Block Codes

Advisor:
Prof. **Riccardo Raheli**
Co-Advisors:
Prof. **Henry D. Pfister**
Dr. **Christian Häger**
Dr. **Marco Martalò**

Thesis presented by:
Fabrizio Carpi

Parma, October 12, 2018

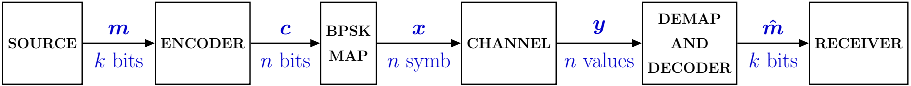
- Thesis done while I was visiting Professor Henry D. Pfister, Duke University, North Carolina (USA).
- Thanks to a 6-months “Overworld” scholarship from Università di Parma.

UNIVERSITÀ DI PARMA

Duke UNIVERSITY

1. Introduction
   - Background
   - Objectives
2. Belief Propagation Decoding Optimization
   - Decoding Scenario
   - Optimization
   - Results
3. Reinforcement Learning for Bit Flipping Decoding
   - Decoding Scenario
   - Learning
   - Results
4. Summary

Linear Block Codes



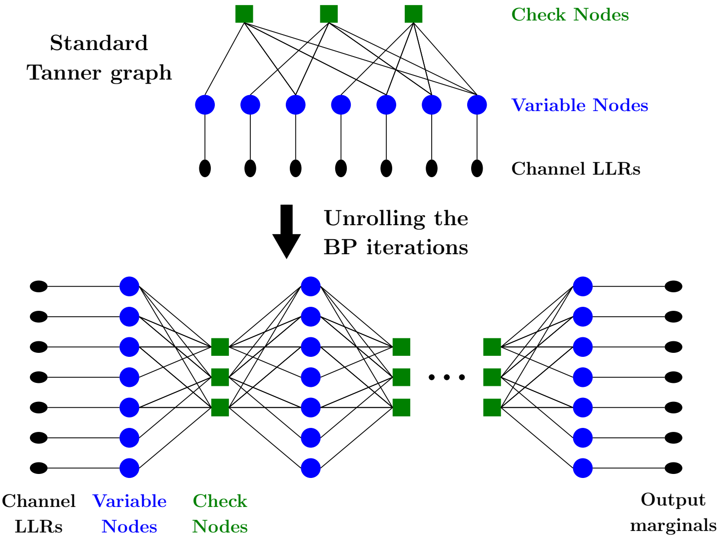
- $\mathcal{C}$ is a linear block code (n, k) described by a Parity Check (PC) matrix $\boldsymbol{H}$ $(m \times n)$.
- We consider Reed–Muller codes with PC matrix $\boldsymbol{H}$ composed of minimum-weight PCs[1].

[1] Elia Santi, Christian Häger, Henry D. Pfister. "Decoding Reed-Muller codes using minimum-weight parity checks", in *Proc. IEEE ISIT* 2018, Vail, Colorado, USA, Jun 2018. arXiv:1804.10319

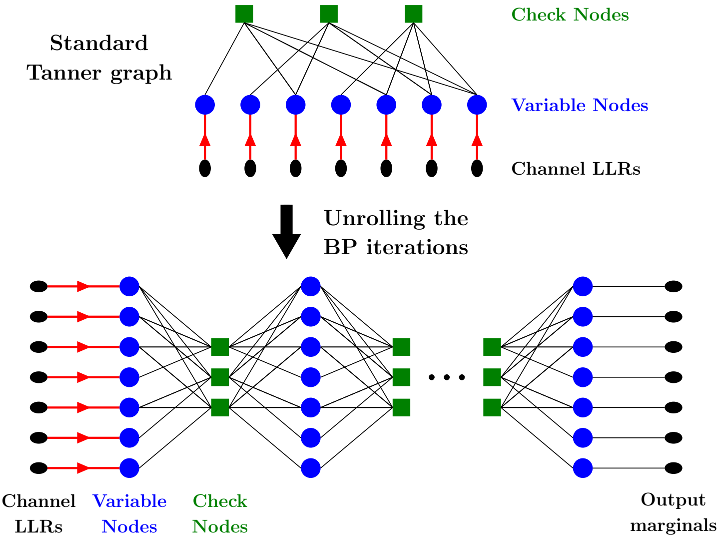
Objectives

- Investigate **machine learning** techniques applied to **communication** problems, in particular the decoding of linear block codes.
  - Belief Propagation (BP) decoding optimization with **supervised learning**.
  - Learning Bit Flipping (BF) decoding with **Reinforcement Learning** (RL).

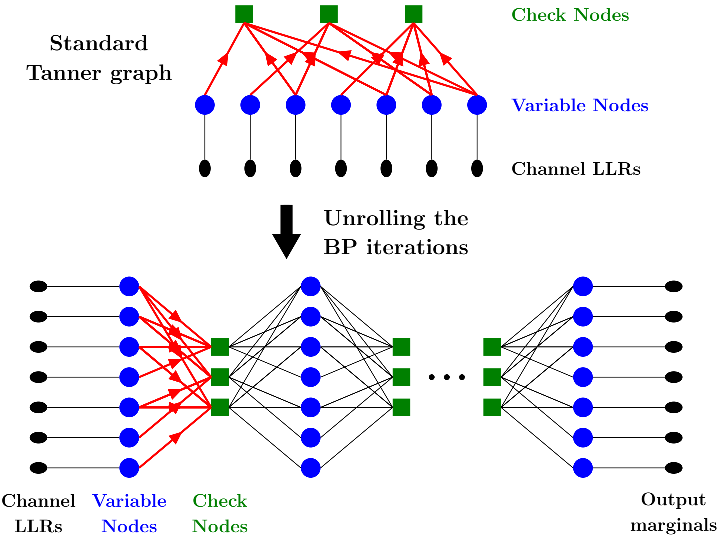
Belief Propagation (BP) Decoding



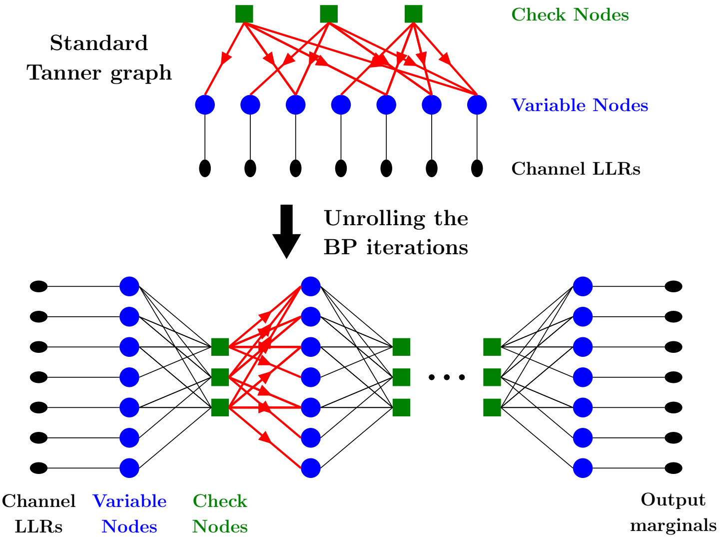
Belief Propagation (BP) Decoding



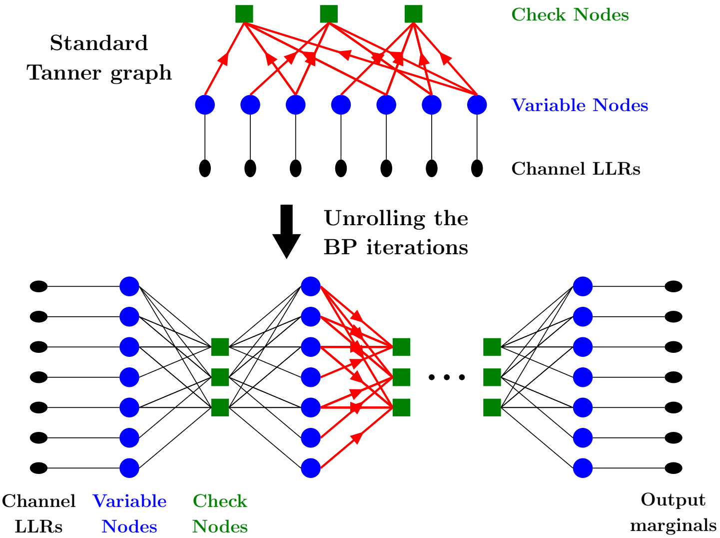
Belief Propagation (BP) Decoding



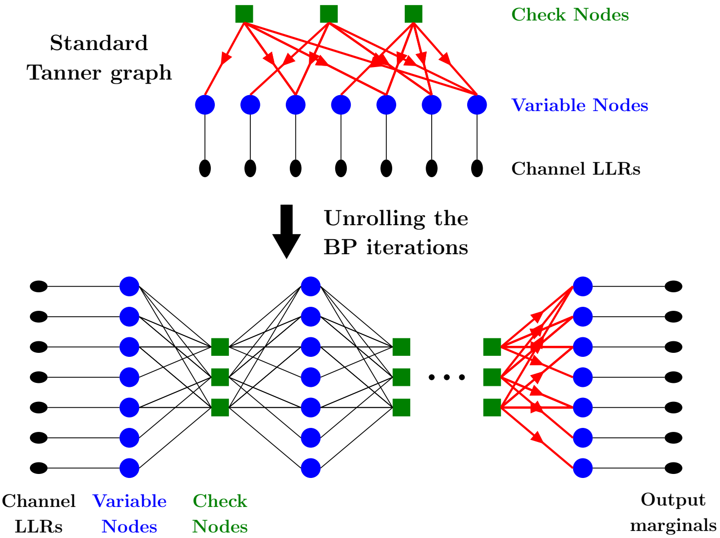
Belief Propagation (BP) Decoding



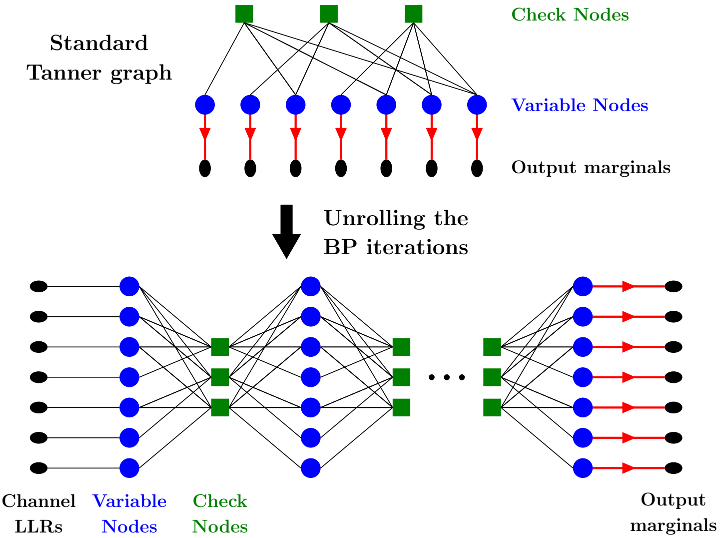
Belief Propagation (BP) Decoding



Belief Propagation (BP) Decoding

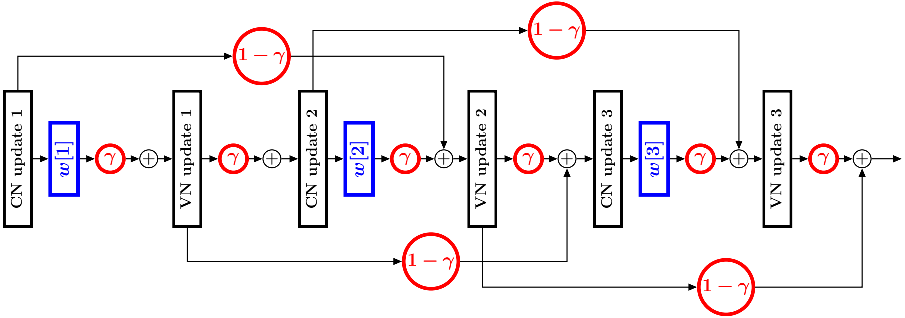


Belief Propagation (BP) Decoding



Parameterized BP

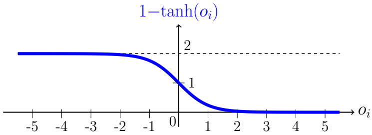
- Interpret the unrolled BP iterations as a deep neural network[2].
- Introduce the following parameters to be optimized: iteration **weights** and **damping** coefficient.

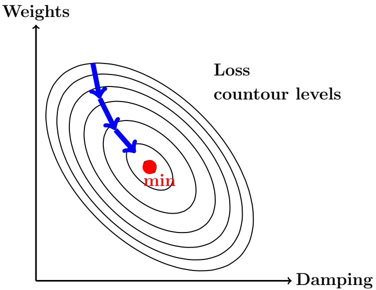


[2] E. Nachmani, Y. Be'ery, D. Burshtein. "Learning to decode linear codes using deep learning", in *Proc. 2016 54th Annual Allerton Conference on Communication, Control, and Computing*, Sep 2016. arXiv:1607.04793

Training

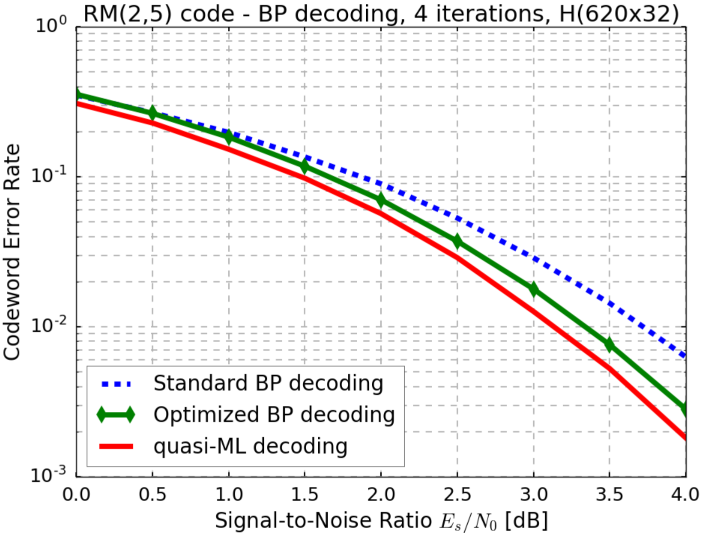
- Always transmit all-zero codeword $\rightarrow$ simple training set.
- Loss function: $\ell(\boldsymbol{o}) = \sum_i (1 - \tanh(o_i))$, where $\boldsymbol{o}$ is the vector of output marginal LLRs. $\ell(\boldsymbol{o})$ is proportional to the bit error rate.[3]
- Stochastic Gradient Descent (SGD) to minimize $\ell(\boldsymbol{o})$ with respect to weights and damping.





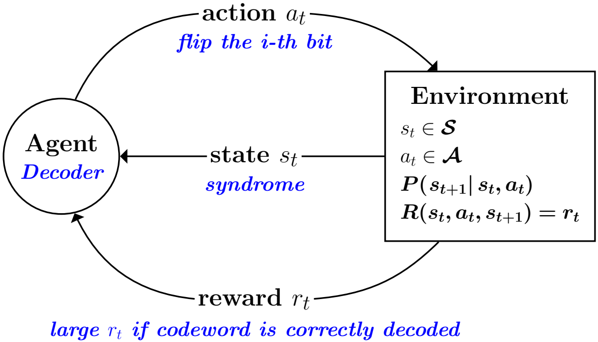
[3] Assume BPSK modulation and Binary-Input AWGN channel.

Results


RM(2,5) code - BP decoding, 4 iterations, H(620x32)
Codeword Error Rate
10^0
10^{-1}
10^{-2}
10^{-3}
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Signal-to-Noise Ratio E_s/N_0 [dB]
Standard BP decoding
Optimized BP decoding
quasi-ML decoding

MDP for Bit Flipping Decoding

- Syndrome: $\boldsymbol{s} = \boldsymbol{H}\,\boldsymbol{z}$, where $\boldsymbol{z} \in \mathbb{F}_2^n$ is the received word.[4] Any codeword $\boldsymbol{c} \in \mathcal{C}$ satisfies $\boldsymbol{H}\,\boldsymbol{c} = \boldsymbol{0}$.
- Standard BF decoding: flip the bit that solves most PCs.
- We approach BF as a Markov Decision Process (MDP).

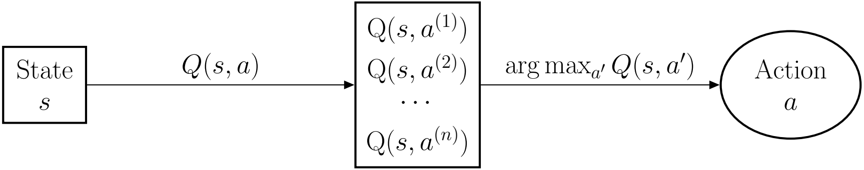


[4] Consider the Binary Symmetric Channel.

Q-Learning

- The goal is to find the optimal policy $\pi^* : \mathcal{S} \to \mathcal{A}$ that maximizes the expected cumulative reward.
- **Solution**: Q-Learning. Use the Q-function to build the action-values for every state

$$Q(s_t, a_t) \simeq r_t + \gamma \underbrace{\max_{a'} Q(s_{t+1}, a')}_{\text{max expected reward for the next state } s_{t+1}}$$

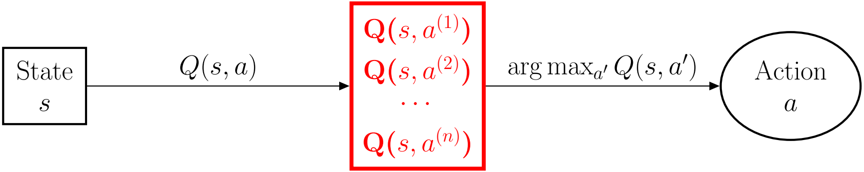


Q-Learning

- The goal is to find the optimal policy $\pi^* : \mathcal{S} \to \mathcal{A}$ that maximizes the expected cumulative reward.
- **Solution**: Q-Learning. Use the Q-function to build the action-values for every state

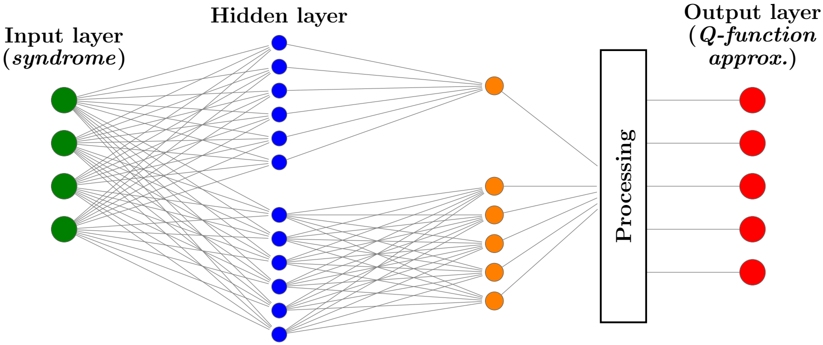
$$Q(s_t, a_t) \simeq r_t + \gamma \underbrace{\max_{a'} Q(s_{t+1}, a')}$$

max expected reward
for the next state s_{t+1}



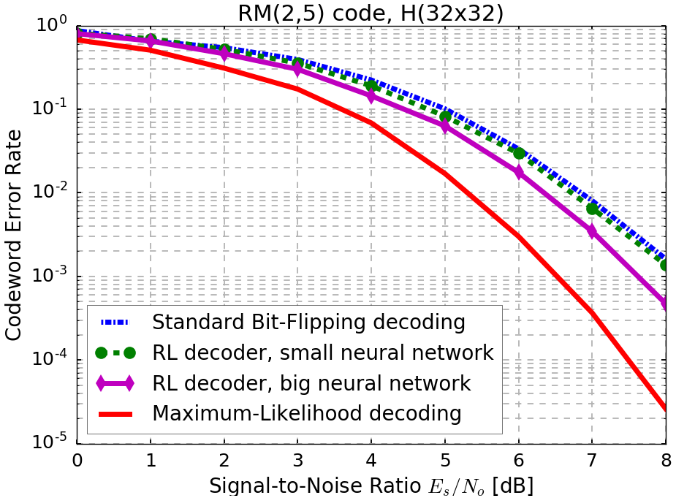
Neural Network Model

- Use a Neural Network (NN) to approximate the Q-values, with Dueling architecture[5].



[5] Z. Wang, T. Schaul, M. Hessel *et al.* "Dueling network architectures for deep reinforcement learning", in *Proc. 33rd International Conference on Machine Learning*, New York, USA, Jun 2016. arXiv:1511.06581

Results


RM(2,5) code, H(32x32)
Codeword Error Rate
Signal-to-Noise Ratio E_s/N_o [dB]
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
0
1
2
3
4
5
6
7
8
Standard Bit-Flipping decoding
RL decoder, small neural network
RL decoder, big neural network
Maximum-Likelihood decoding

Summary

- BP optimization with **supervised learning**.
  - Code Tanner graph $\approx$ deep NN.
  - Transmit all-zero codeword $\rightarrow$ simple training set and loss.
  - ✓ optimized coefficients for the BP graph.
- **Reinforcement learning** for BF decoding.
  - MDP framework for the BF decoding.
  - The agent starts with no prior knowledge about the code and the MDP. The code is intrinsic in the MDP.
  - NN allows the agent to have a higher-dimensional representation of the syndrome that may outperform standard BF decoding.
- Future work
  - Investigate parameterized BP on other codes.
  - Adapt RL model for BI-AWGN channel.

Thank you!

UNIVERSITÀ DI PARMA

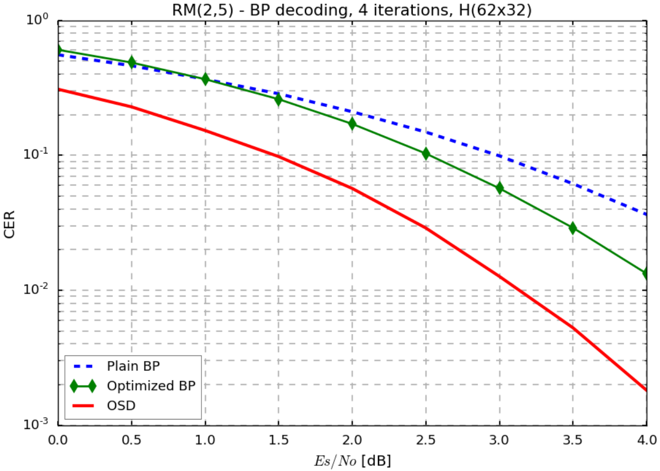
Department of Engineering and Architecture
Master of Science in Communication Engineering
(Laurea Magistrale)

Exploring Machine Learning Algorithms for Decoding Linear Block Codes

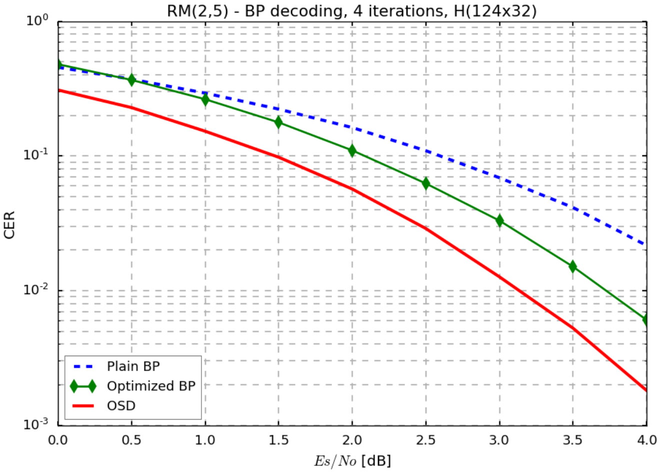
Advisor:
Prof. **Riccardo Raheli**
Co-Advisors:
Prof. **Henry D. Pfister**
Dr. **Christian Häger**
Dr. **Marco Martalò**

Thesis presented by:
Fabrizio Carpi

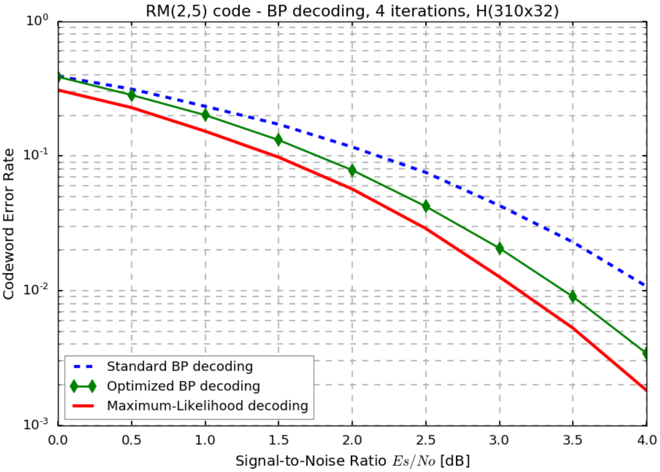
Backup slides I



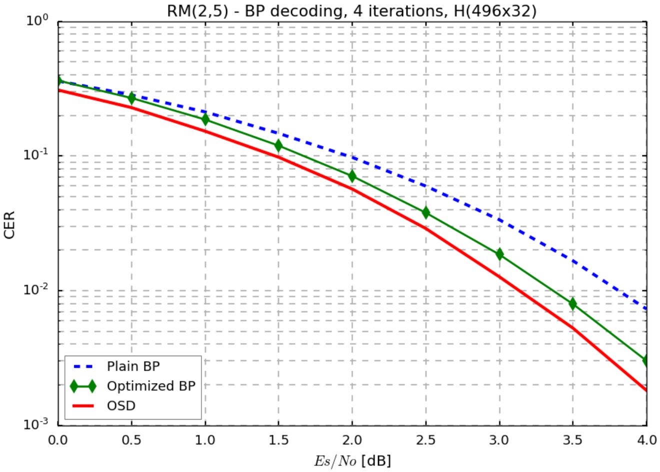
Backup slides II



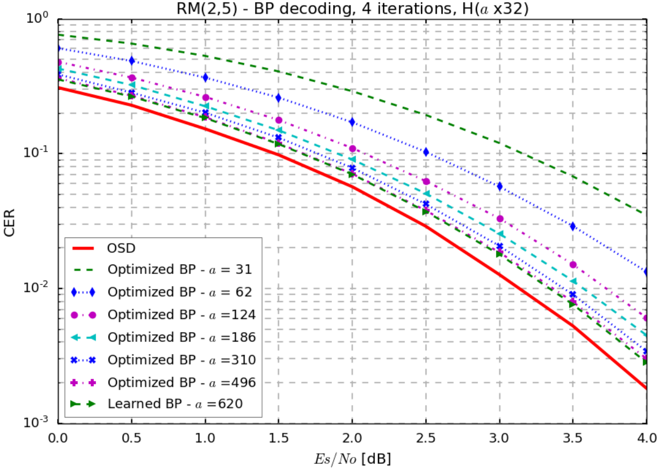
Backup slides III



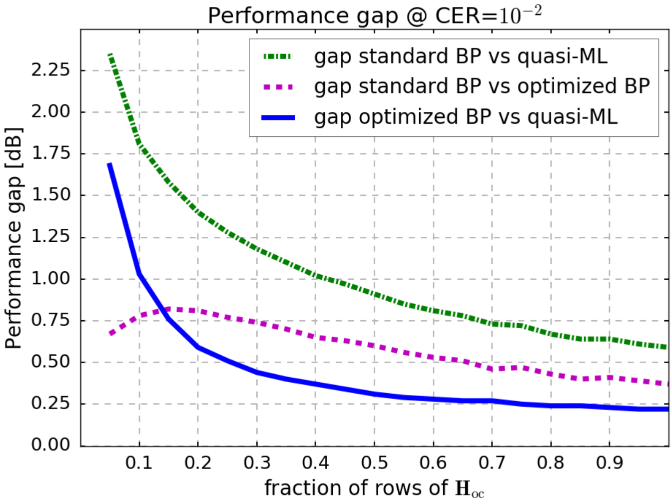
Backup slides IV



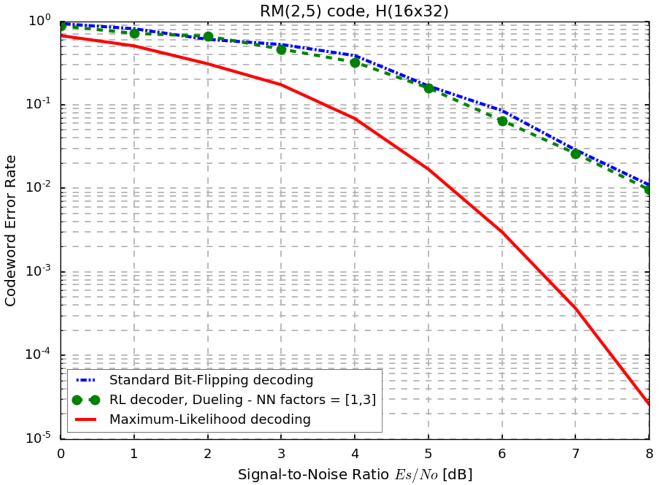
Backup slides V



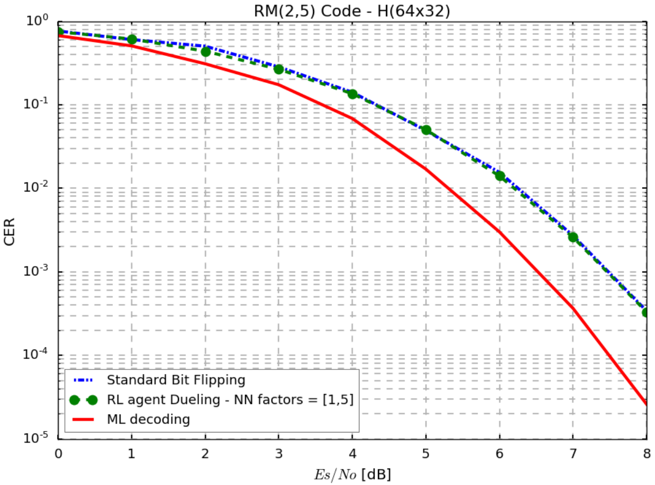
Backup slides VI



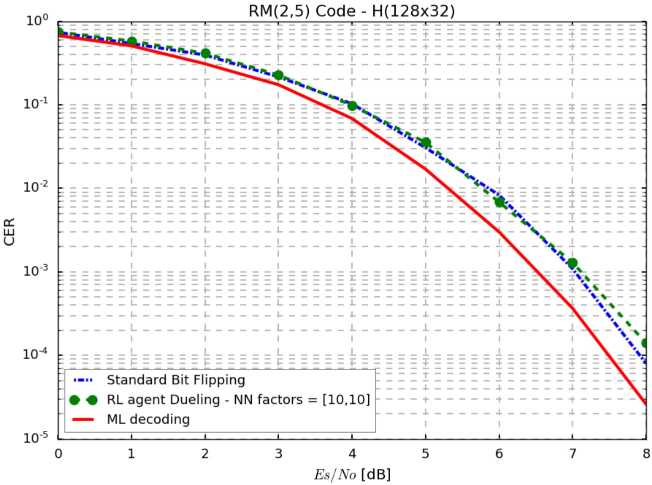
Backup slides VII



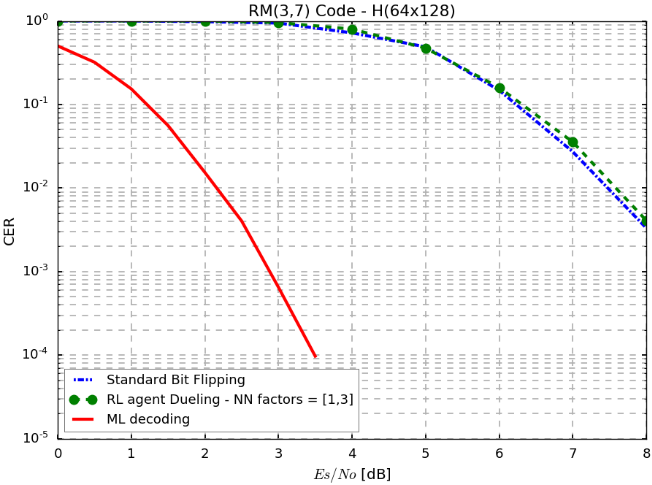
Backup slides VIII



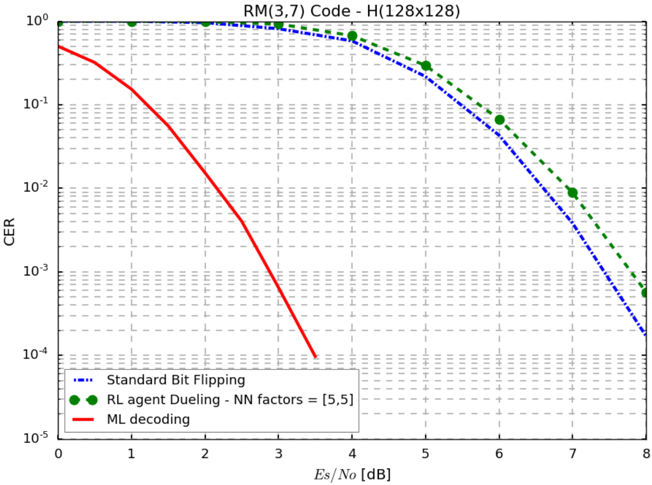
Backup slides IX



Backup slides X



Backup slides XI



Backup slides XII

$\theta^- = \theta_{t-\tau}$
with $0 < \tau < t$

$$\mathbb{E}_{(s_t,a_t,r_t,s_{t+1})}\left[\left(\underbrace{r_t + \gamma \overbrace{\max_{a'} Q\left(s_{t+1}, a'; \theta^-\right)}^{\text{max expected reward for the next state } s_{t+1}}}_{\text{Q-function expression}} - \underbrace{Q\left(s_t, a_t; \theta\right)}_{\text{output from NN}}\right)^2\right].$$