The Max-LUT Method: Mutual-Information Maximizing Lookup Tables

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY
1990

Brian M. Kurkoski

Japan Advanced Institute of Science and Technology

June 22, 2017

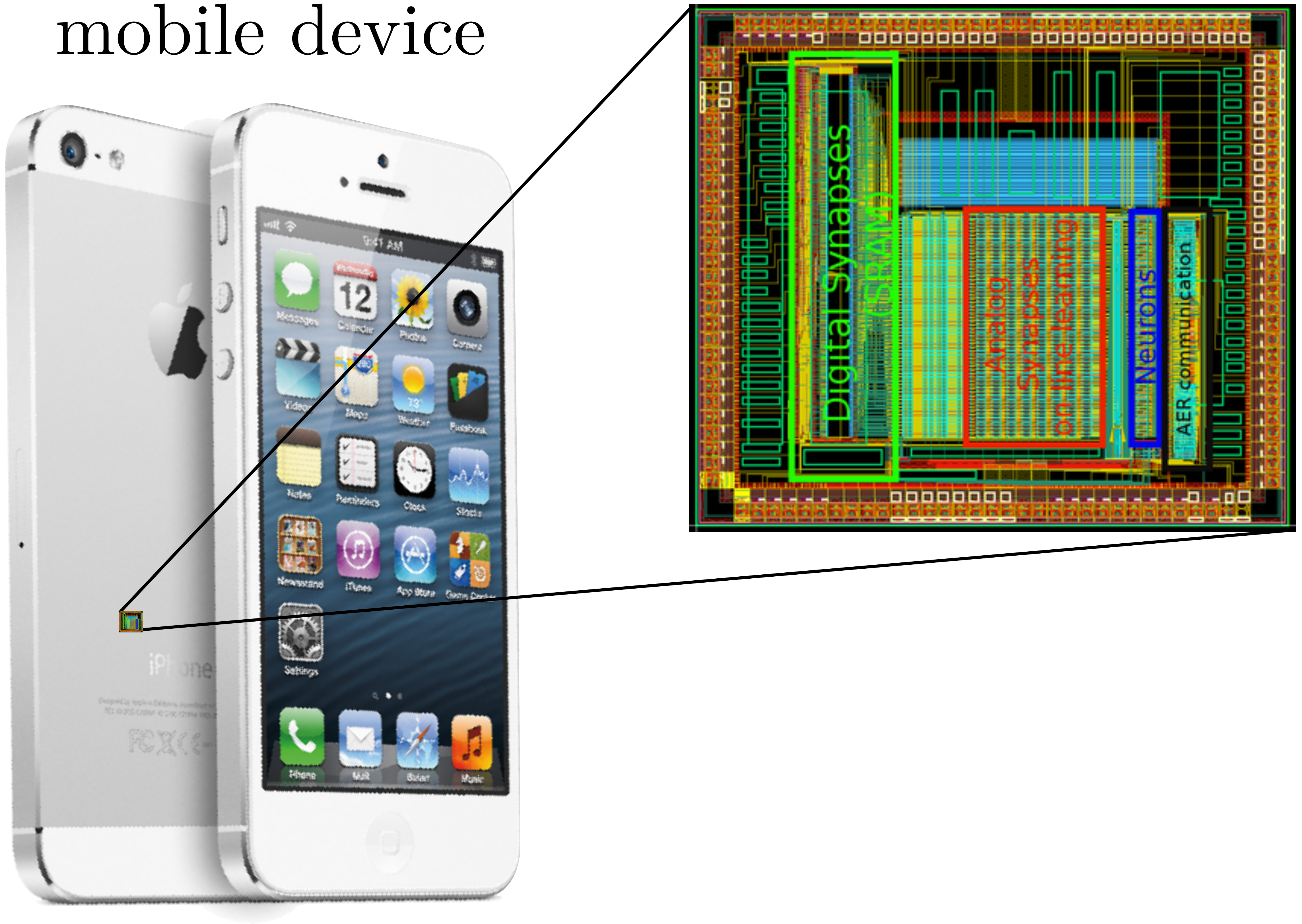
Asia-European Workshop on Information Theory

Boppard, Germany

Hardware-Aware Information Theory

mobile device

VLSI



LDPC codes are widely used.

In communications:

- 5G, WiFi 802.11n, video broadcasting
- Ethernet over twisted pair

In data storage:

- flash memories and SSD drives

Numerous ECC decoding algorithms:

- implemented in VLSI
- "ad hoc" implementation by engineers

Can we give a theoretical foundation to implementations?

Conventional Algorithm Design Process

New Communications Problem

Human that designs an algorithm

Decoding Algorithm

```
input: y, H
1. Compute s = Hy
2 …
```

From Human-Designed Decoding Algorithms, to Machine-Assisted Ones

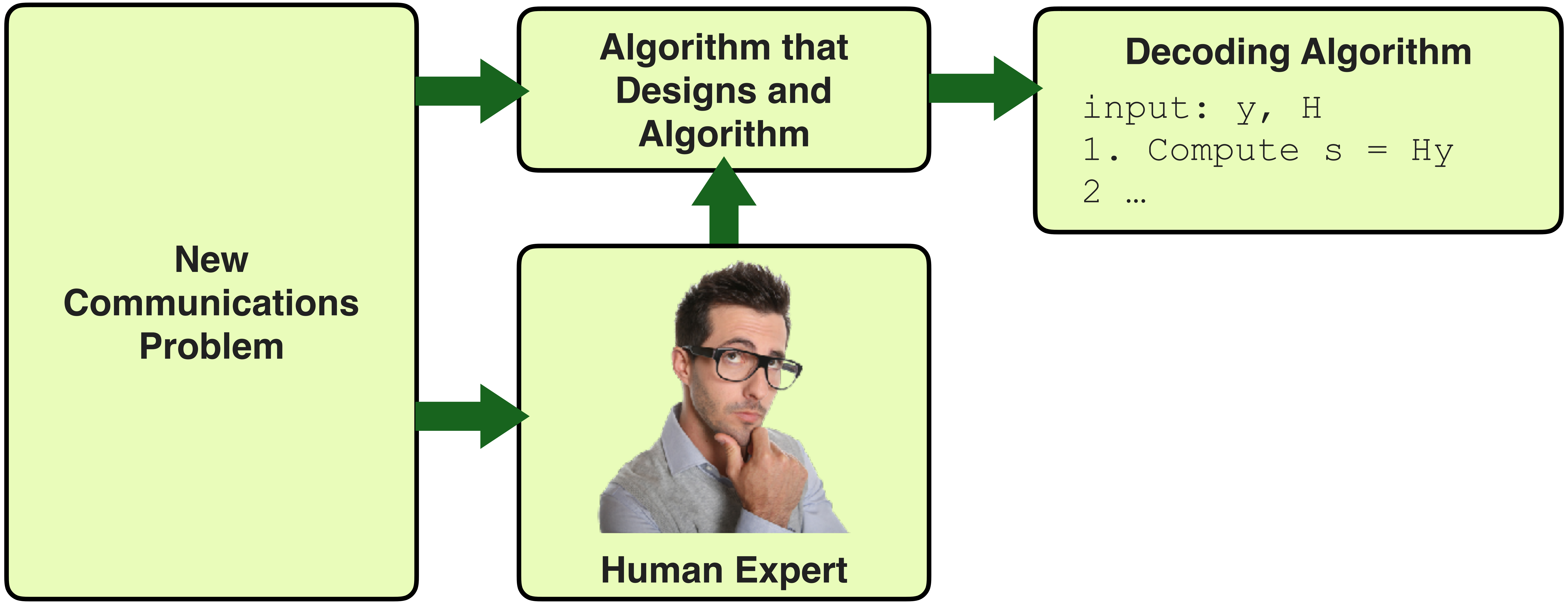
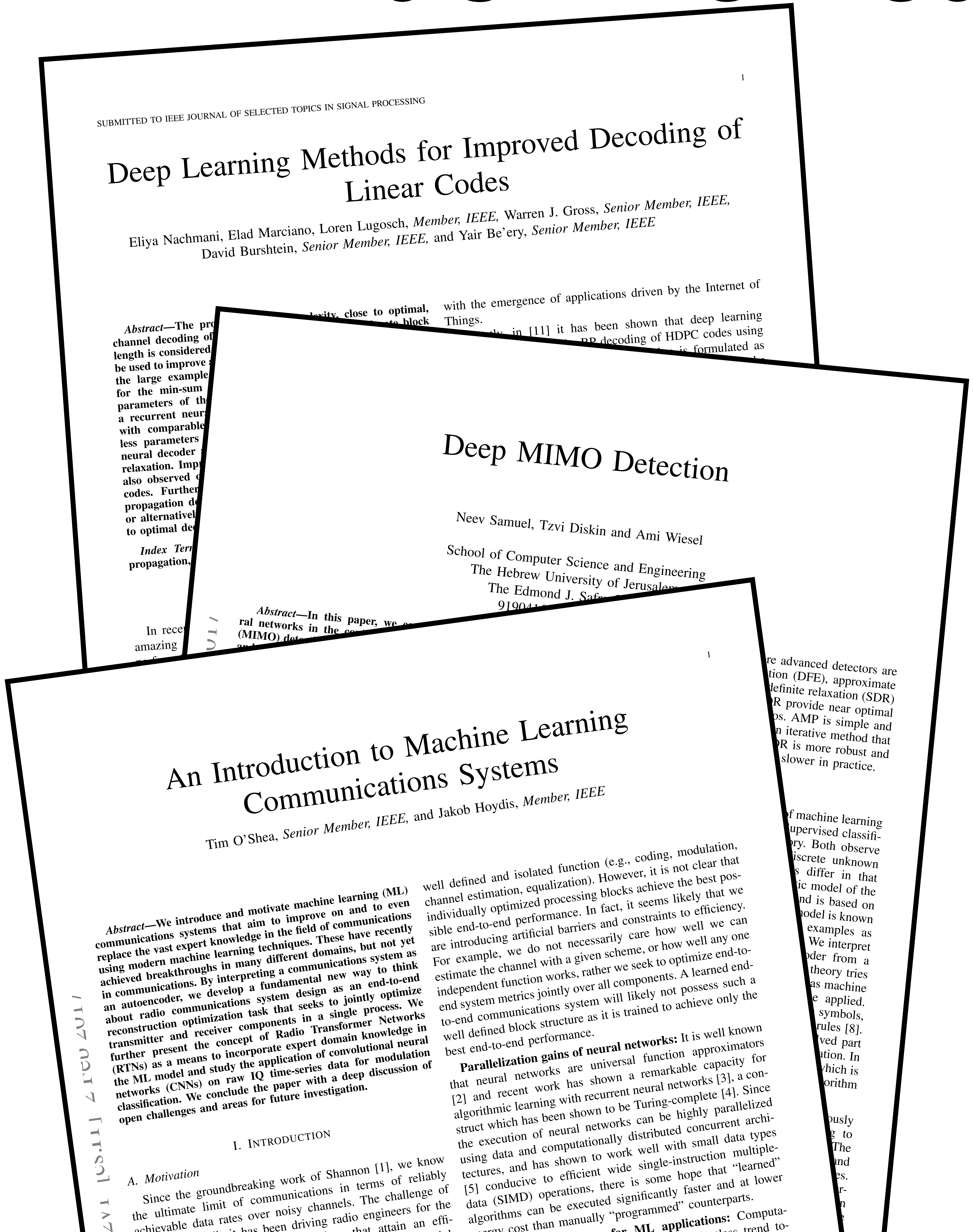


image: https://www.mindsonar.info http://bit.ly/2nbSXrp

Machine Learning for Coding



Final Progarm | Ad-hoc Meeting on Deep Learning & Coding | TAU, Israel, June 19, 2017

JUMP TO OTHER IT SOCIETY WEBSITES:

Jump to child site

Ad-hoc Meeting on Deep Learning & Coding, TAU, Israel, June 19, 2017

Ad-hoc Meeting on Deep Learning & Coding will take place at Tel Aviv University, Israel, on Monday, June 19, 2017.

Location: Room 011, Classrooms (Kitot) Building, Faculty of Engineering - see attached map & link.

The program below includes talks that will review and update on the current research covered by papers published during the last year on Deep Learning & Coding - see link below for **Abstracts & References.**

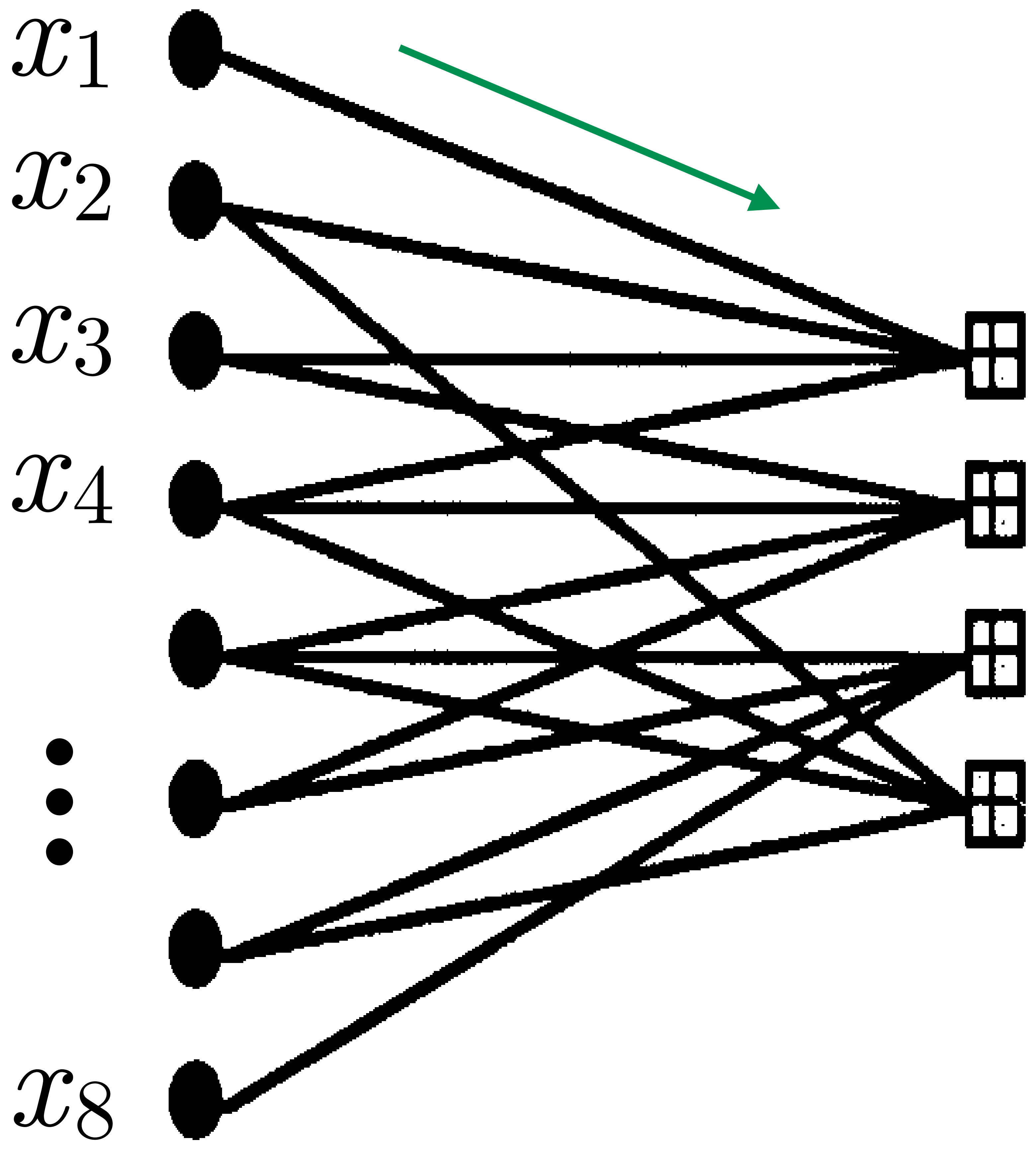
These talks are free and open to all interested faculty & students, as well as for participants from research institutions and industry.

Problem Setting

Design fixed precision LDPC decoders:

- Everything is discrete: channel, messages, nodes
- Nodes implement discrete mappings

  > find these decoding mappings by maximizing mutual information



Outline

1 Factor graphs and Conventional VLSI Implementation

- Factor graphs
- Quantization of factor graph messages

2 Max-LUT Method

- Quantization of factor graph messages
- Max-LUT method

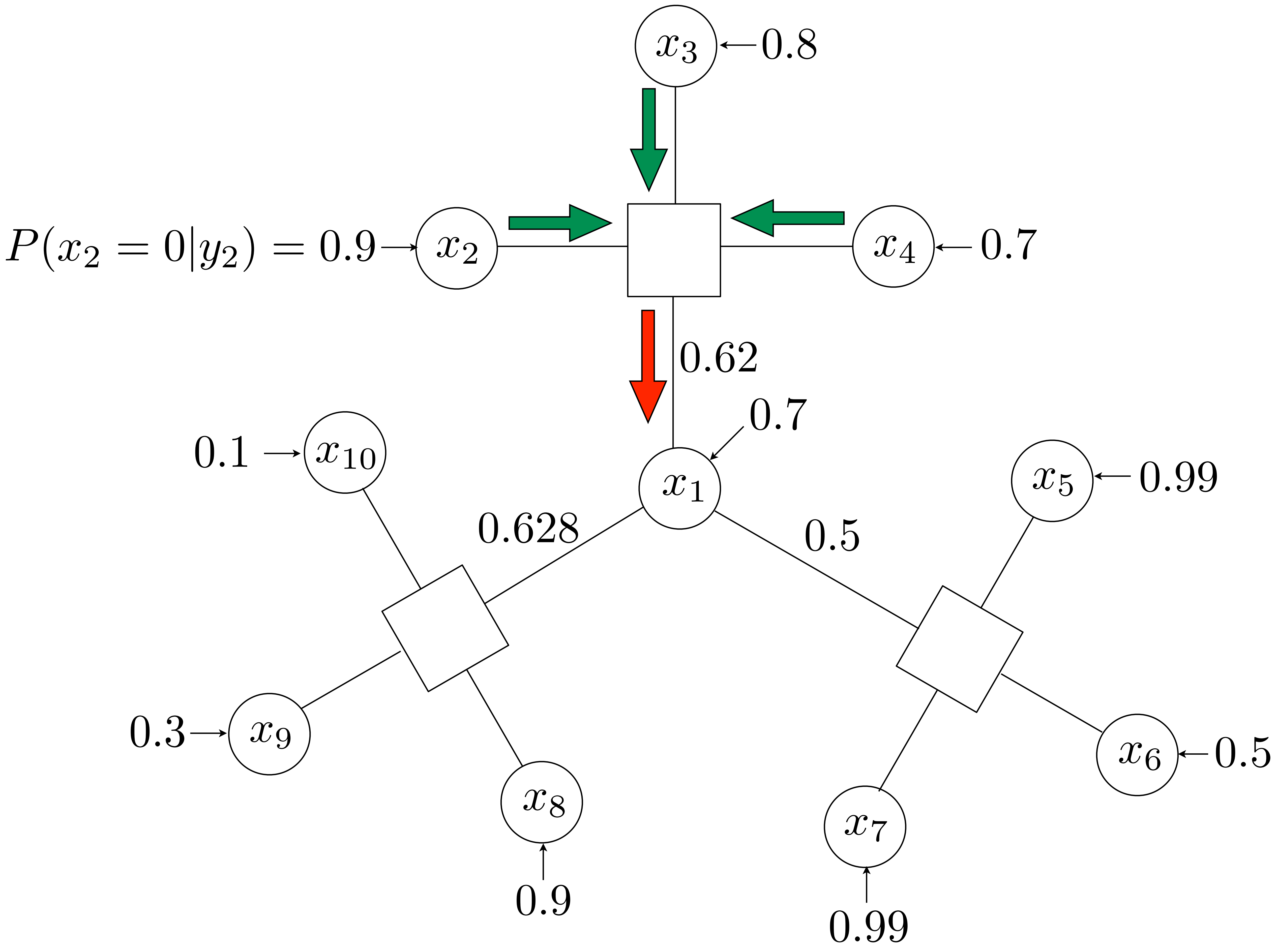
3 Application to LDPC code decoding

- Numerical results: 4 bits/message “performs like floating point”

4 Discussion

1 Factor Graph Representation of Decoders

input:
from channel



output:
transmitted
information
estimate

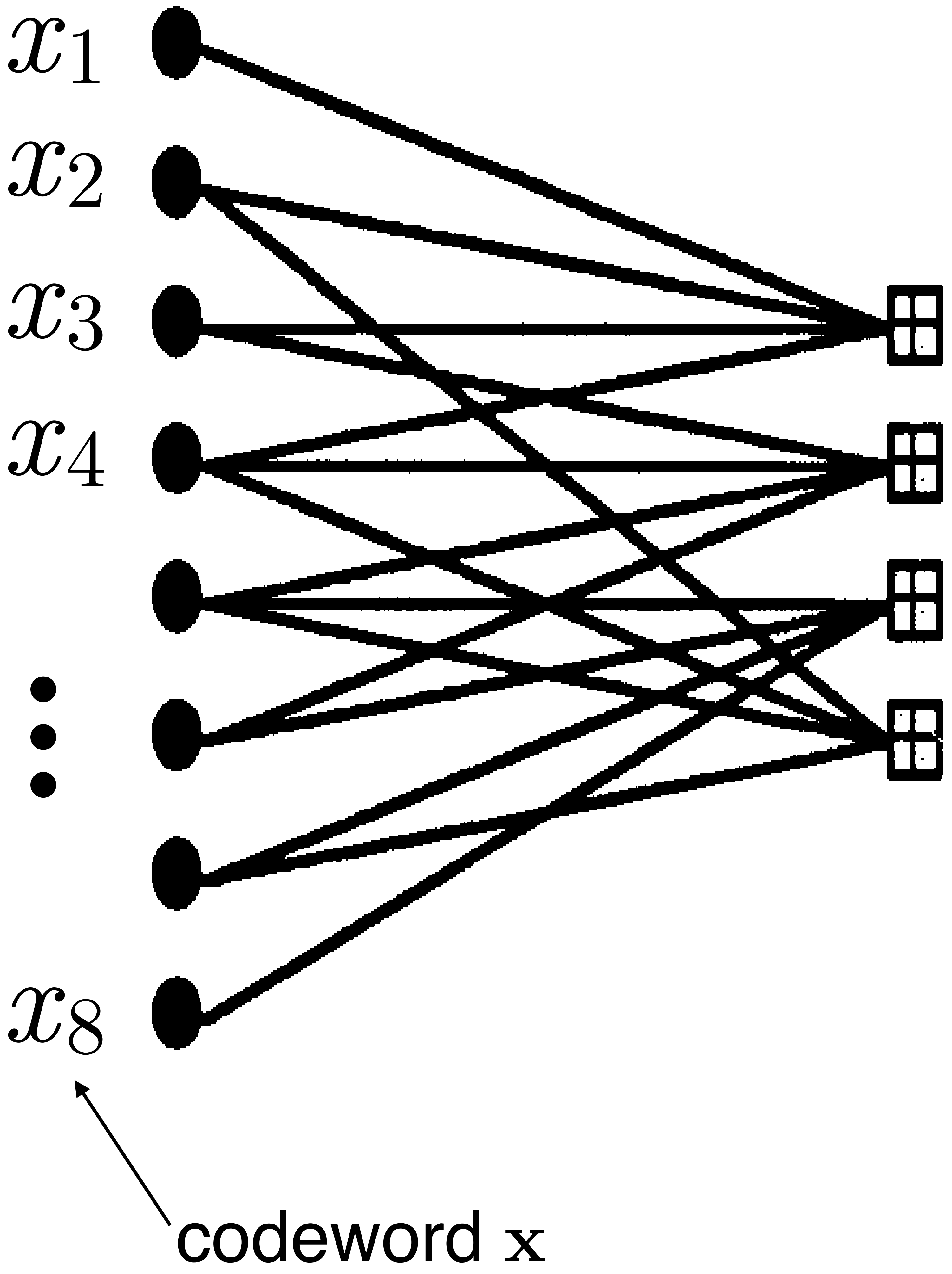
“messages” are just numbers, “nodes” compute functions

Factor Graphs: Local Functions

Variable node
local function

"equality" restriction

$x_4 = x_4 = x_4$



Check node
local function

$x_1 + x_2 + x_3 + x_4 = 0$

or

$f(x_1, x_2, x_3) = x_4$

Factor Graphs: Update Rules

Variable node update rule:

$$L_v = Y_v + \sum_{i \in \mathcal{N}(v) \backslash v} V_i$$

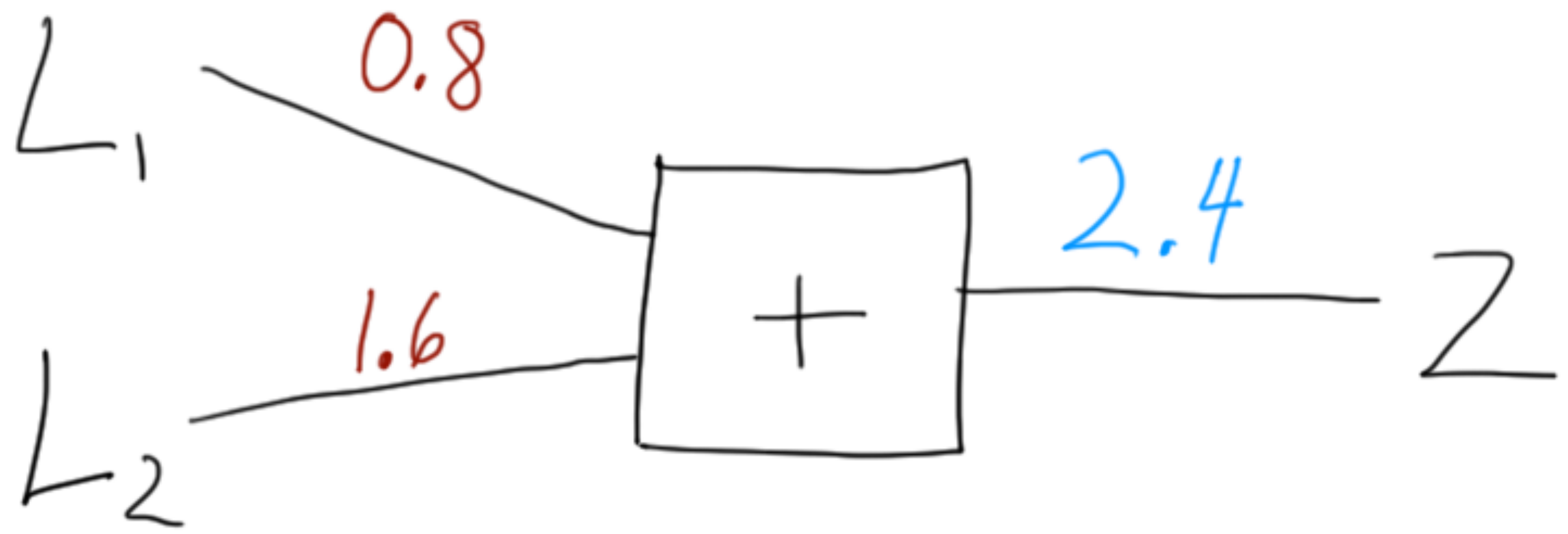
Check node update rule:

$$R_c = -2 \tanh^{-1} \Big(\prod_{i \in \mathcal{M}(c) \backslash c} \tanh \big(-\frac{L_i}{2} \big) \Big)$$

Final output

$$\widehat{L}_v = Y_v + \sum_{i \in \mathcal{N}(v)} V_i$$

Quantization in VLSI Receivers

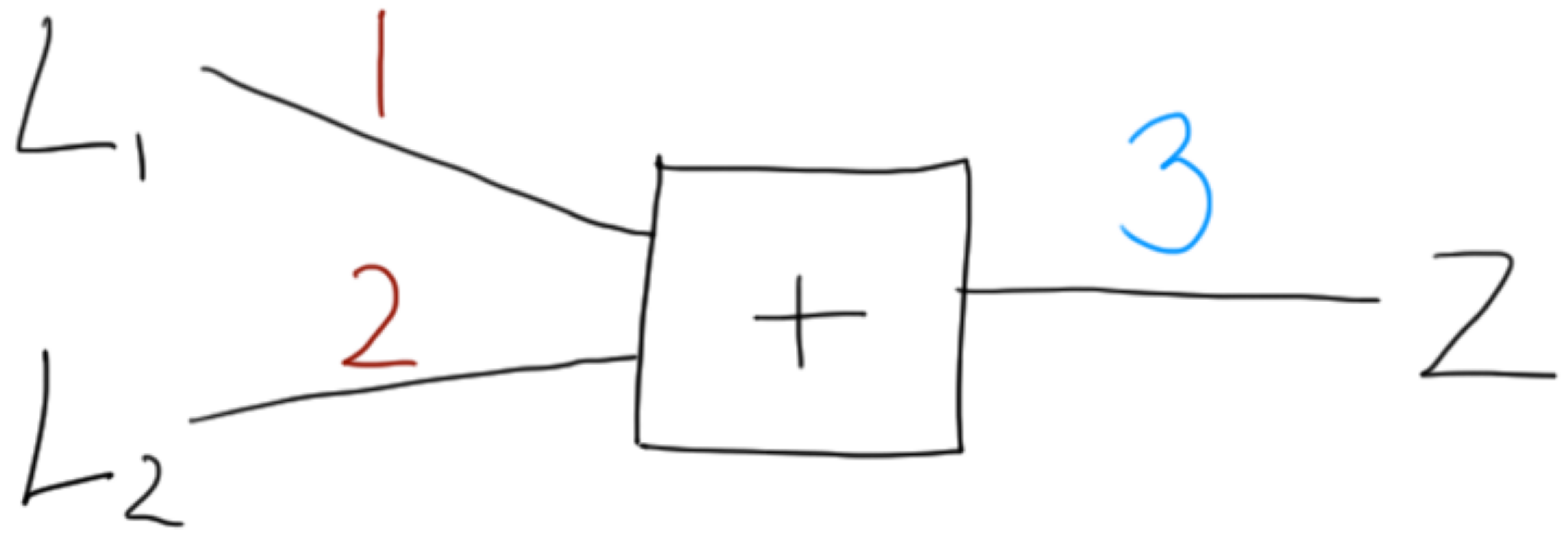


Desired: efficient quantization schemes

VLSI quantization schemes are chosen ad hoc way by engineers

Quantization in VLSI Receivers

L_1, Z are "messages" or LLRs

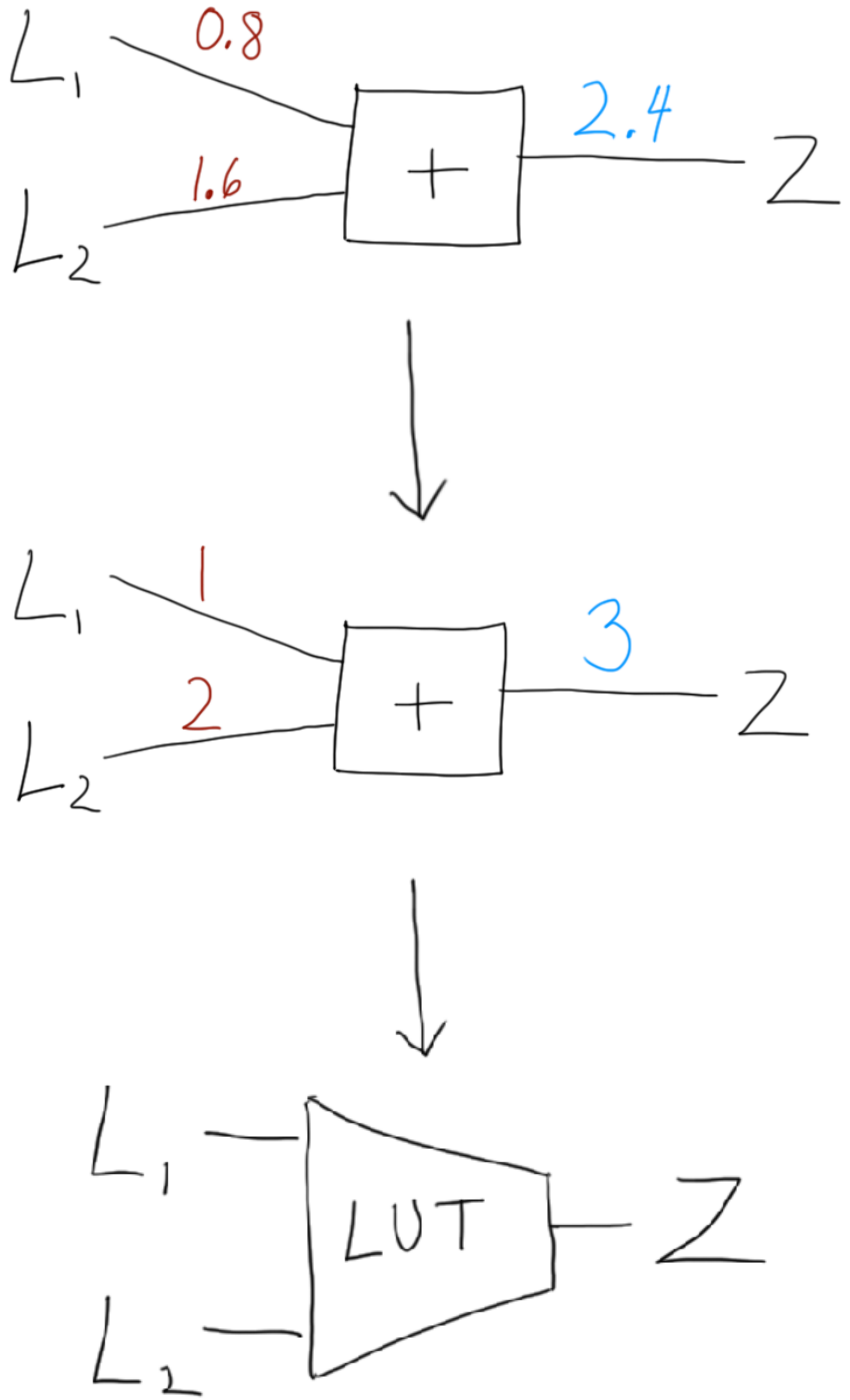


Desired: efficient quantization schemes

VLSI quantization schemes are chosen ad hoc way by engineers

2 Max-LUT Method

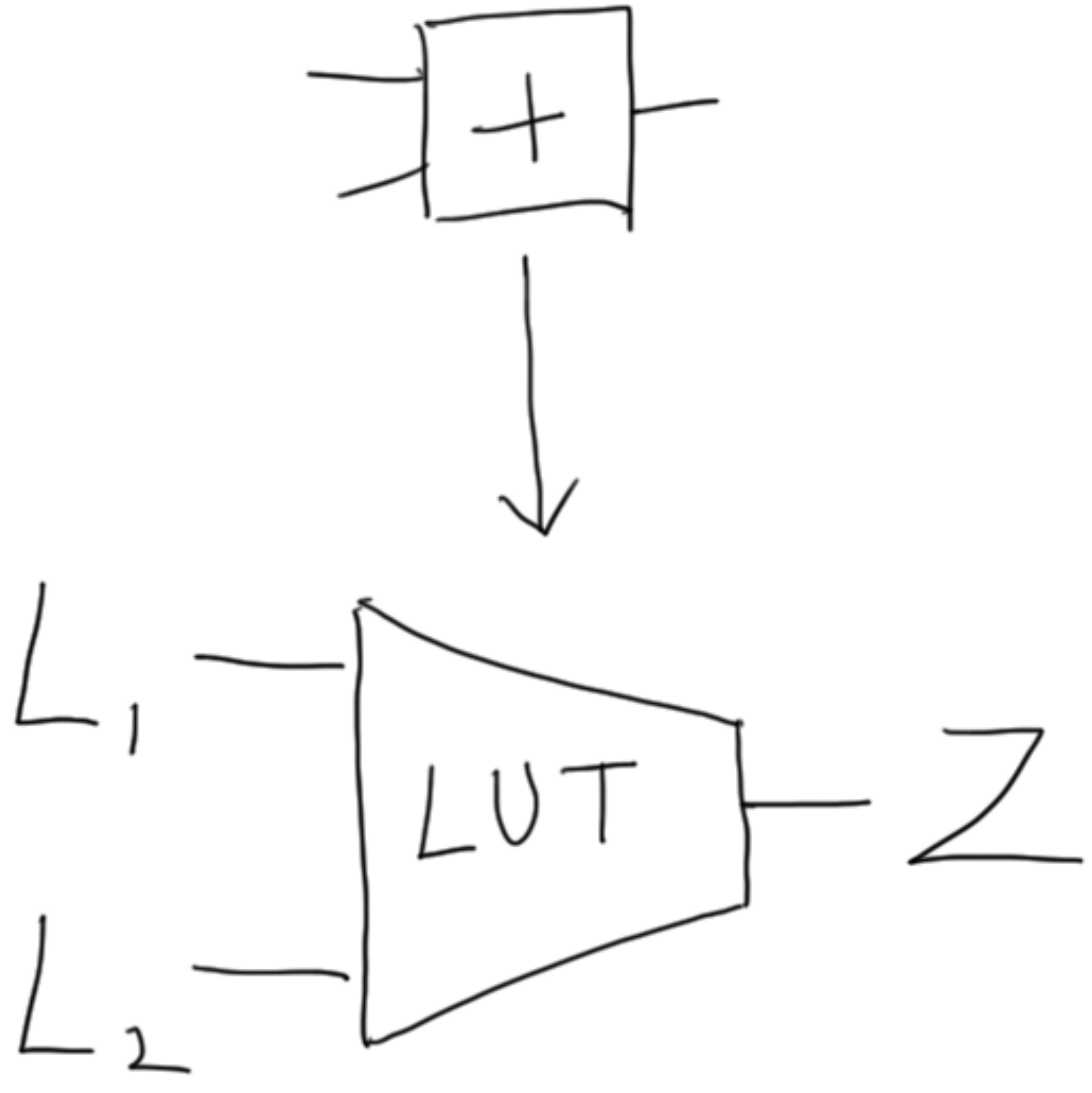
Max-LUT is a method for implementing the node decoder functions for graph-based decoders, using lookup tables that maximize mutual information.



Characteristics of the Max-LUT Method

- We need a factor graph
- We need an input distribution
- Factor graph messages are discrete
- Decoding functions are look up tables (LUT)
- Lookup tables are designed to maximize mutual information

Lookup Table (LUT) Implementation

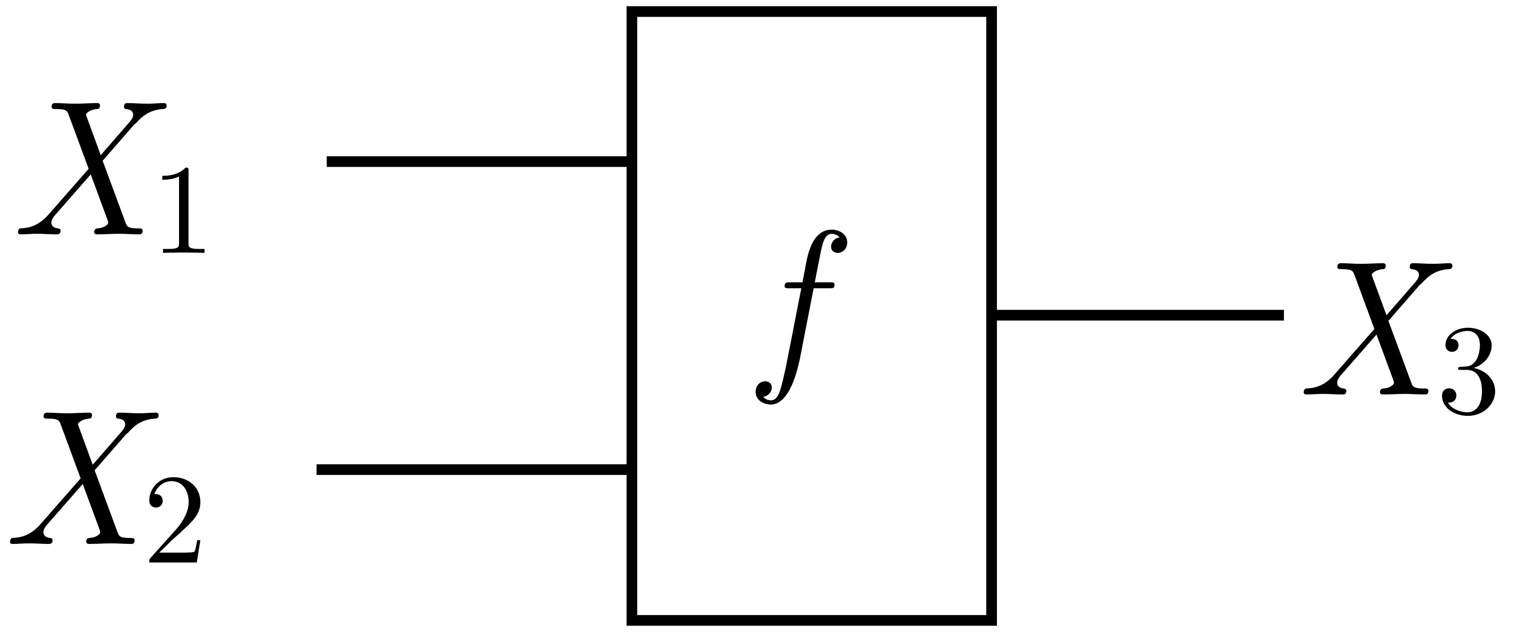


Small LUTs are easily implemented in VLSI

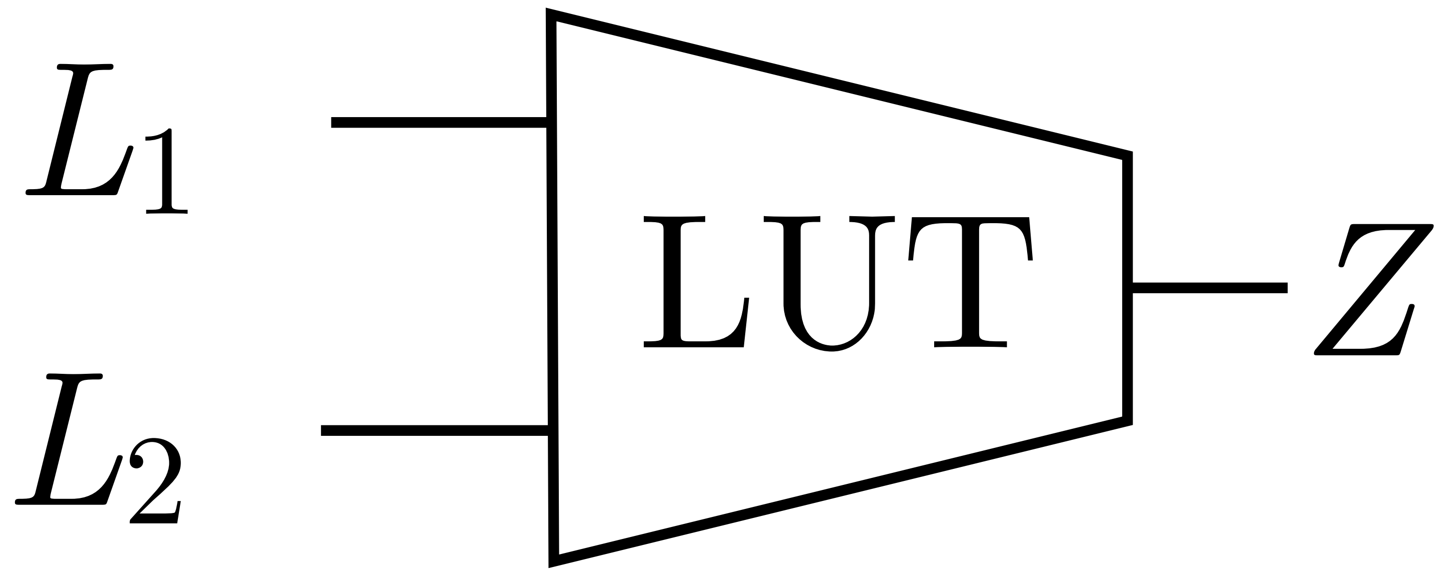
L_1 \ L_2	…	−2	−1	0	1	2	…
−4	…	−4	−4	−4	−3	−2	…
−3		−4	−4	−3	−2	−1	
−2		−4	−3	−2	−1	0	
−1		−3	−2	−1	0	1	
⋮				⋮			
3	…	1	2	3	3	3	…

Max-LUT Method: Central Idea

Encoder Side. Code symbols X



Decoder Side



- Check node $x_3 = x_1 + x_2$
- Var node $x_1 = x_2 = x_3$
- etc.

L_i is a noisy version of X_i ,

Z is a noisy version of X_3

Choose LUT to maximize mutual information

$$\max I\big(\mathsf{X}_3; \mathsf{Z}\big) = \max_{\mathrm{LUT}} I\big(\mathsf{X}_3; \mathrm{LUT}(\mathsf{L}_1, \mathsf{L}_2)\big)$$

Max-LUT Method: Three Steps

- Step 1: Find joint distribution from marginal distributions
- Step 2: Quantize joint distribution maximize mutual information
- Step 3: Find LUT from the quantizer

Example

- LDPC variable node, two inputs $\mathsf{L}_1, \mathsf{L}_2$ with $\Pr(\mathsf{L}_i | \mathsf{X}_i)$
- local constraint: "$x_1 = x_2 = x_3$"
- Goal: find max-MI lookup table $\mathsf{Z} = \mathrm{LUT}(\mathsf{L}_1, \mathsf{L}_2)$

Max-LUT Step 1: Joint Distribution

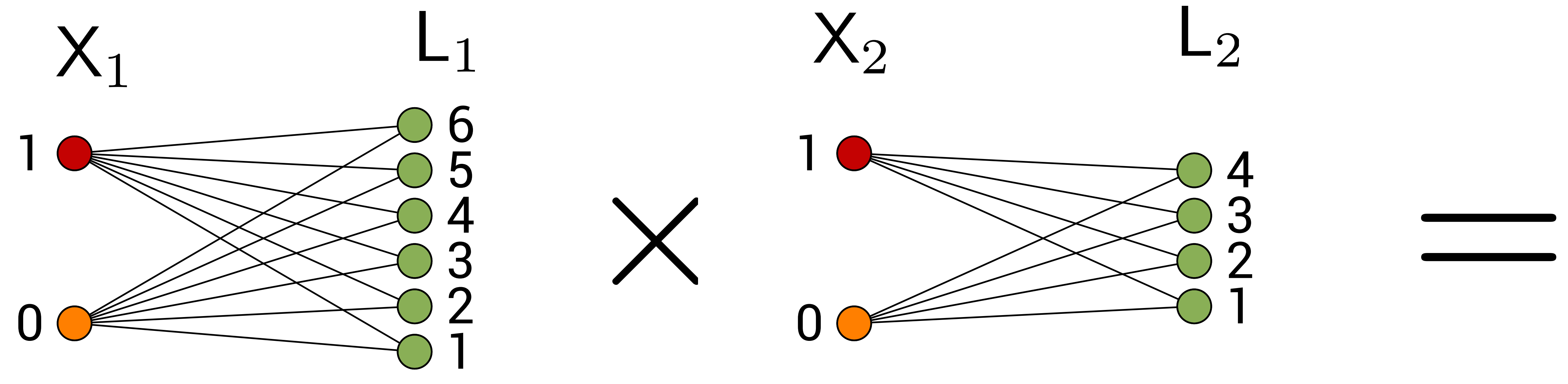
- Step 1: Construct joint distribution

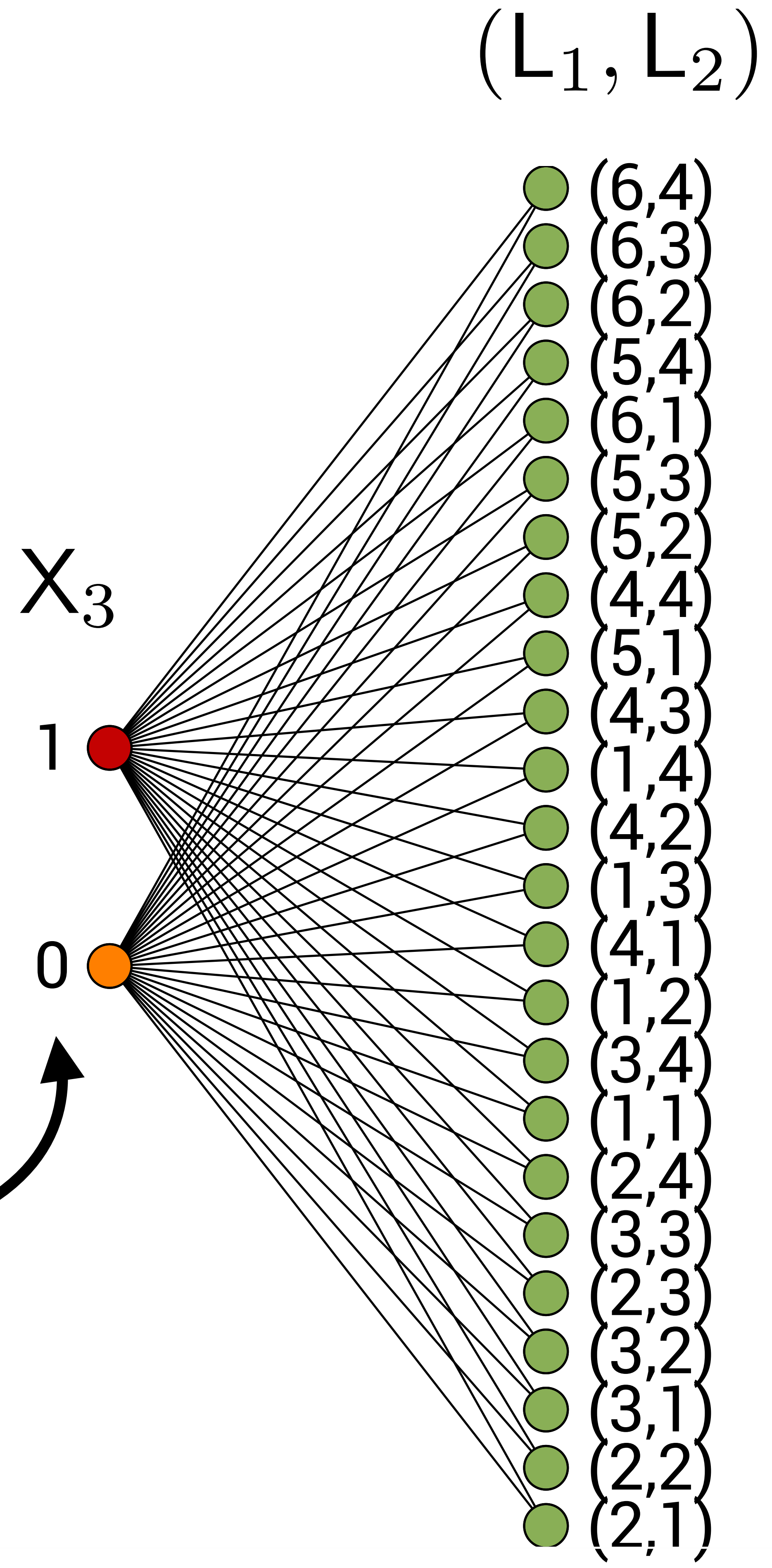
$\Pr(\mathsf{L}_1, \mathsf{L}_2 | \mathsf{X}_3)$

$\Pr(\mathsf{L}_1 | \mathsf{X}_1)$

$\mathsf{L}_1 \in \{1, 2, 3, 4, 5, 6\}$

$\Pr(\mathsf{L}_2 | \mathsf{X}_2)$



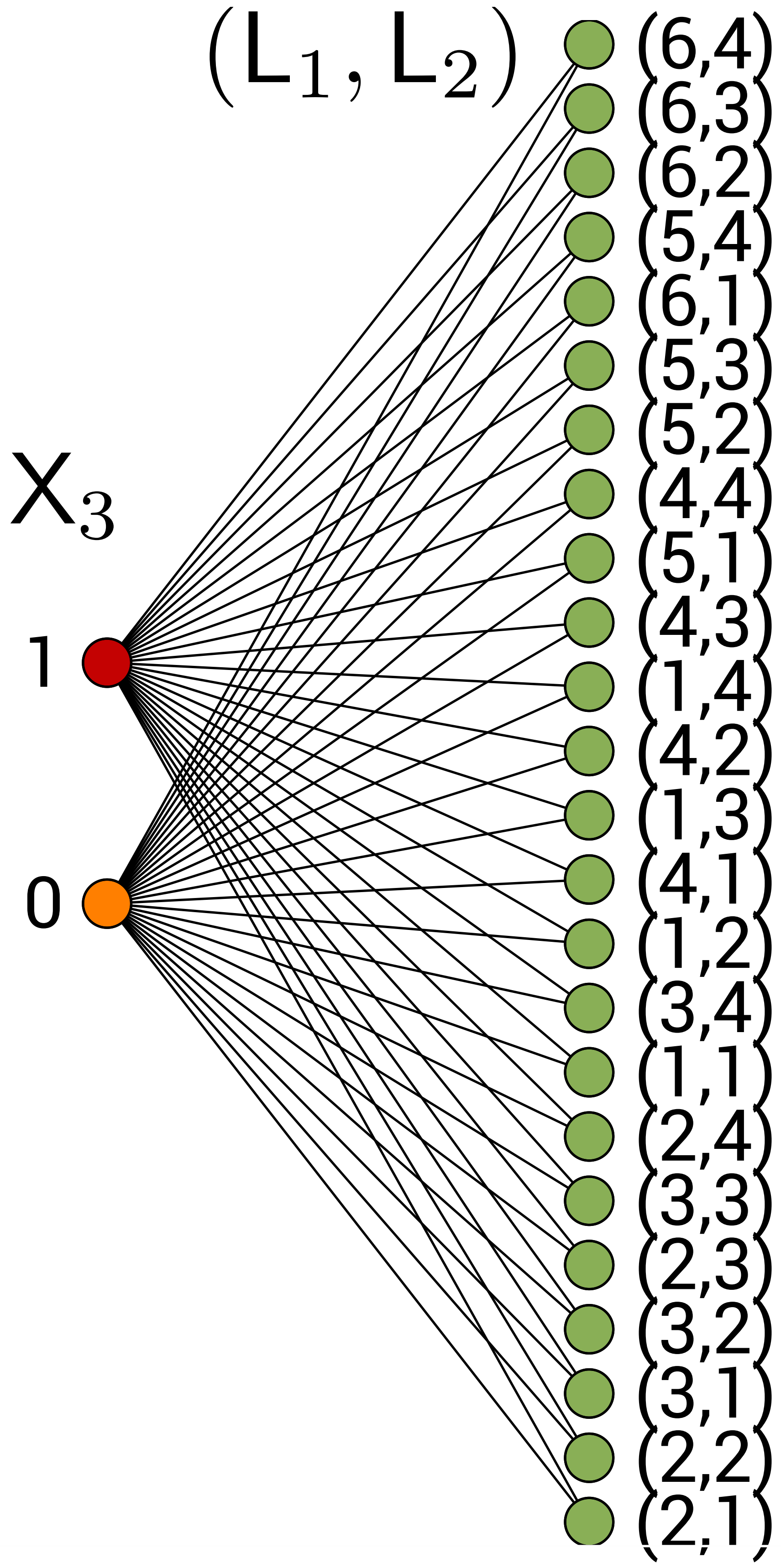


local constraint:

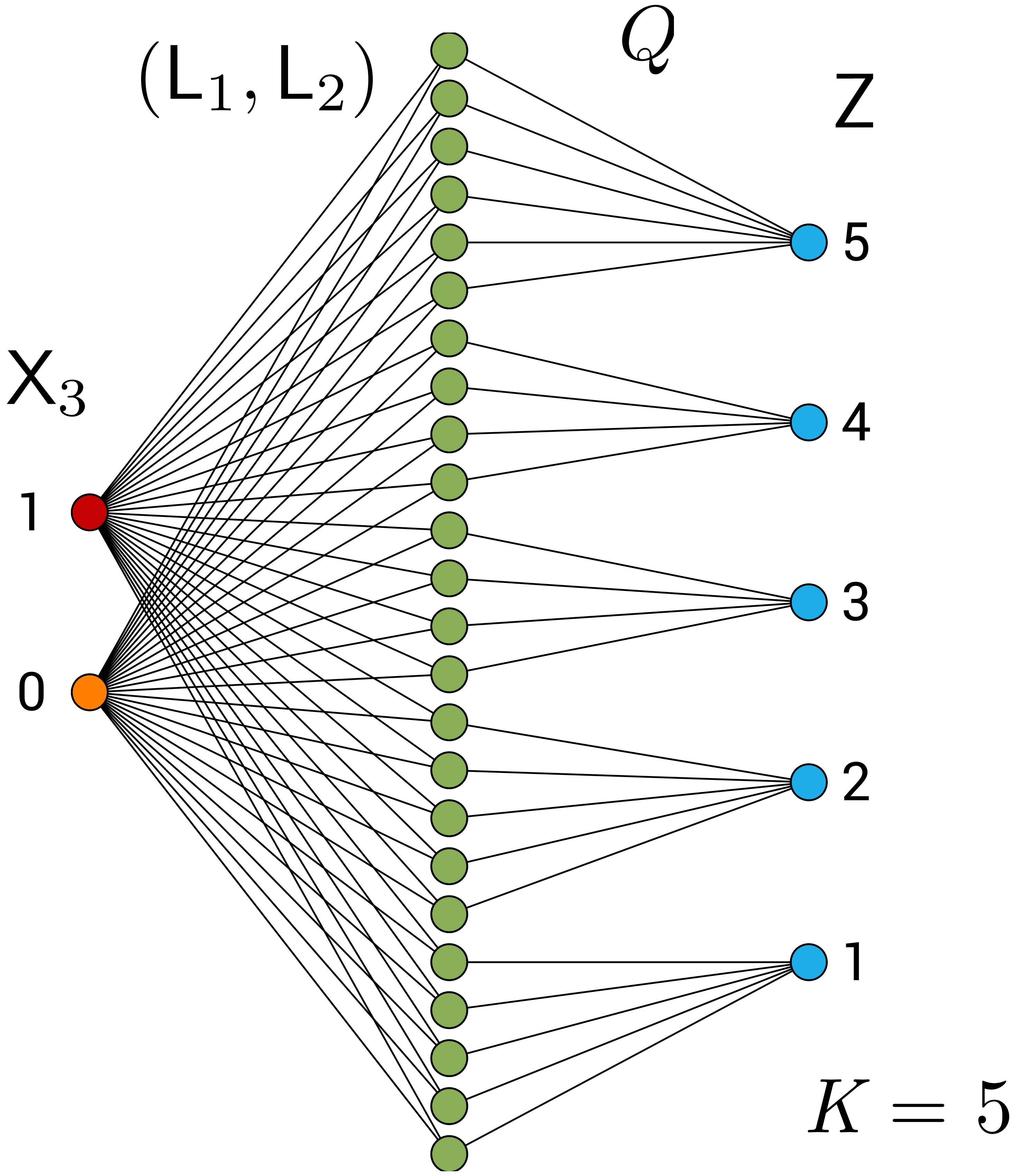
$f(x_1, x_2) = x_3$

Max-LUT Step 2: Quantize

- Too many levels! Reduce to Z with K levels
- Quantizer is a mapping from $(\mathsf{L}_1,\mathsf{L}_2)$ to Z

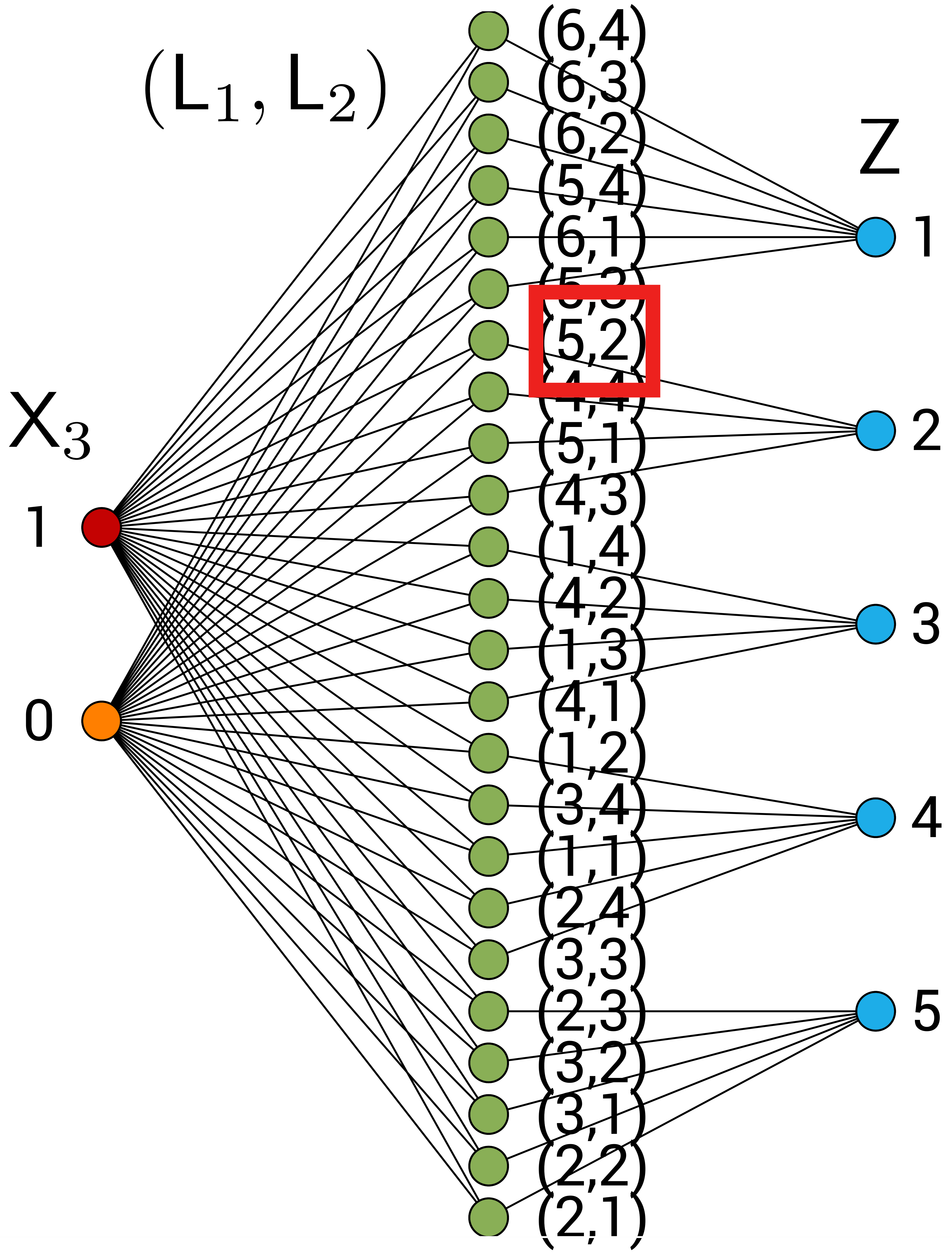


$$\max_{Q} I(\mathsf{X}_3;\mathsf{Z})$$



$$Q : \mathcal{L}_1 \times \mathcal{L}_2 \to \mathsf{Z}$$

Max-LUT Step 3: Lookup Table

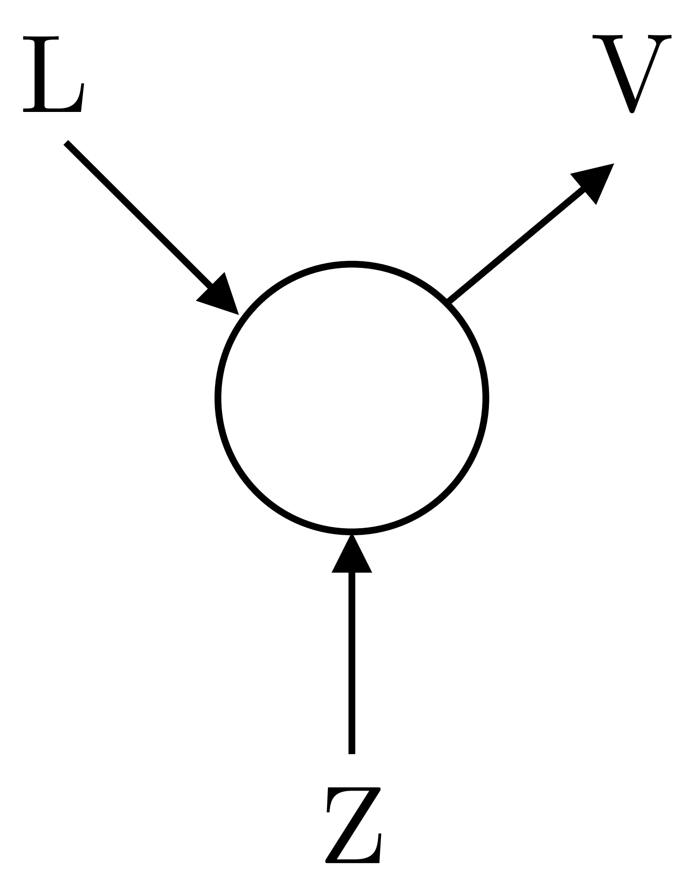


L_1 \ L_2	1	2	3	4
1	4	4	3	3
2	5	5	5	4
3	5	5	4	4
4	3	3	2	2
5	2	$\boxed{2}$	1	1
6	1	1	1	1

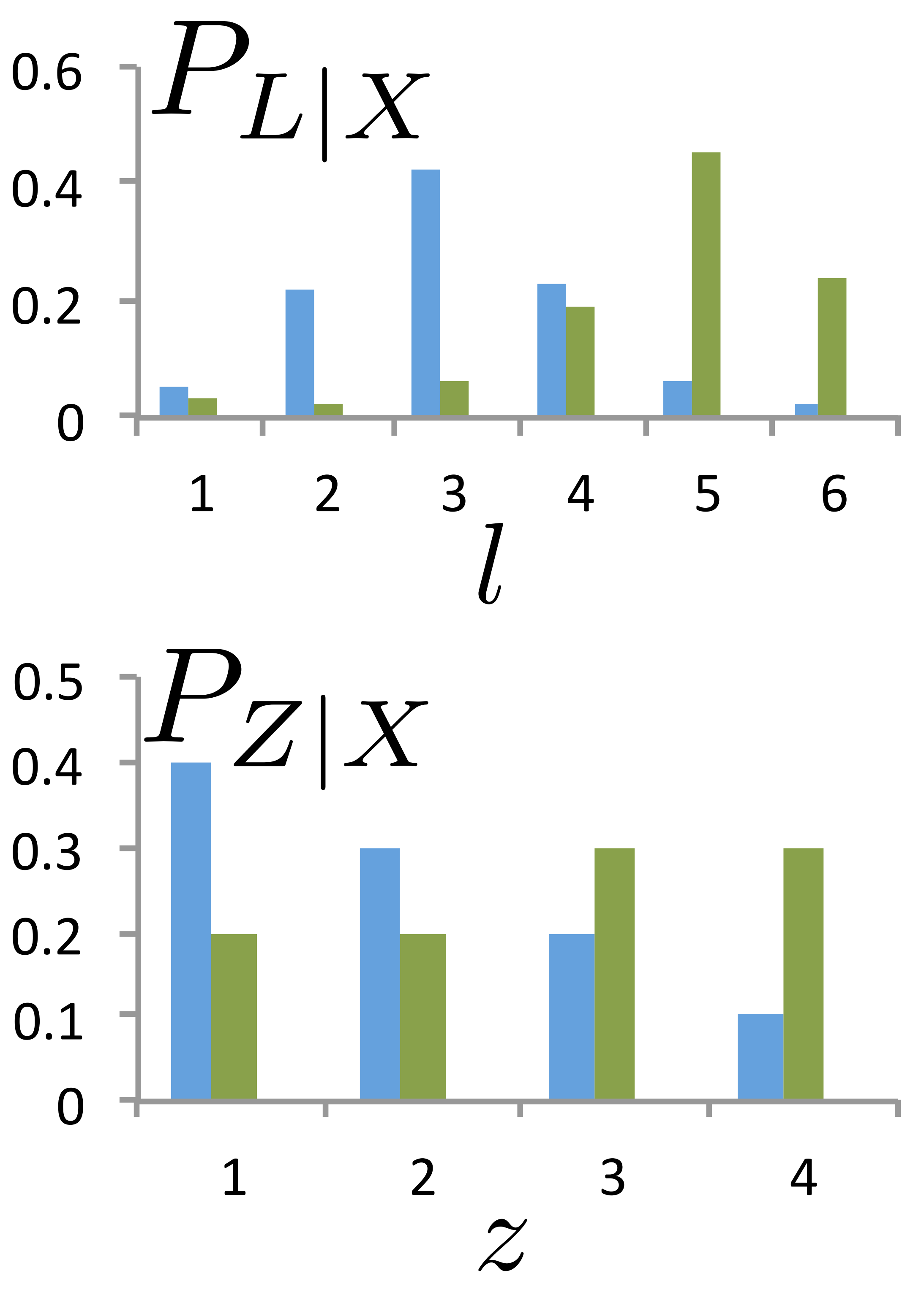
Lookup table:

$Z = \mathrm{LUT}(L_1, L_2)$

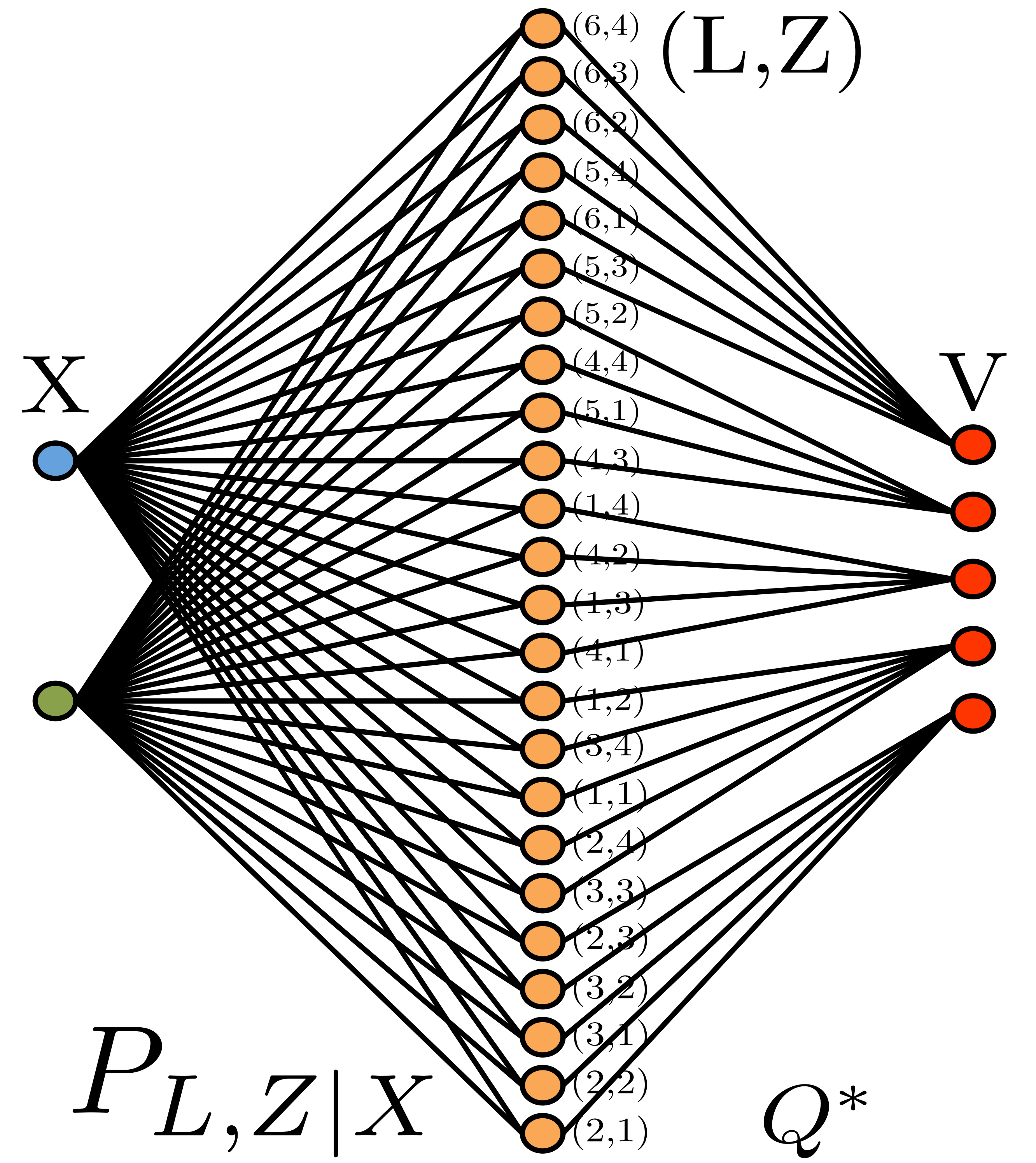
Max-LUT Method in One Slide



Variable node



Conditional distribution



Joint distribution Quantizer

	Z			
L	1	2	3	4
1	4	4	3	3
2	5	5	5	4
3	5	5	4	4
4	3	3	2	2
5	2	2	1	1
6	1	1	1	1

Lookup table
(mapping)

3 Application to LDPC Code Decoding

- How to obtain the densities needed by Max-LUT method?

  $>$ Density evolution

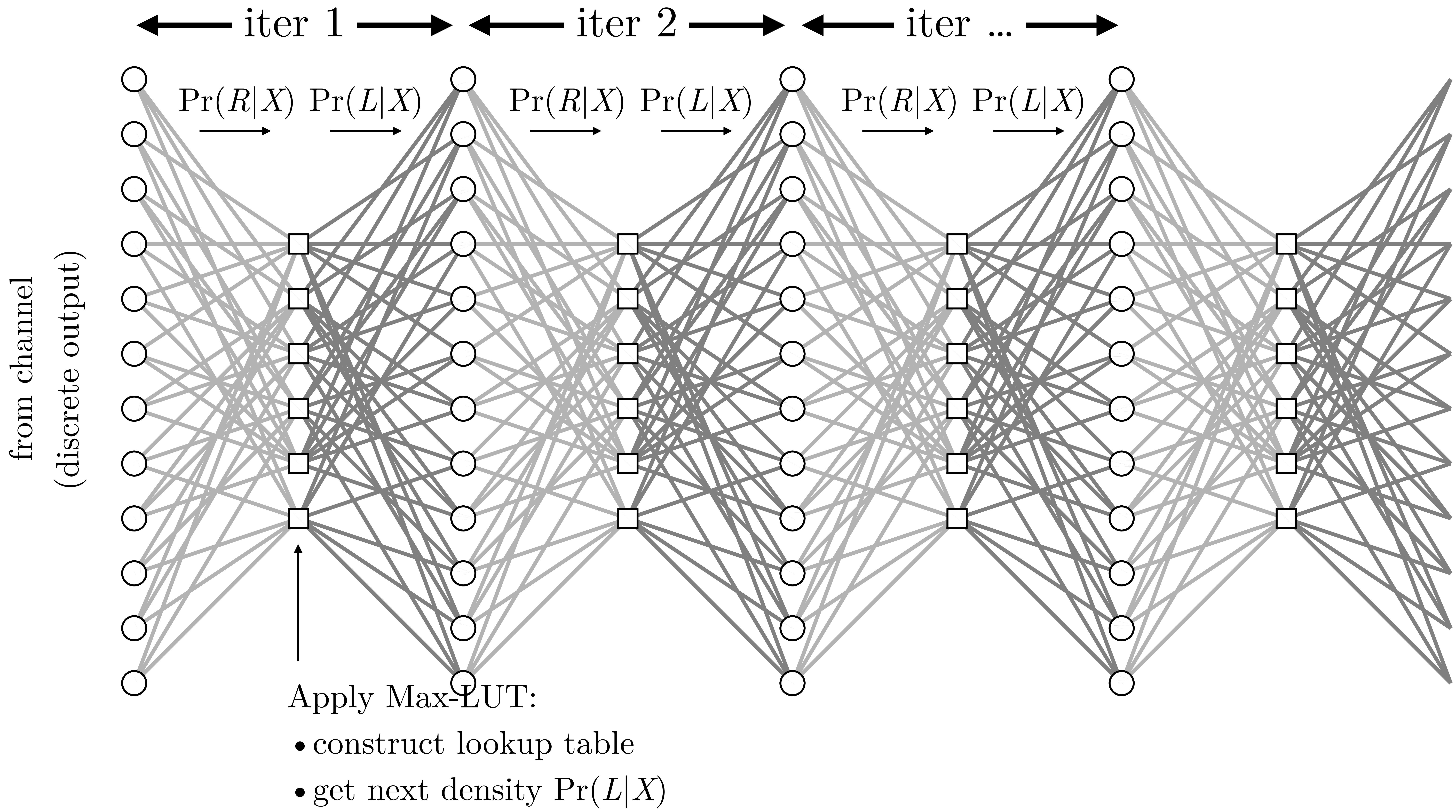
- How to keep the lookup table reasonable size?

  $>$ Node decomposition or “opening the node”

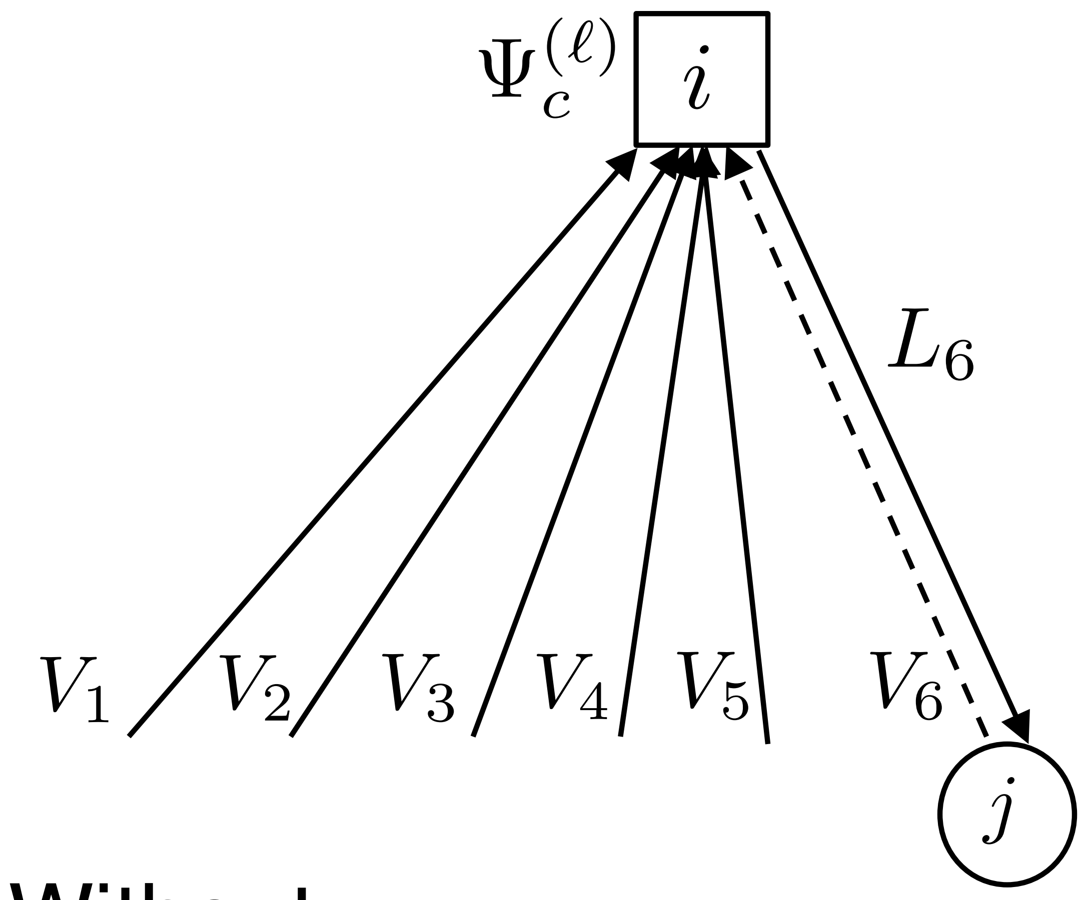
- How does it perform numerically?

  $>$ Similar to BP with four bits/message

Density Evolution Unwraps the Graph



Node Decomposition (Opening the node) Reduces size of lookup table



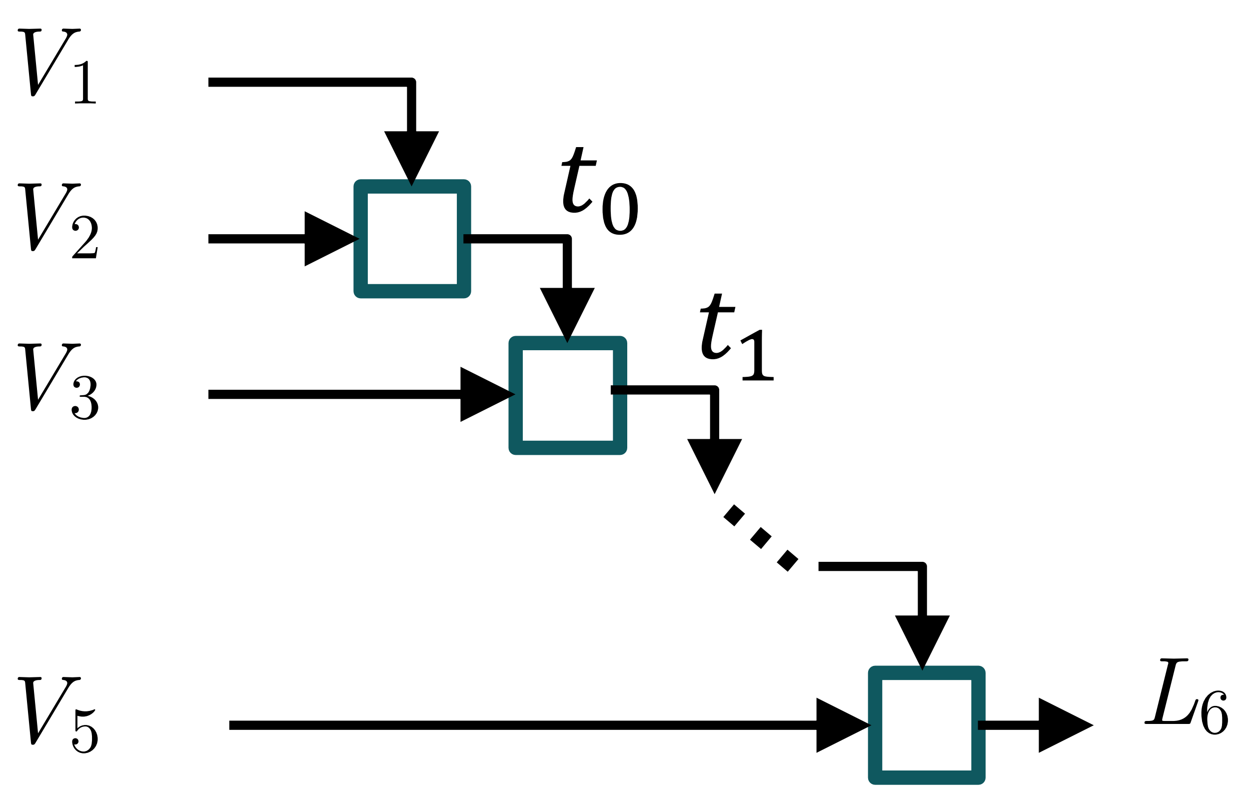
Without decomposition

Memory locations for the lookup table $\Psi_c^{(\ell)}$ (Δ = number of bits per message V)

$$2^{(d_c-1)\Delta}$$

Example: $d_c = 6$ and $\Delta = 3$ bits per message

$$2^{(d_c-1)\Delta} = 32768$$



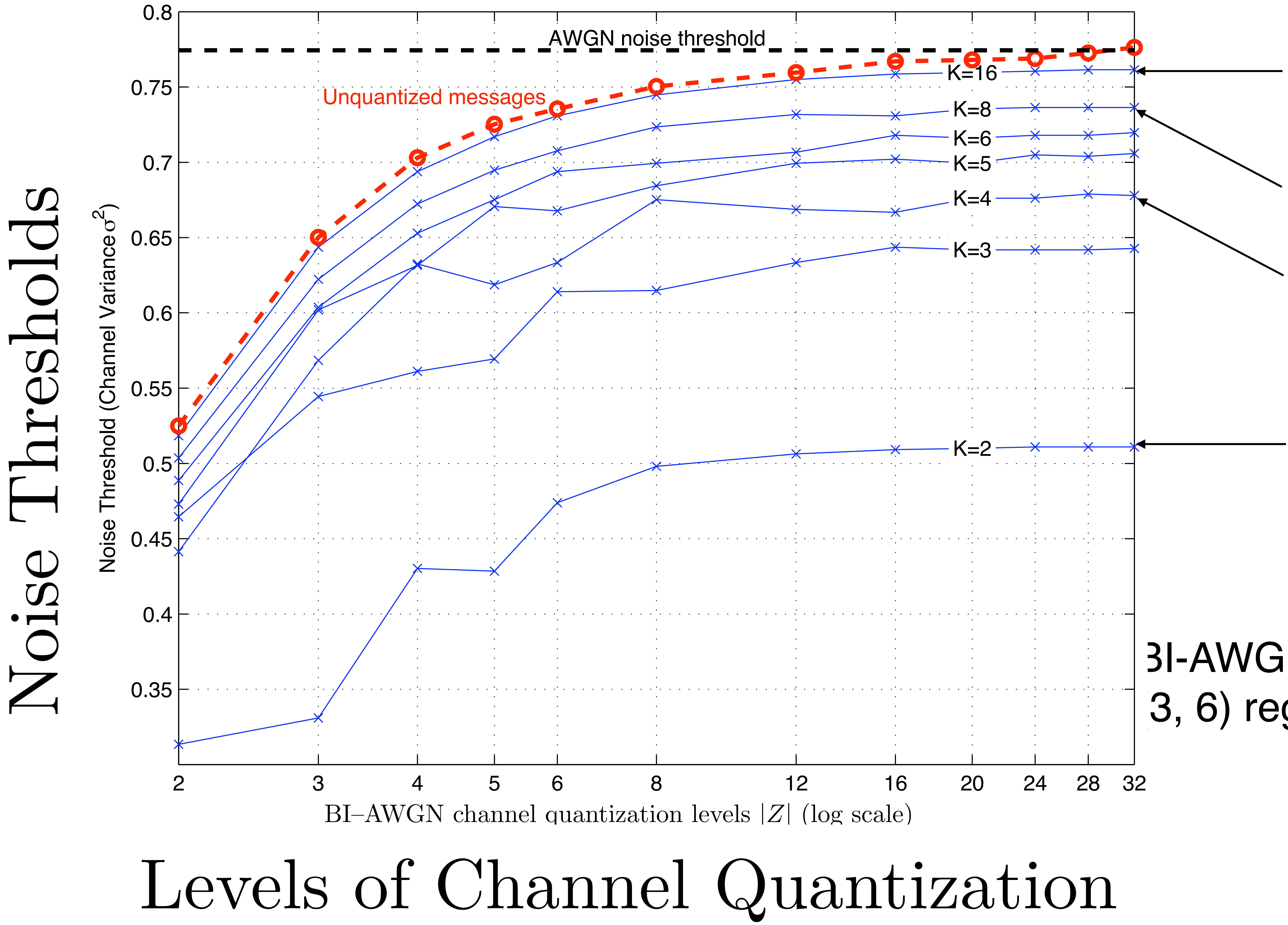
With decomposition

Memory locations for $\psi_1^{(\ell)}, \ldots, \psi_4^{(\ell)}$

$$(d_c - 2)2^{2\Delta}$$

$$(d_c - 2)2^{2\Delta} = 4 \times 64$$
$$= 256$$

Noise Thresholds with Quantization



BI-AWGN channel,
(3, 6) regular LDPC code

4 bits/message close to BP

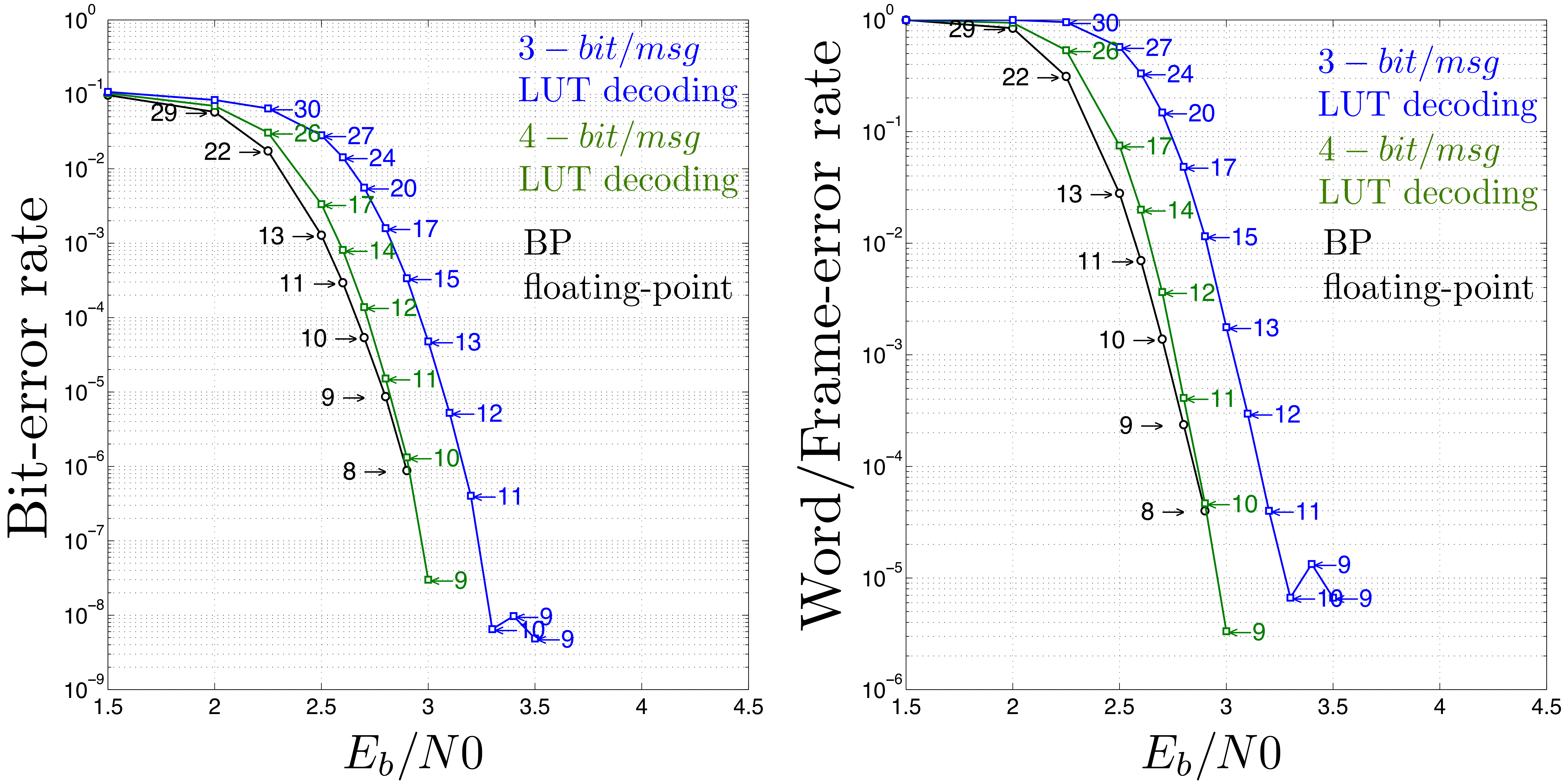


FIGURE 8. BER and WER results for the LUT decoding algorithm. $d_v = 4$, $d_c = 9$, $R = 0.56$, $N = 4113$, Max. Iter.= 30, Array code [2]. The numbers on the graph represent the average number of iteration per $E_b/N0$.

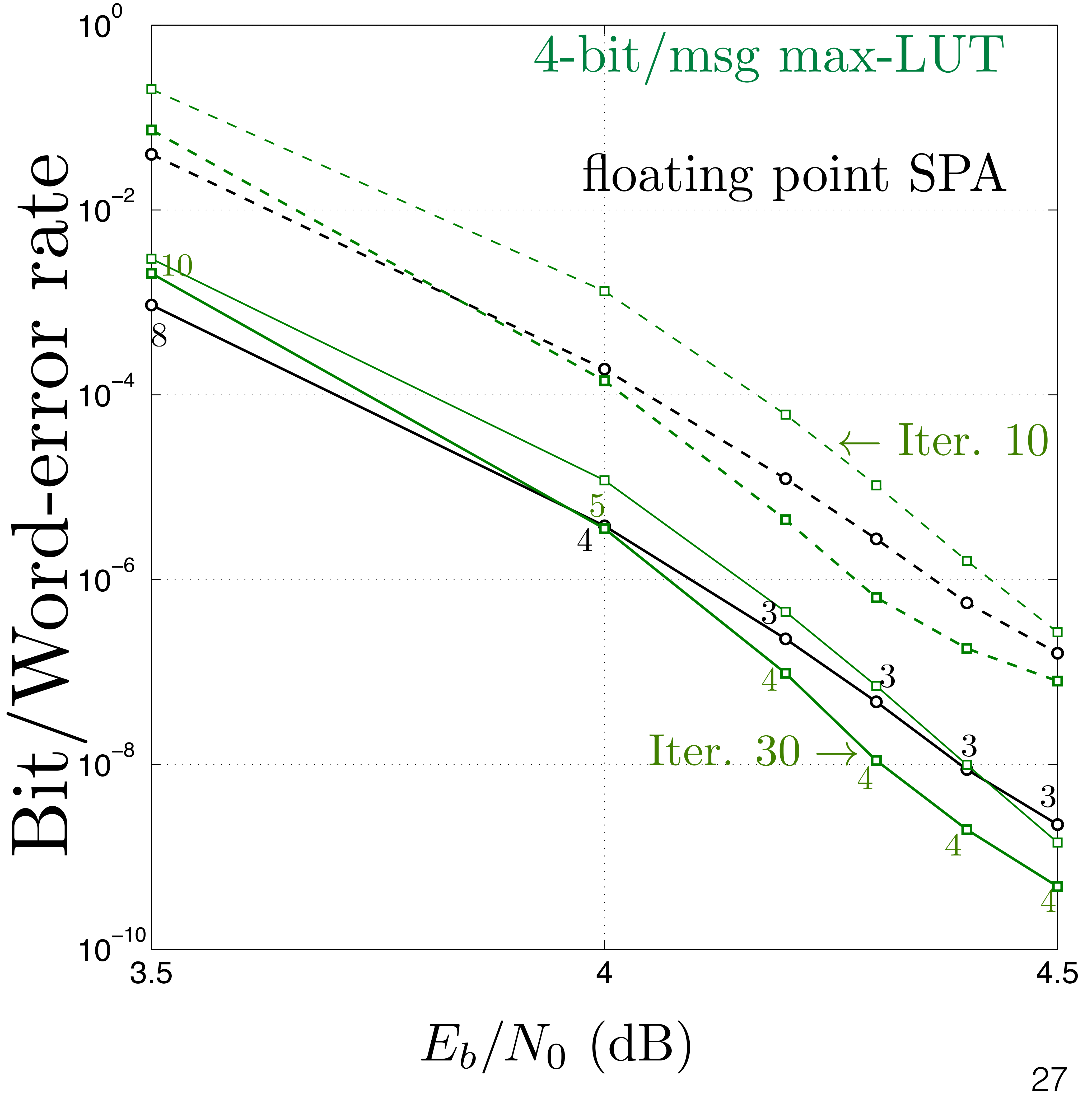
BI-AWGN: Lower Error Floors, Fewer Iterations

$N = 2048, (d_v = 6, d_c = 32), R = 0.84$ and Max. iter $= 30$

This code is used in IEEE 802.3an 10GBase-T standard producing an operation of 10 Gb/s.

The proposed decoding mapping functions

- using10 iterations can achieve the same BER performance than full SPA using 30 iterations.
- using 30 iterations can surpass the BER performance of full SPA using 30 iterations.



4 Discussion

- Where are the LLRs?
- Asymmetric or arbitrary noise models
- BCJR algorithm/turbo decoding

Where are the LLRs?

algorithm engineer

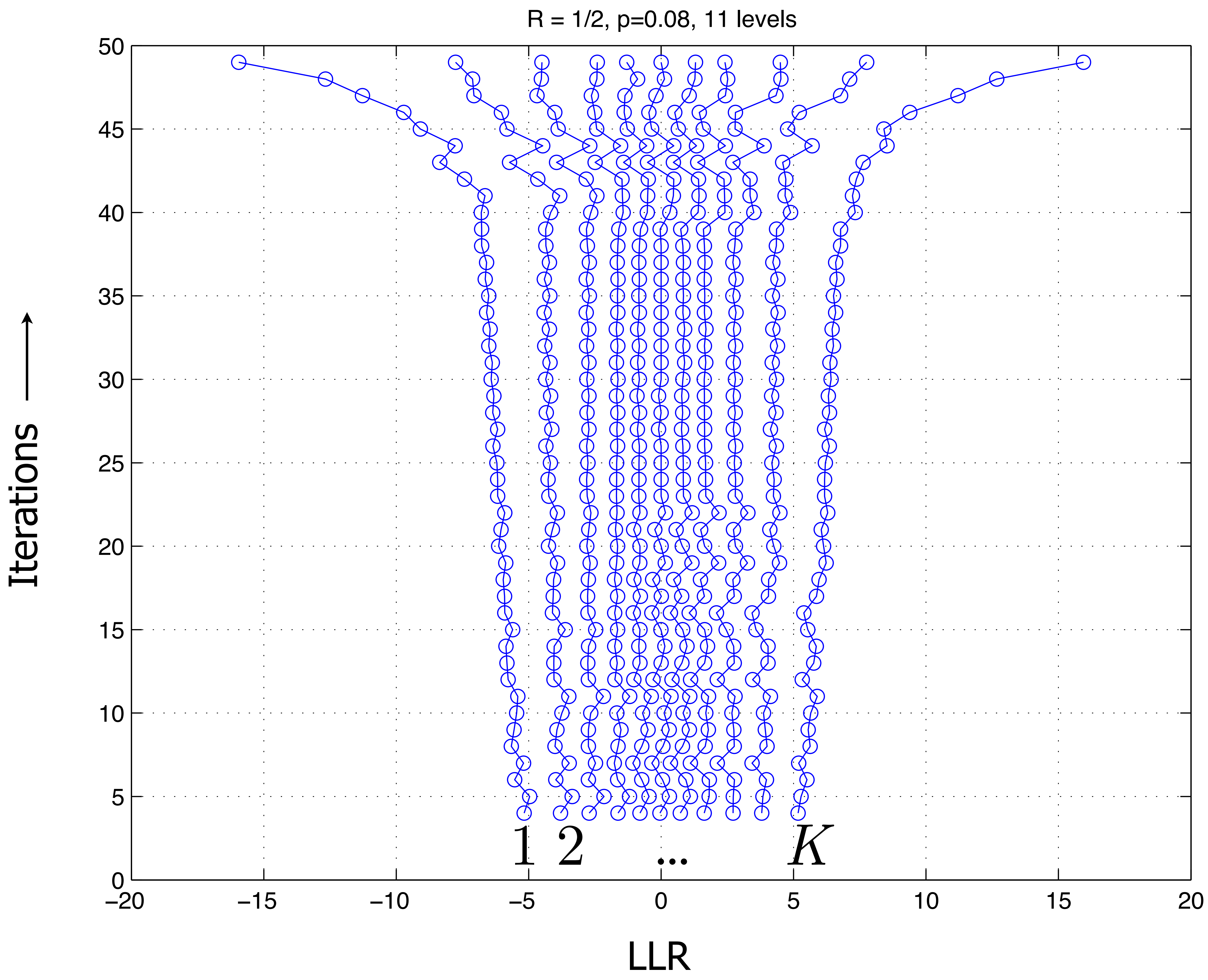
You can't just consider the LLRs quantization, you have to consider their distribution

Actually, we don't even consider the LLRs, we *only* consider the distribution

But then what are the LLRs?

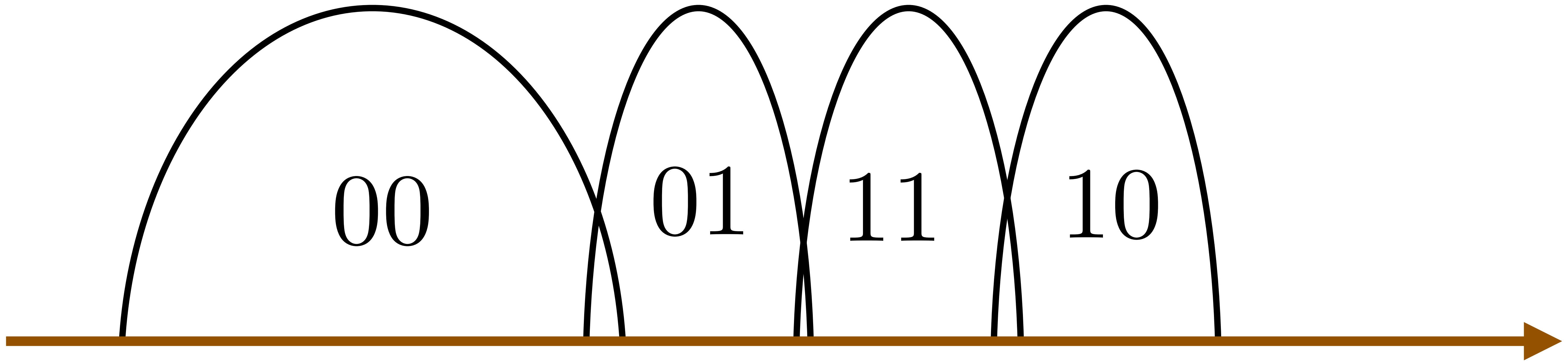
Only distributions are needed to compute mutual information.
Instead of LLRs, we just use integers {1,2,... K}

Here are the LLRs: After quantization



Can find the LLRs since we know the probability distributions.

Max-LUT Handles Arbitrary Noise



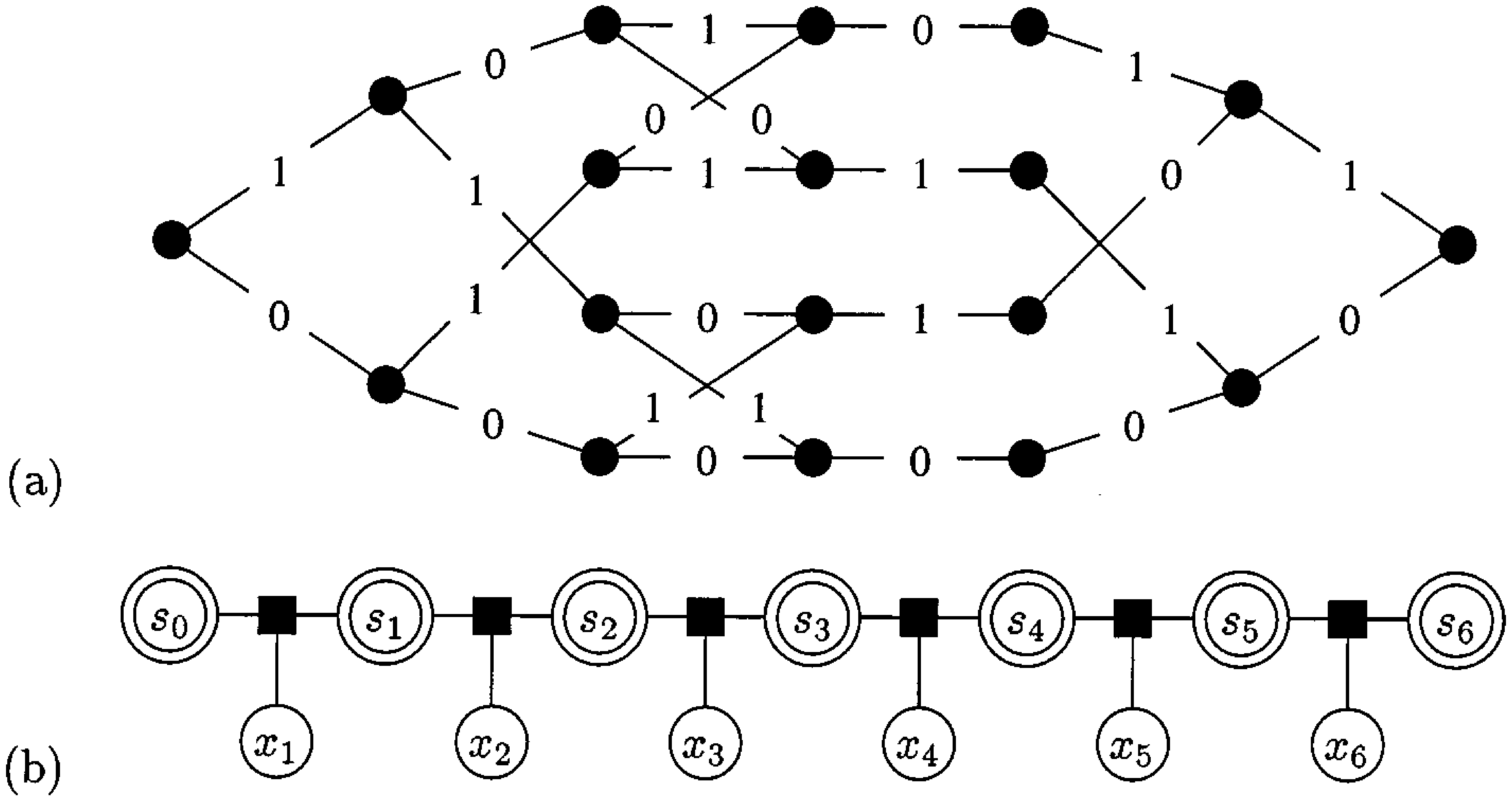
MLC flash with asymmetric noise

- Max-LUT method does not assume symmetric noise
- Density evolution using $\Pr(y \mid X = 0)$ and $\Pr(y \mid X = 1)$
- Generates lookup-tables optimized for asymmetric noise
- Suitable for flash memories and non-linear wireless channels.

BCJR algorithm/turbo decoding

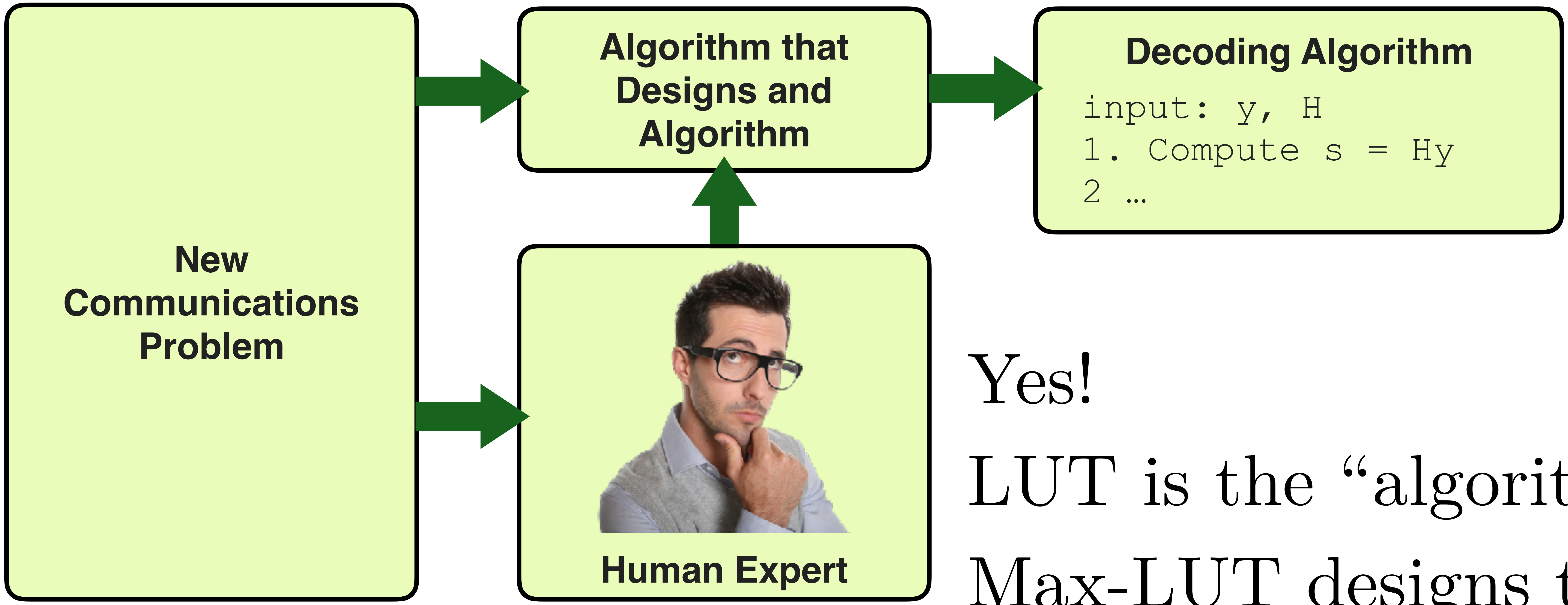
Factor graph for Markov chains (including time-varying)

Convolutional codes, turbo codes, intersymbol interference channels



F R Kschischang et al, “Factor Graphs and the Sum-Product Algorithm,” IT Trans, Feb. 2001

Conclusion: Did we make an “algorithm that designs algorithms”?



Yes!

LUT is the “algorithm”

Max-LUT designs the LUT

image: https://www.mindsonar.info http://bit.ly/2nbSXrp