Belief propagation decoding by passing quantum messages

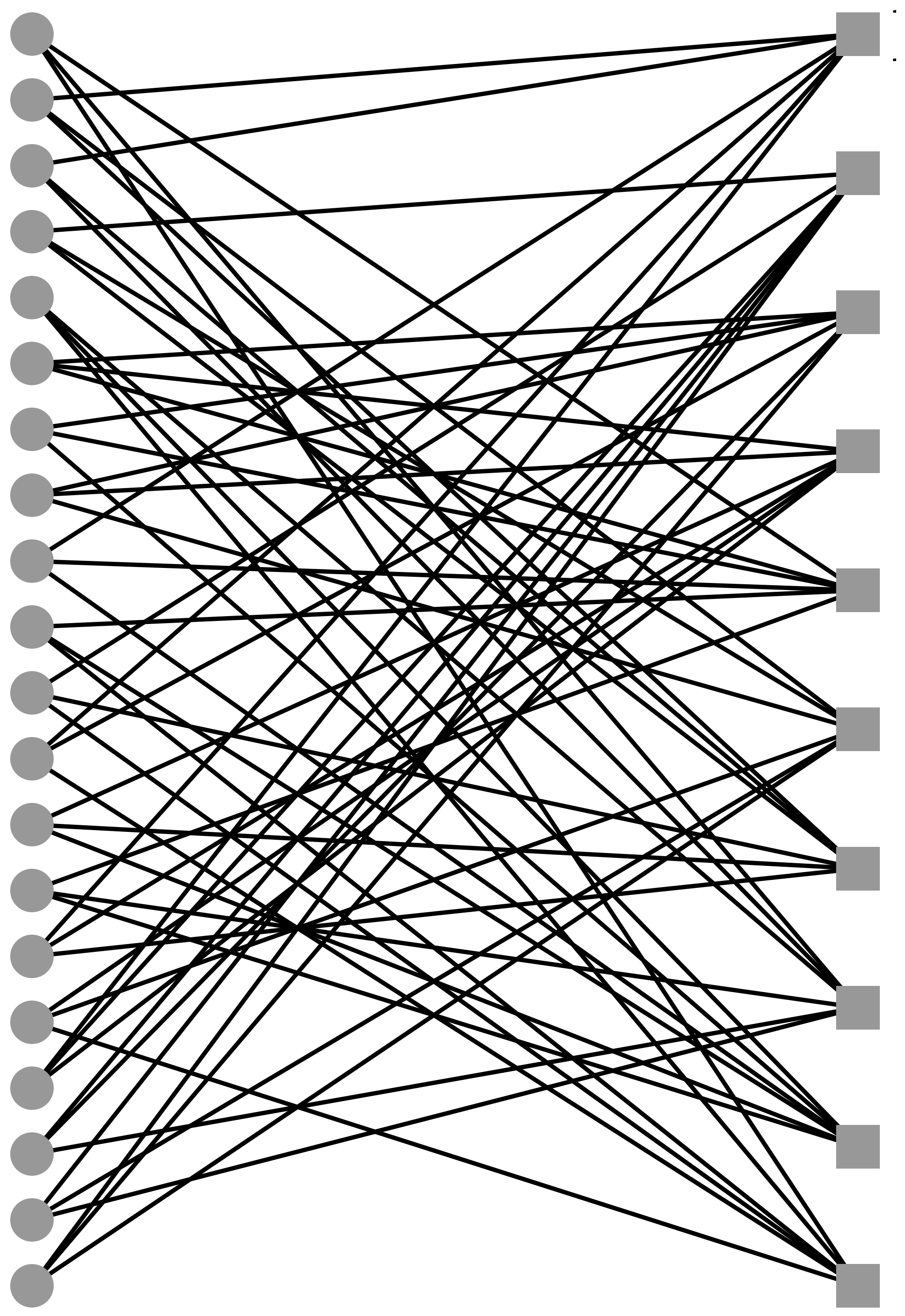
Joseph M. Renes, ETH Zürich

QuSoft Seminar, 26 February 2021

Joint work with Christophe Piveteau

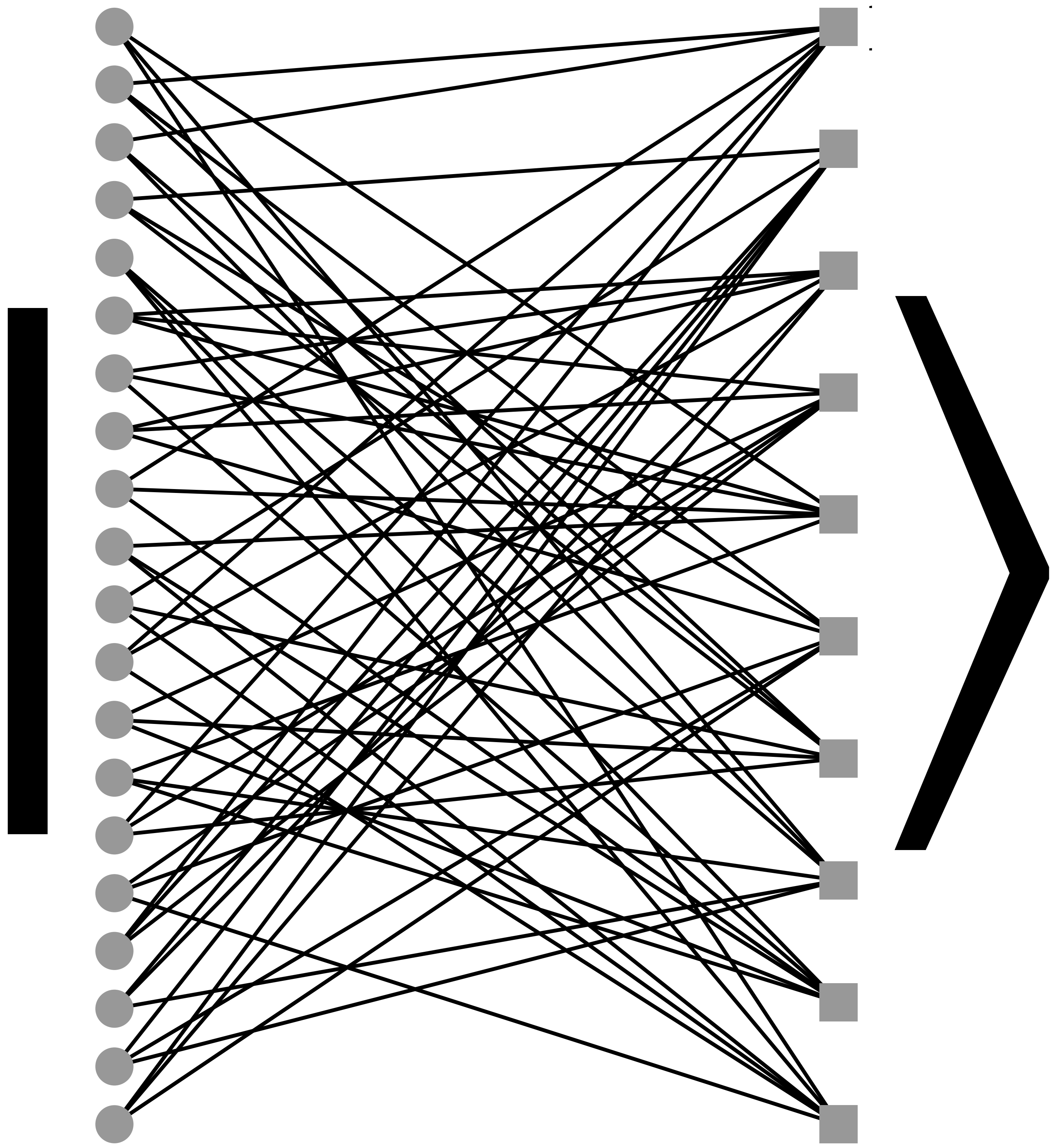
arXiv:1607.04833

arXiv:2103.?????



What is perhaps most distinctive about the graphical model approach is its **naturalness** in formulating probabilistic models of complex phenomena in applied fields, while maintaining control over the **computational cost** associated with these models.

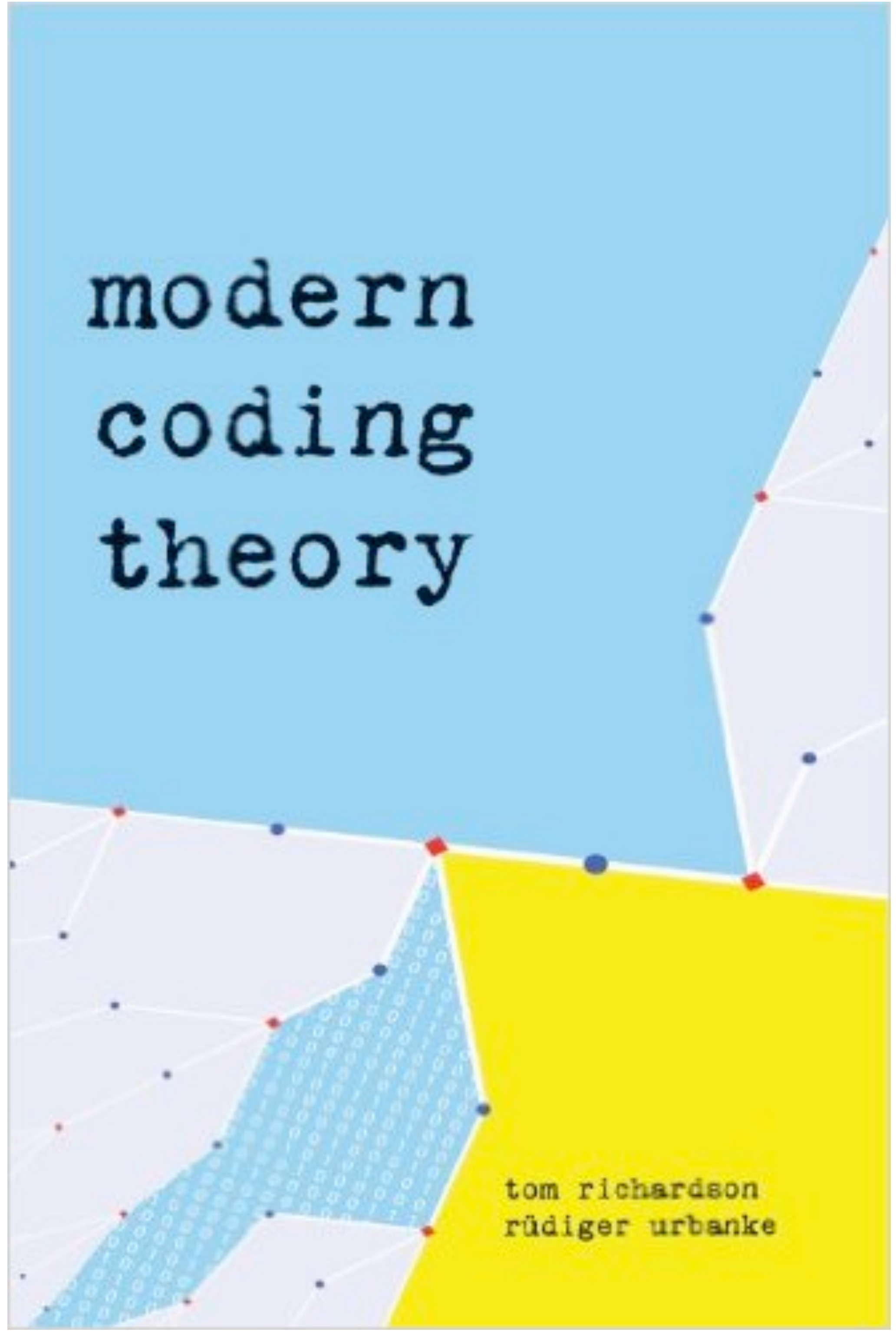
—Michael I. Jordan



?

To do research in quantum information theory, pick a favorite text on classical information theory, open to a chapter, and translate the contents into quantum-mechanical language.

—Benjamin Schumacher



Example 3.8 ((3, 6)-Regular Code). Consider the parity-check matrix

$$
(3.9) \qquad
\begin{array}{c}
\\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10
\end{array}
\begin{array}{c}
\begin{array}{cccccccccccccccccccc} 1&2&3&4&5&6&7&8&9&10&11&12&13&14&15&16&17&18&19&20 \end{array} \\
\begin{pmatrix}
0&0&0&0&1&0&0&0&1&1&1&0&0&0&0&1&0&0&0&1 \\
0&0&0&0&0&0&1&1&0&0&1&1&0&1&0&1&0&0&0&0 \\
0&1&1&0&0&0&1&0&0&0&0&0&0&0&0&1&0&1&0&1 \\
0&0&0&0&0&1&0&1&0&1&0&0&0&0&0&0&1&1&1&0 \\
1&1&0&0&1&0&0&0&0&0&0&0&1&0&0&0&1&0&1&0 \\
0&0&0&0&0&0&1&0&0&0&1&1&0&1&1&0&0&0&0&1 \\
0&0&0&1&1&1&0&1&0&0&0&0&1&0&1&0&0&0&0&0 \\
1&0&1&0&0&0&0&0&1&0&0&0&1&1&1&0&0&0&0&0 \\
1&1&1&1&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0 \\
0&0&0&1&0&1&0&0&1&0&0&1&0&0&0&0&0&1&1&0
\end{pmatrix}
\end{array}
= H.
$$

The bipartite graph representing $C(H)$ is shown on the left of Figure 3.10. Each check

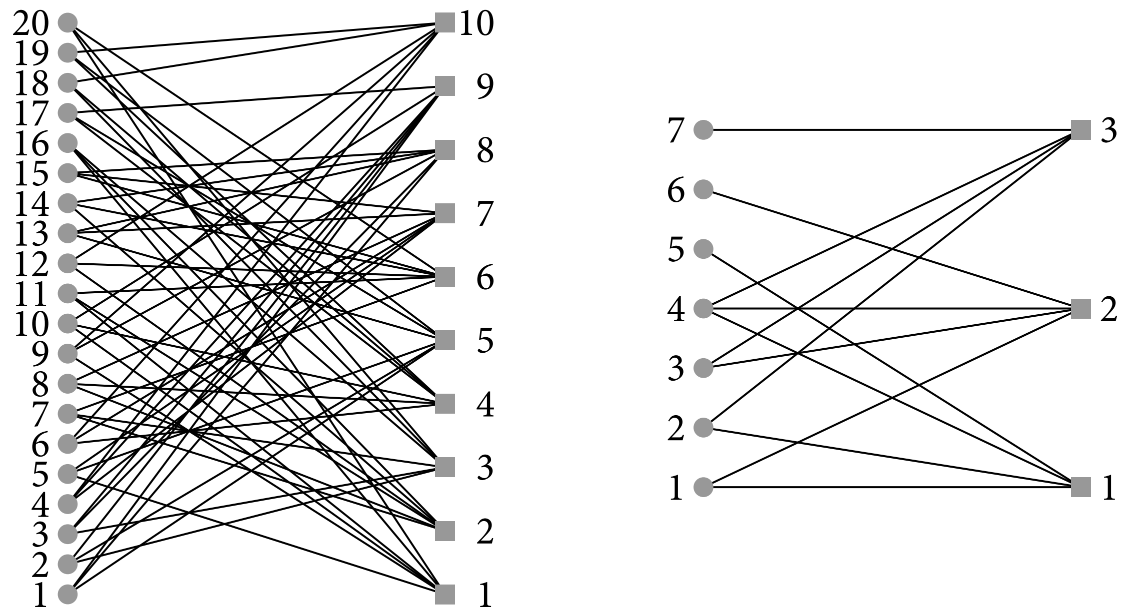


Figure 3.10: Left: Tanner graph of H given in (3.9). Right: Tanner graph of $[7, 4, 3]$ Hamming code corresponding to the parity-check matrix on page 15. This graph is discussed in Example 3.11.

node represents one linear constraint (one row of H). For the particular example we start with 20 degrees of freedom (20 variable nodes). The 10 constraints reduce the number of degrees of freedom by at most 10 (and exactly by 10 if all these constraints are linearly independent as in this specific example). Therefore at least 10 degrees of freedom remain. It follows that the shown code has rate (at least) one-half. ◇

§3.4. Low-Density Parity-Check Codes

In a nutshell, *low-density parity-check* (LDPC) codes are linear codes that have at least one *sparse* Tanner graph. The primary reason for focusing on such codes is

Quantum belief propagation decoding?

Outline

- Factor graphs and belief propagation
- BPQM: Passing quantum messages for single bit estimation
- Successive BPQM for entire codewords
- Applications: Efficient decoders at capacity
- Summary and open questions

Outline

- Factor graphs and belief propagation
- BPQM: Passing quantum messages for single bit estimation
- Successive BPQM for entire codewords
- Applications: Efficient decoders at capacity
- Summary and open questions

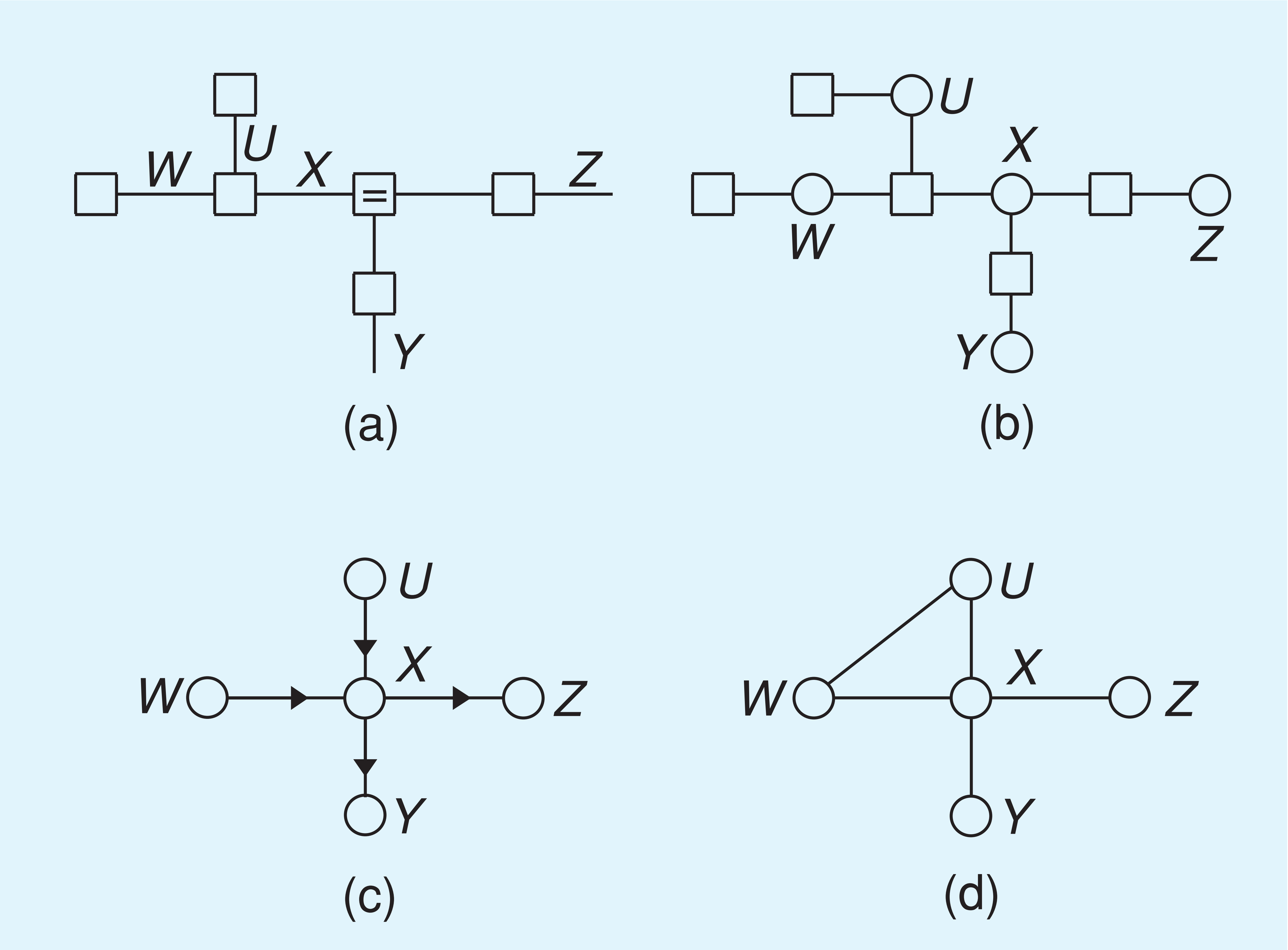
Graphical models

Forney-style factor graph

Usual factor graph

Bayesian network

Markov network



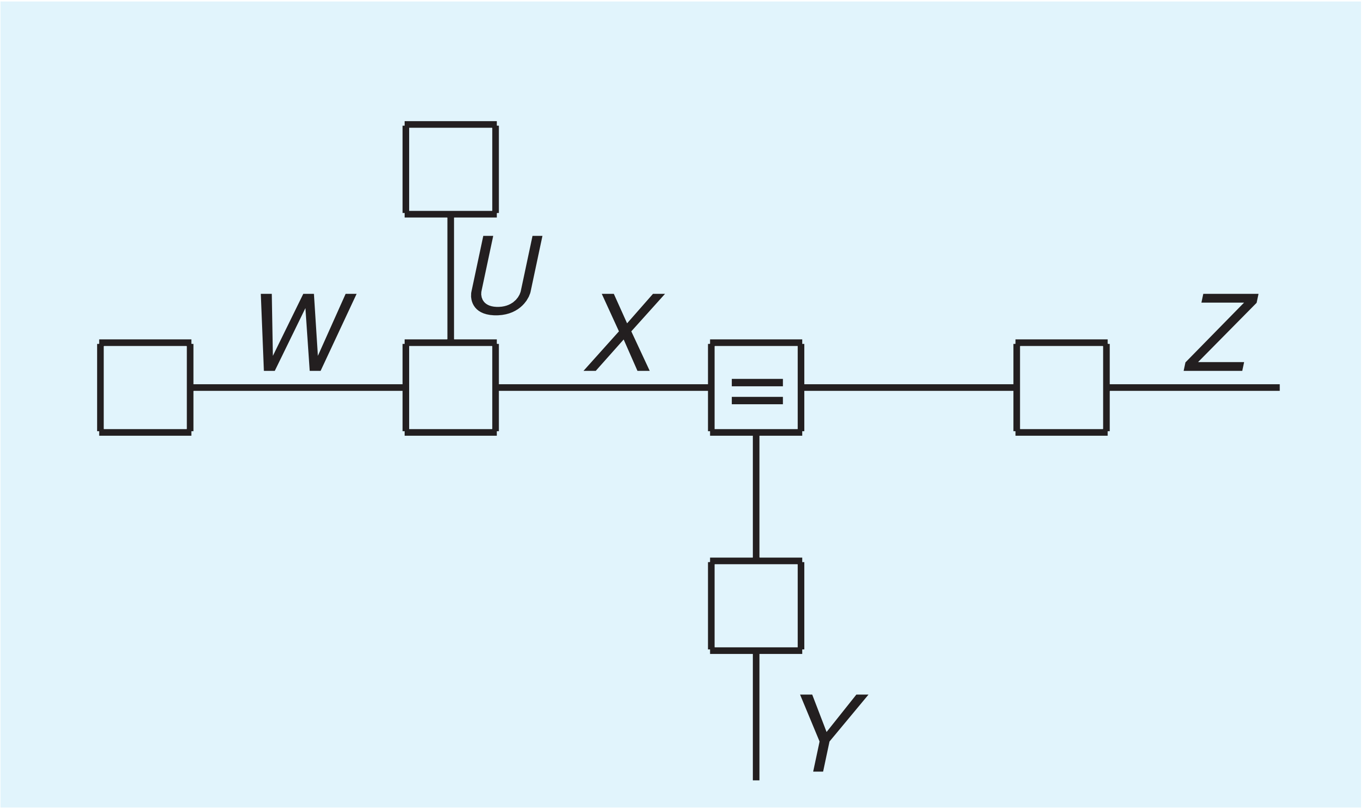
Loeliger, IEEE Signal Processing Magazine (2004)

$$P_{UWXYZ} = P_U P_W P_{X|UW} P_{Y|X} P_{Z|X}$$

Forney-style factor graphs… …are tensor networks!

Represent factorization of a function, e.g. a probability distribution

- One vertex / node per factor,
- Edge or half-edge per variable (duplicate variables with equality nodes if necessary),
- Node *f* connected to edge *x* when *x* appears in *f,*
- Sum over variables associated to edges

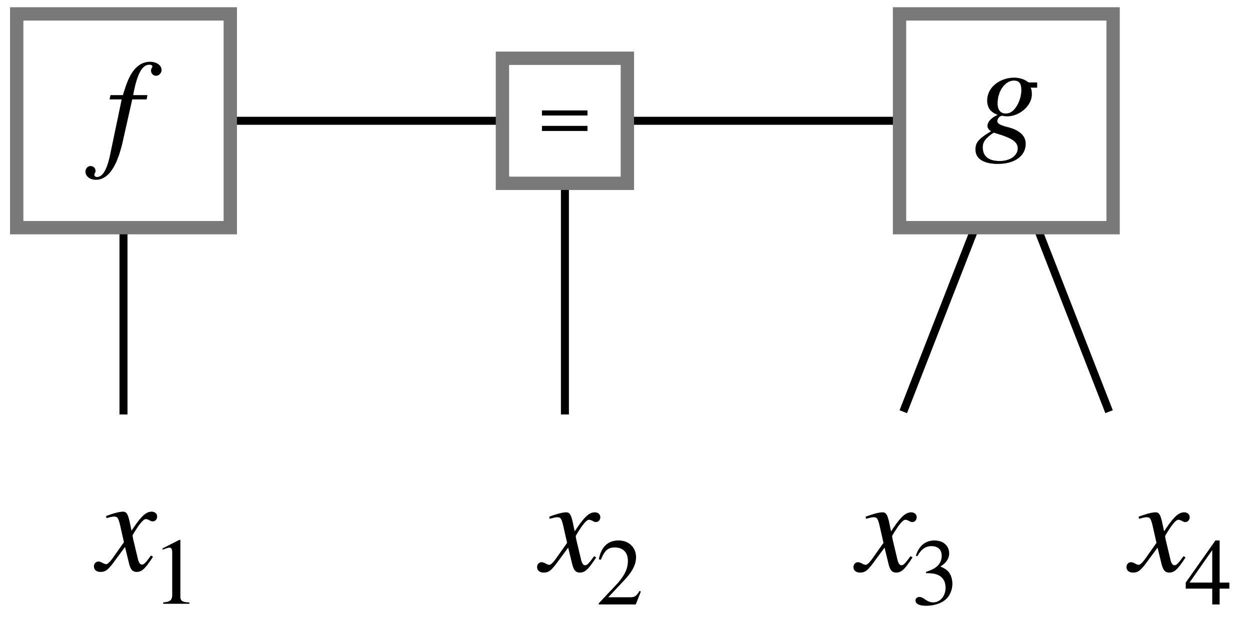


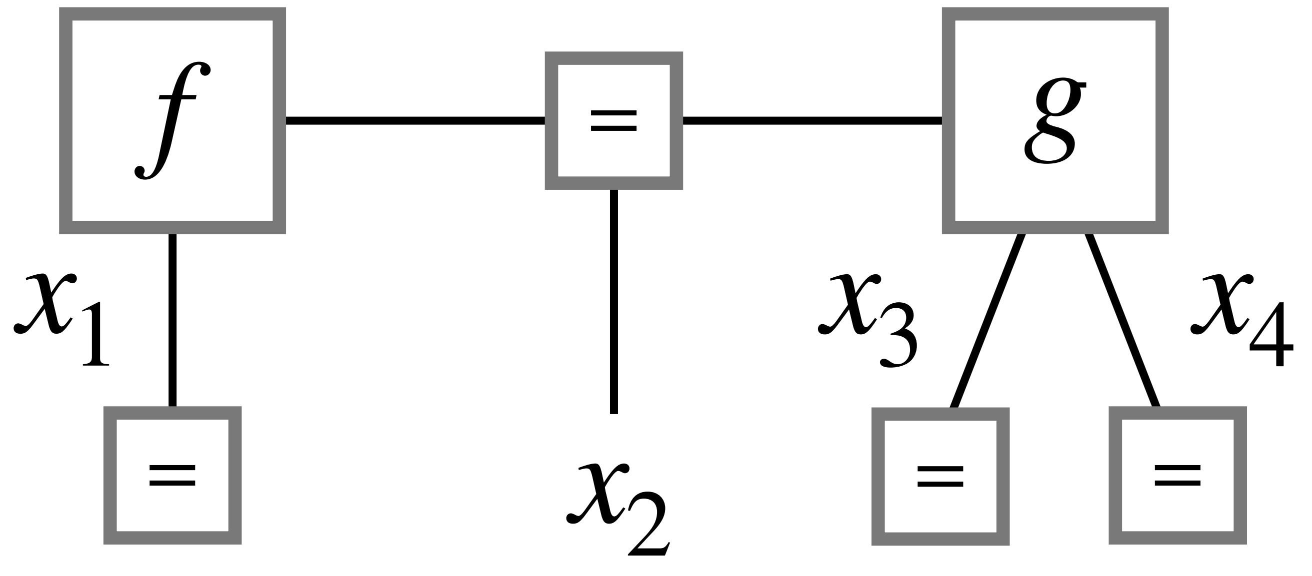
$$P_{YZ}(y,z) = \sum_{u,w,x} P_U(u)\, P_W(w)\, P_{X|UW}(x\,|\,u,w)\, P_{Y|X}(y\,|\,x)\, P_{Z|X}(z\,|\,x)$$

Belief propagation: marginalization by message passing

$$P(x_1, x_2, x_3, x_4) = f(x_1, x_2)\, g(x_2, x_3, x_4)$$

$$P(x_2) = \sum_{x_1, x_3, x_4} f(x_1, x_2)\, g(x_2, x_3, x_4)$$

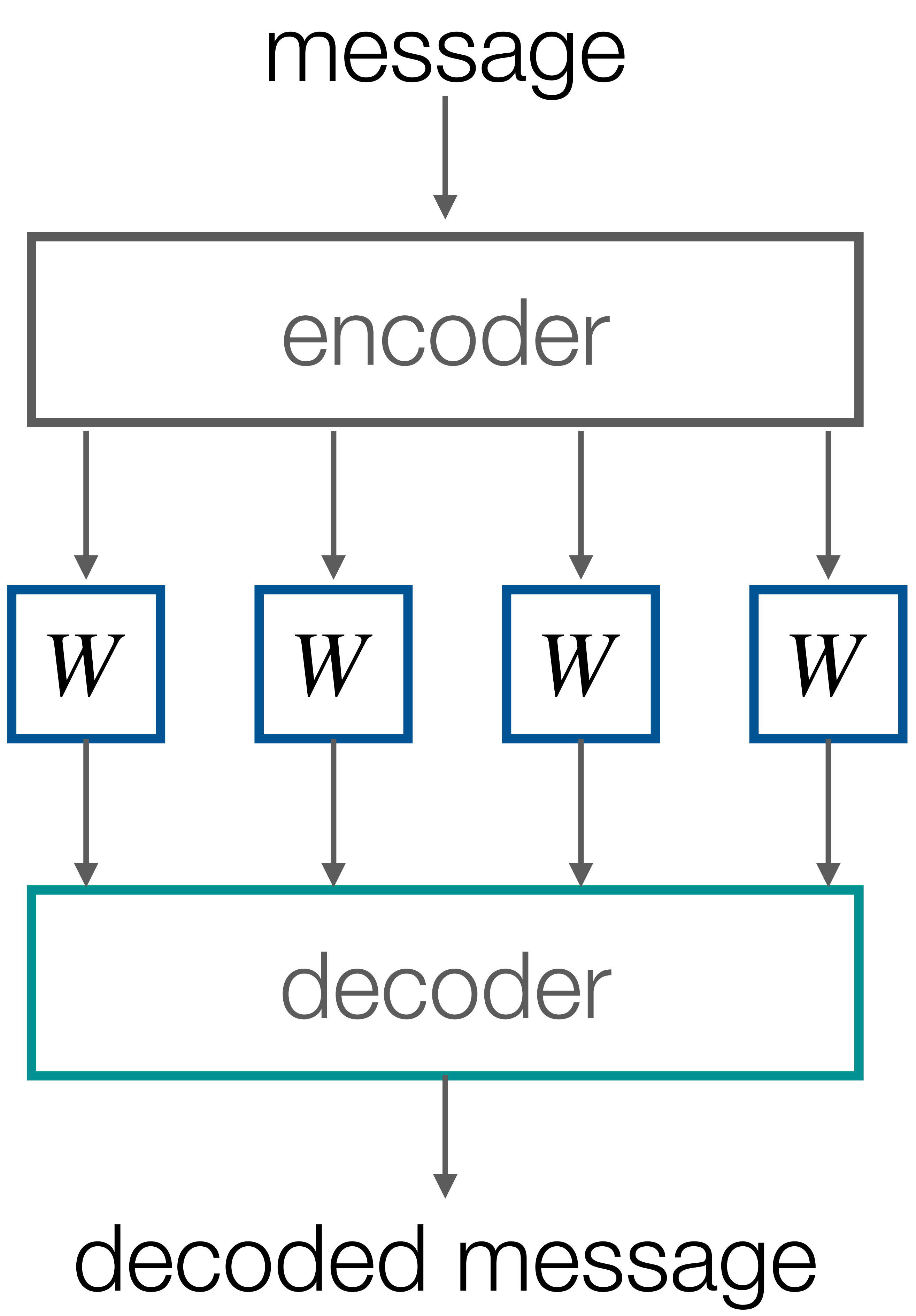




Messages are contracted tensors

Easy for tree factor graphs

Belief propagation in coding

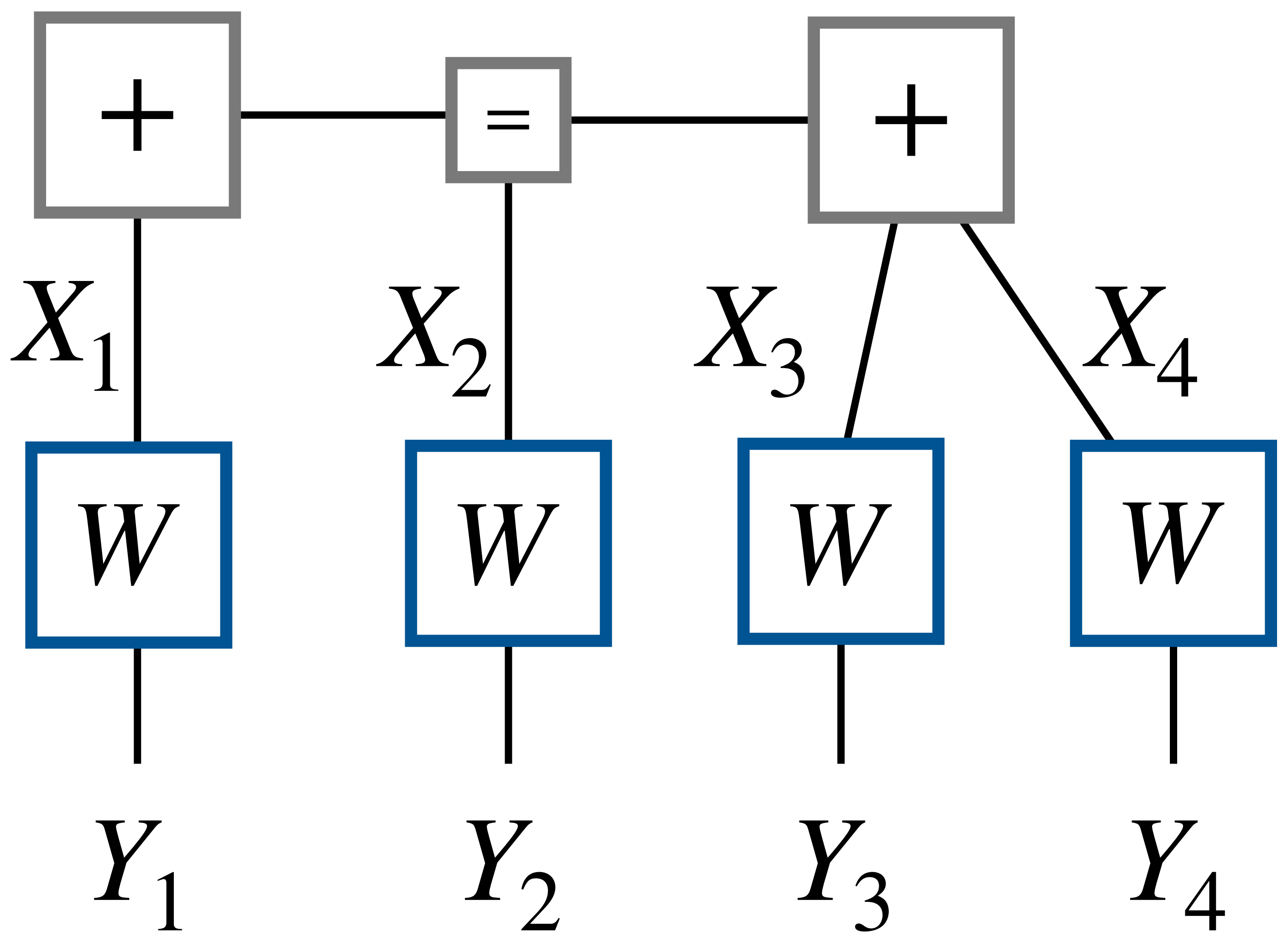


uniformly random

linear code

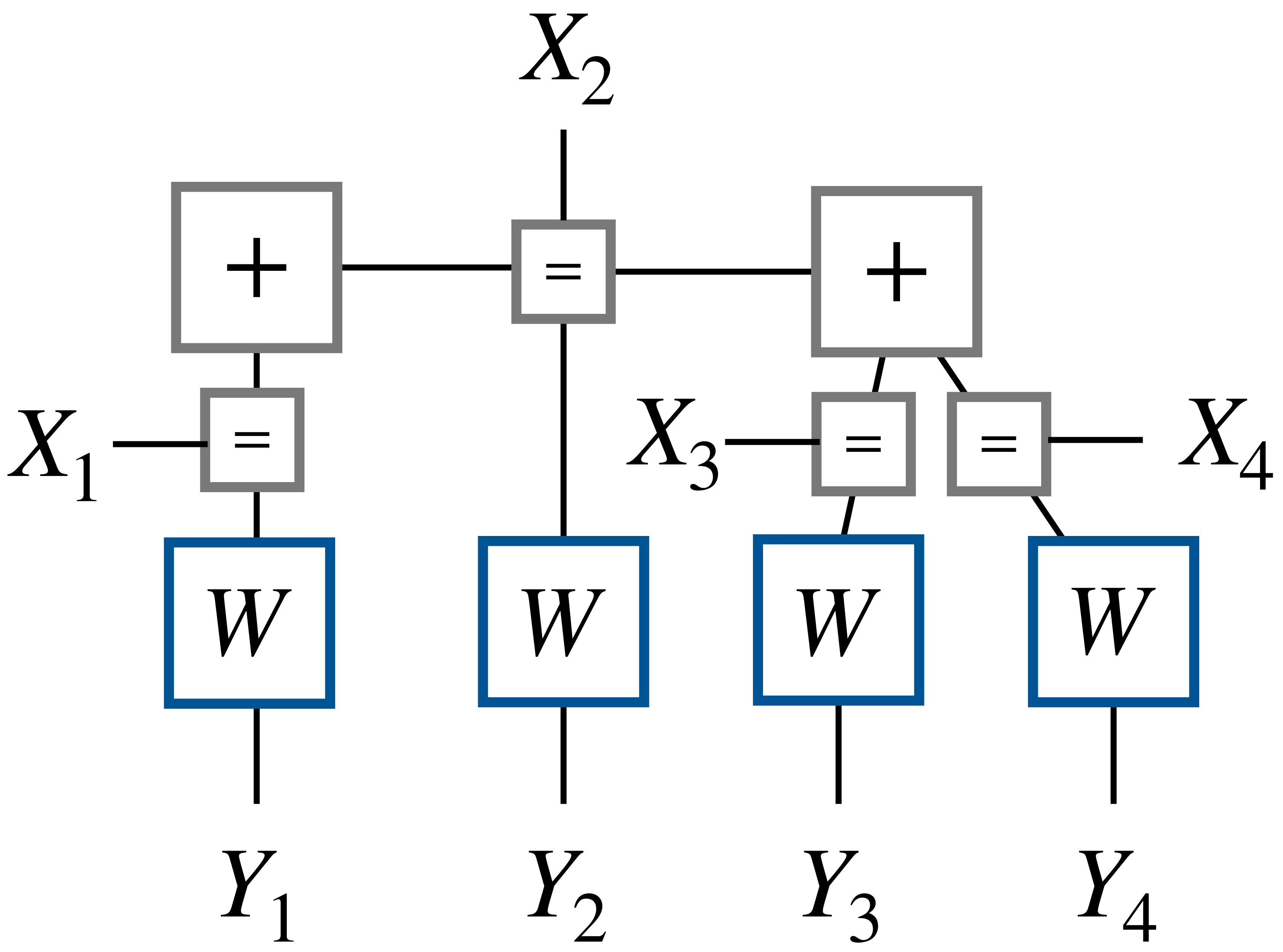
decode bitwise

Belief propagation in coding



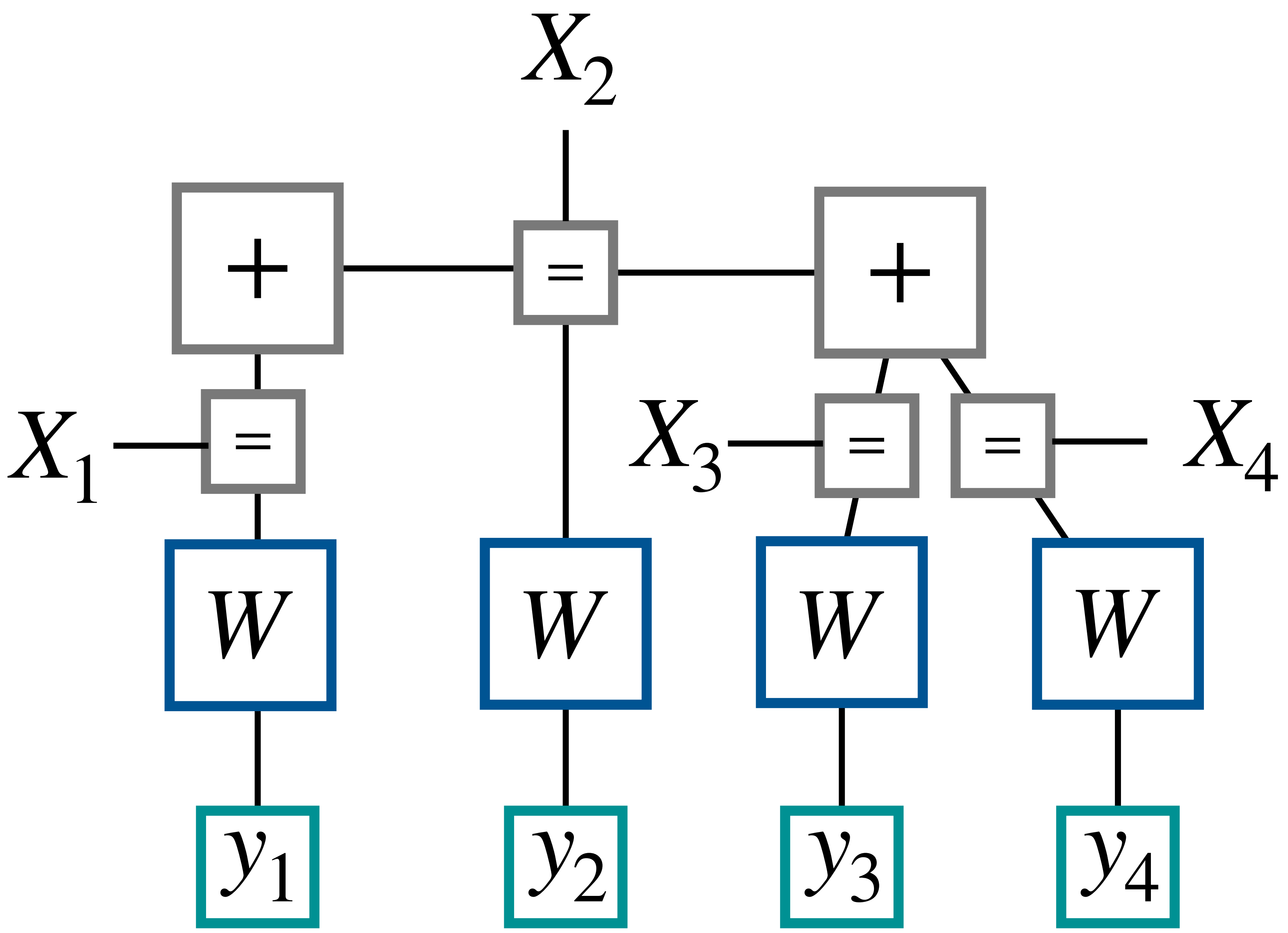
$$H = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

Belief propagation in coding



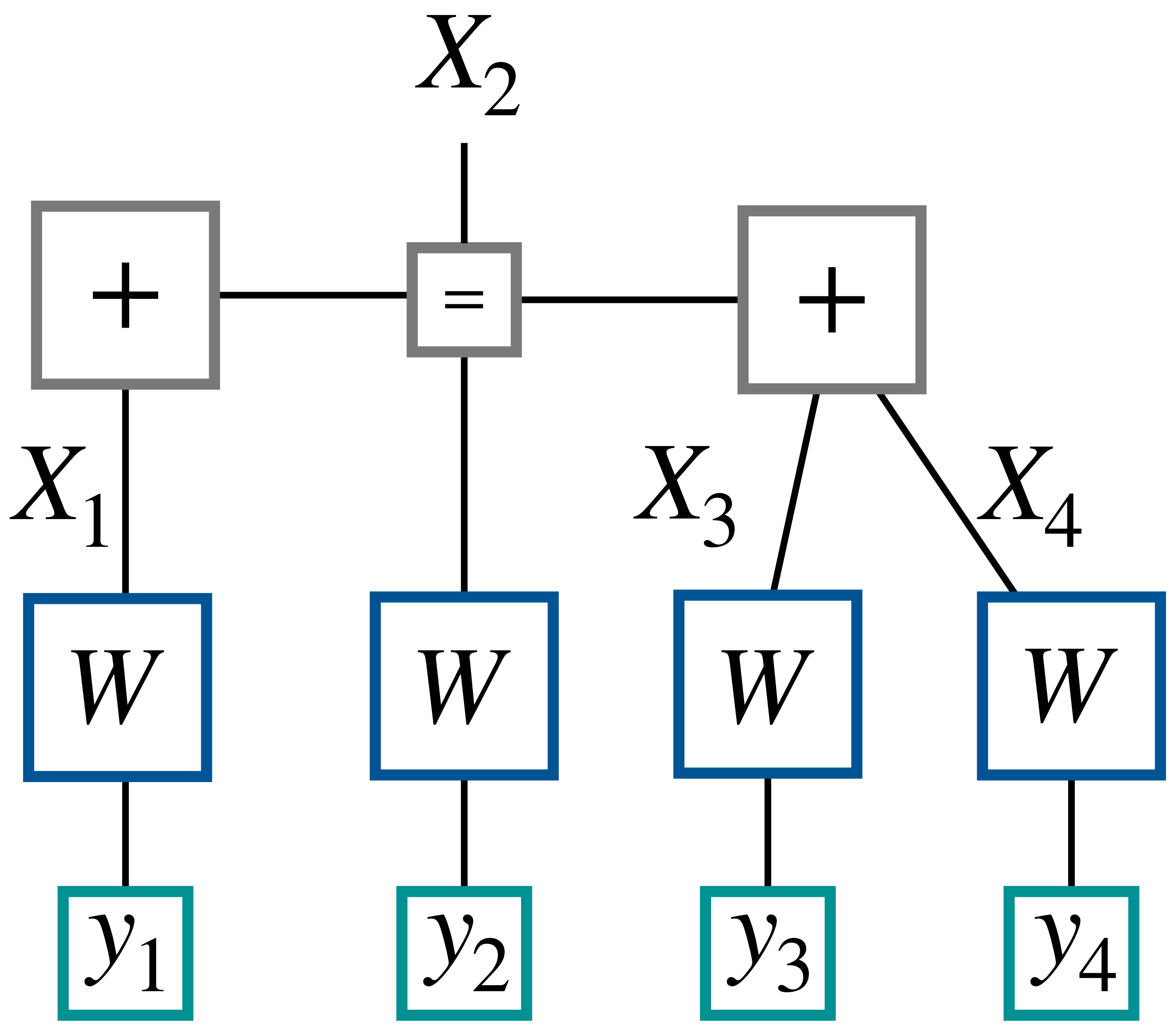
$$H = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

Belief propagation in coding



$$H = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

Belief propagation in coding



Contract to find estimate of X_2 given observed $y_1 y_2 y_3 y_4$.

Run in parallel to estimate all other codeword bits.

Other uses for BP (not the topic of this talk)

- Compute Bethe-Peierls approximate free energy (classical or quantum)
- Marginalize density matrices
- Decode quantum stabilizer codes (classical BP)

Yedidia, Freeman, Weiss, IEEE TIT 2005

Hastings, PRB, 2007

Poulin & Chung, QIC 2008

Liefer & Poulin, Ann. Phys. 2008

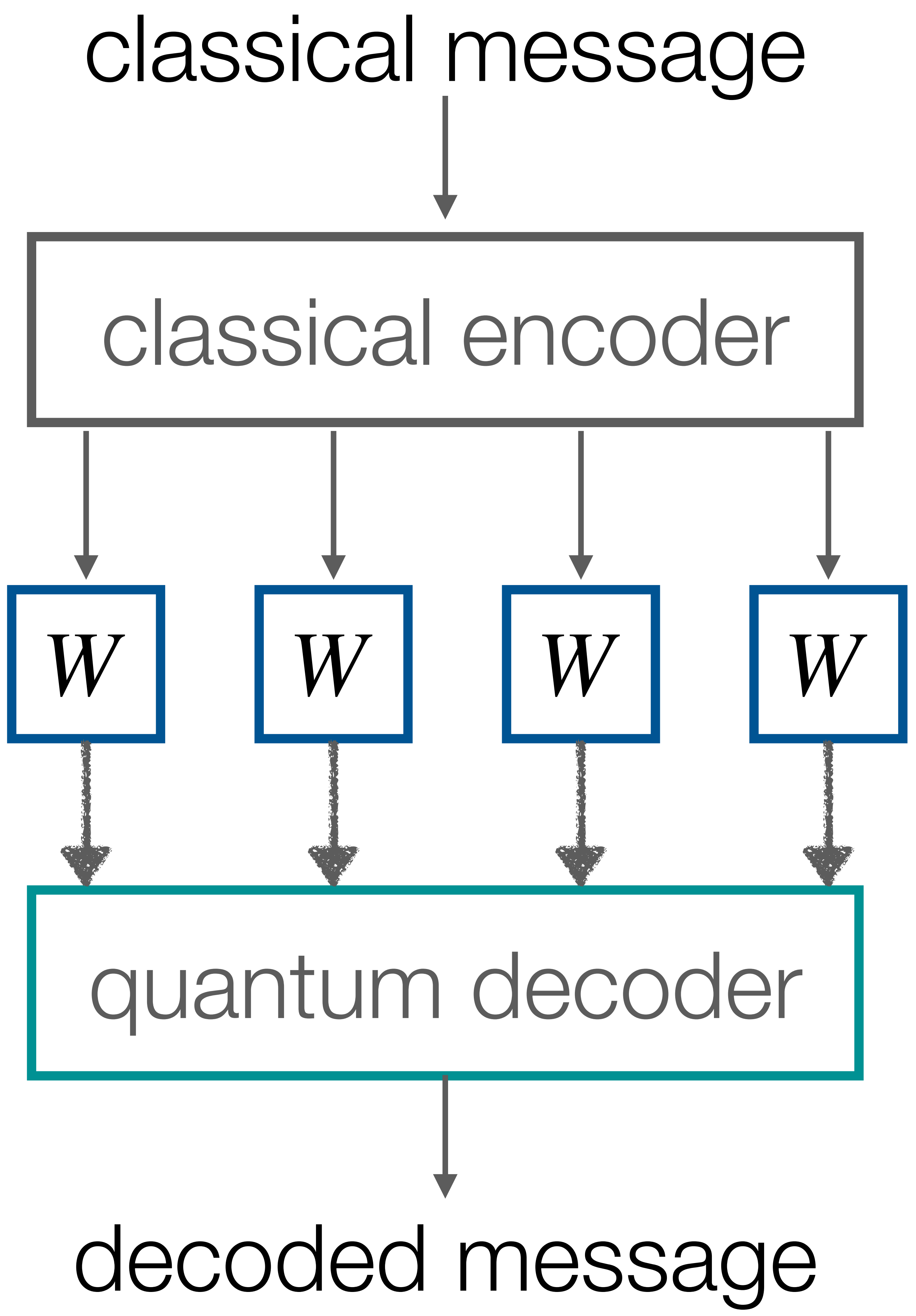
Poulin & Bilgin, PRA 2008

Poulin & Hastings, PRL 2011

Outline

- Factor graphs and belief propagation
- BPQM: Passing quantum messages for single bit estimation
- Successive BPQM for entire codewords
- Applications: Efficient decoders at capacity
- Summary and open questions

BP for decoding quantum codes



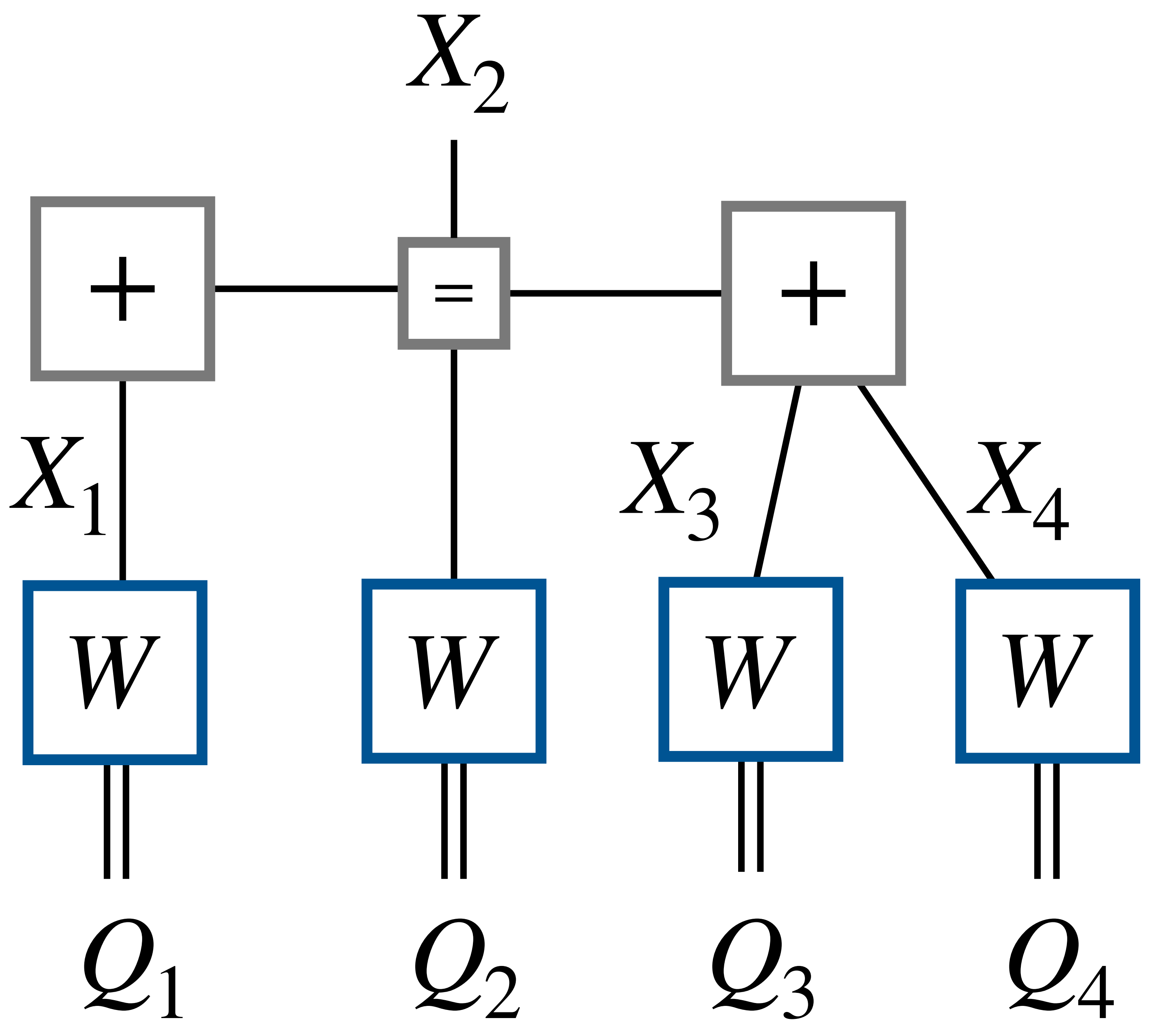
uniformly random

linear code

CQ channel

decode bitwise

BP for decoding quantum codes



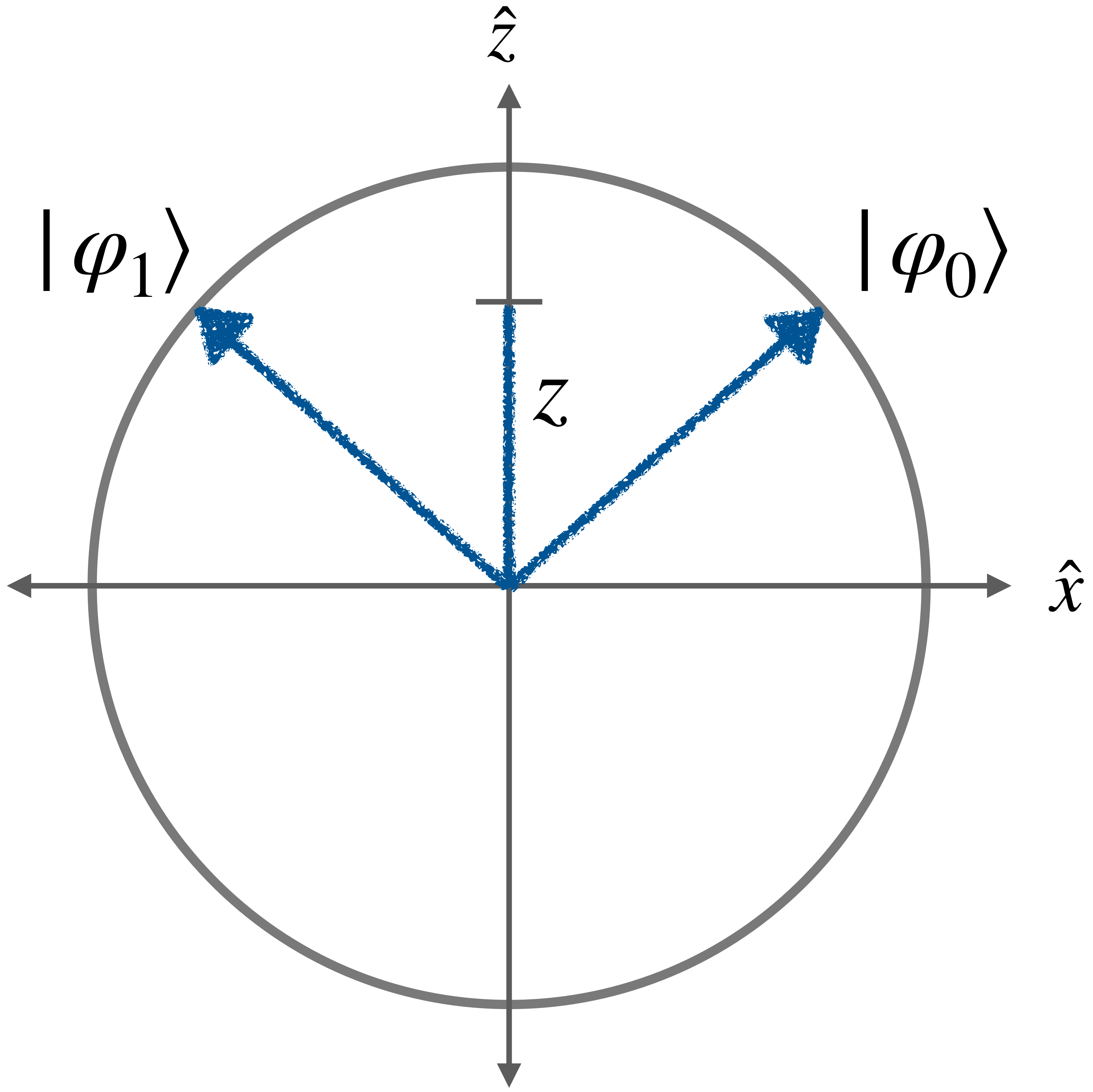
Pick the simplest possible quantum extension:

Channel with symmetric pure state outputs $|\varphi_x\rangle$

Need to construct a measurement to estimate X_2 from $Q_1Q_2Q_3Q_4$

Tensor network contraction is not enough!

CQ channel output description

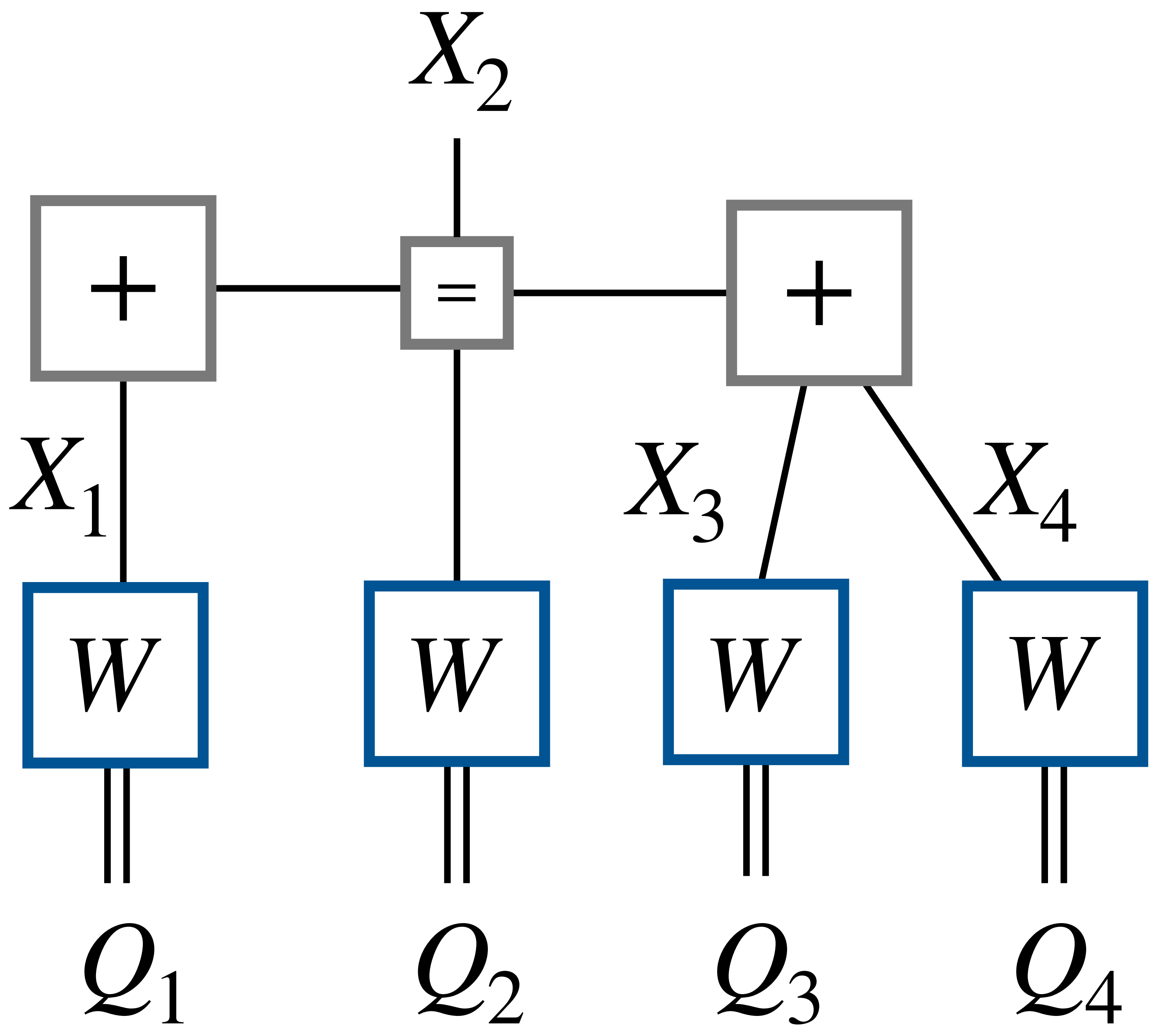


Bloch sphere

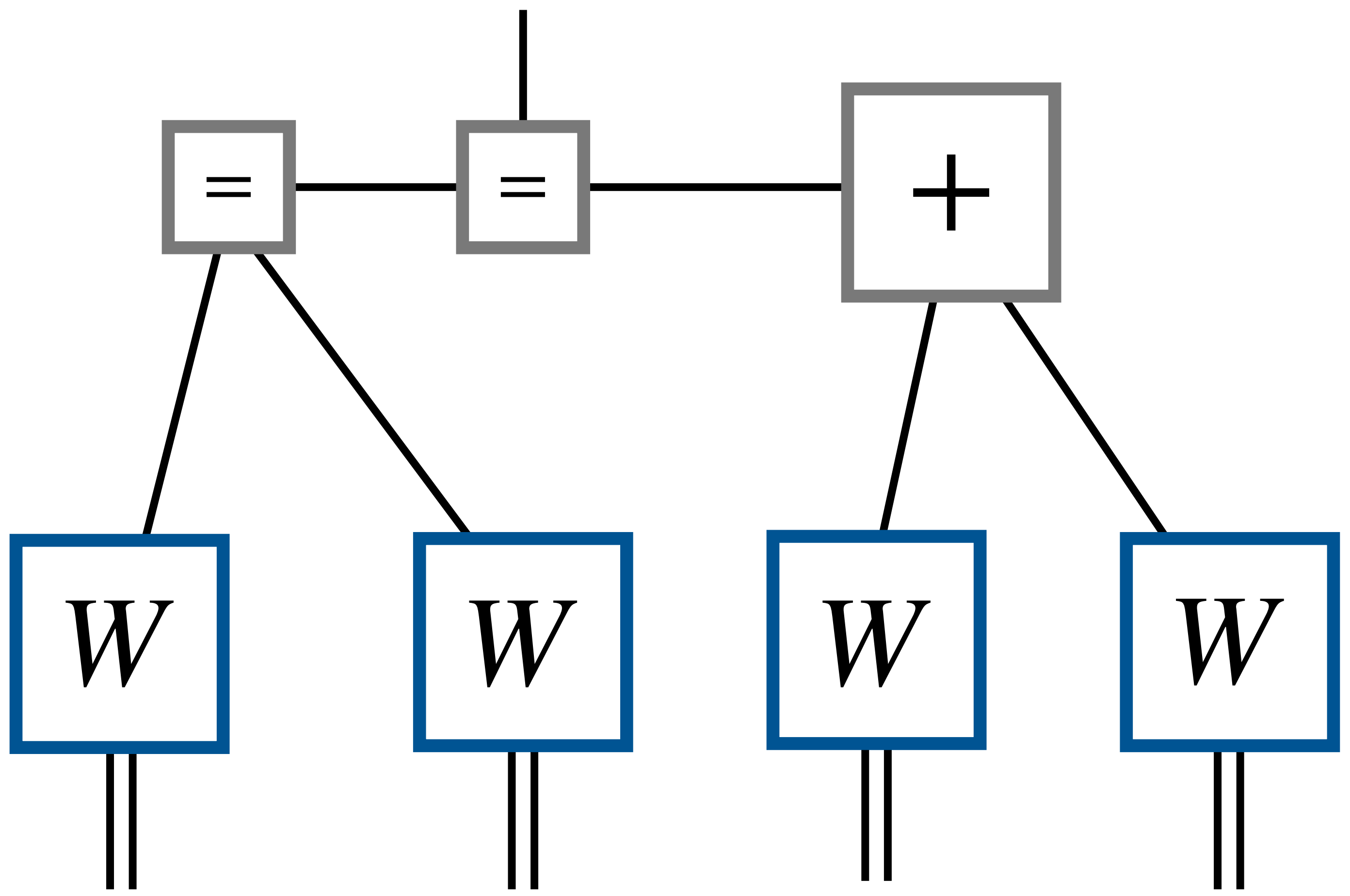
Bloch vector:

$$\hat{n} = z\,\hat{z} + (-1)^x \sqrt{1 - z^2}\ \hat{x}$$

Quantum message passing algorithm: BPQM

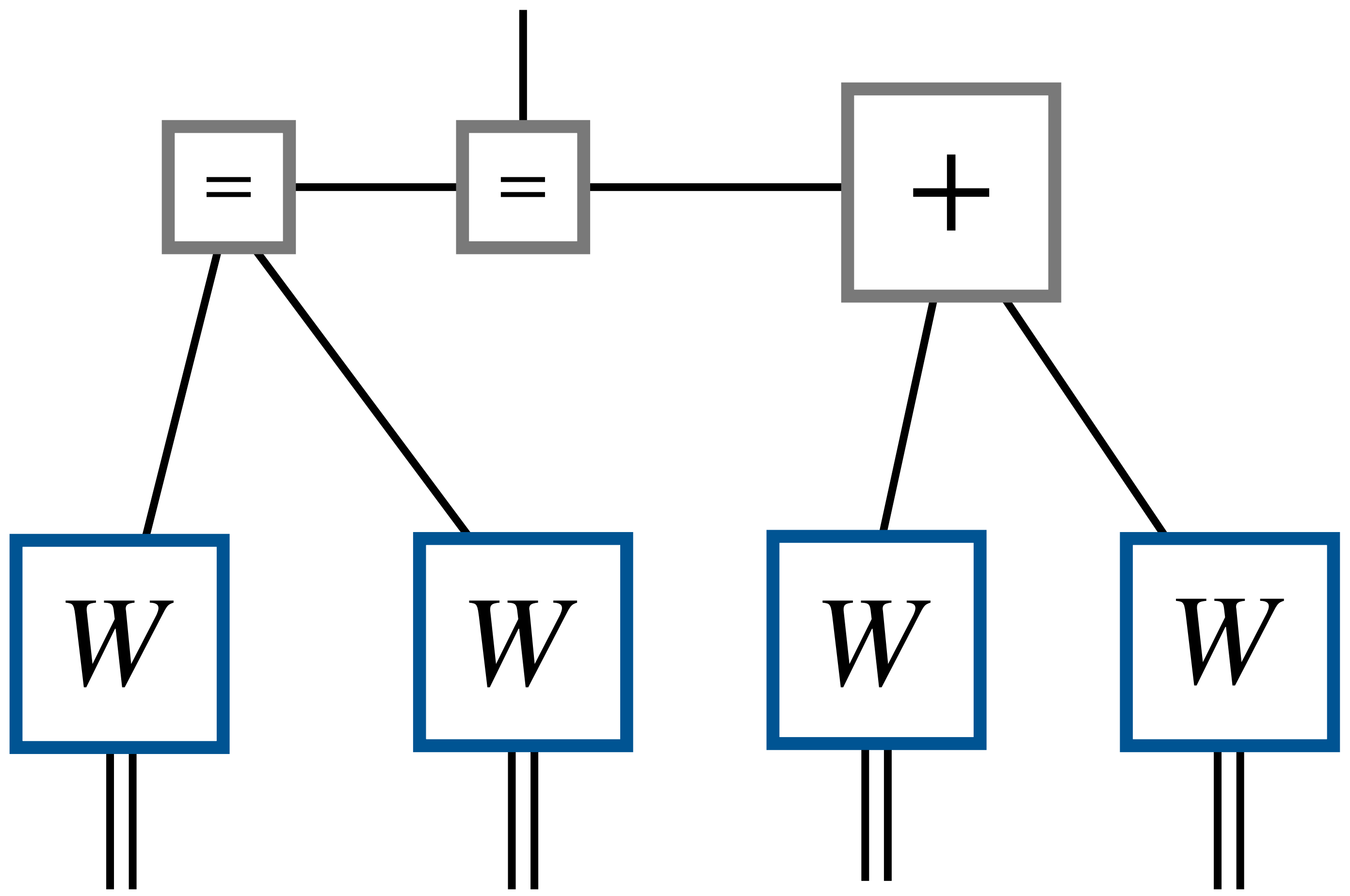


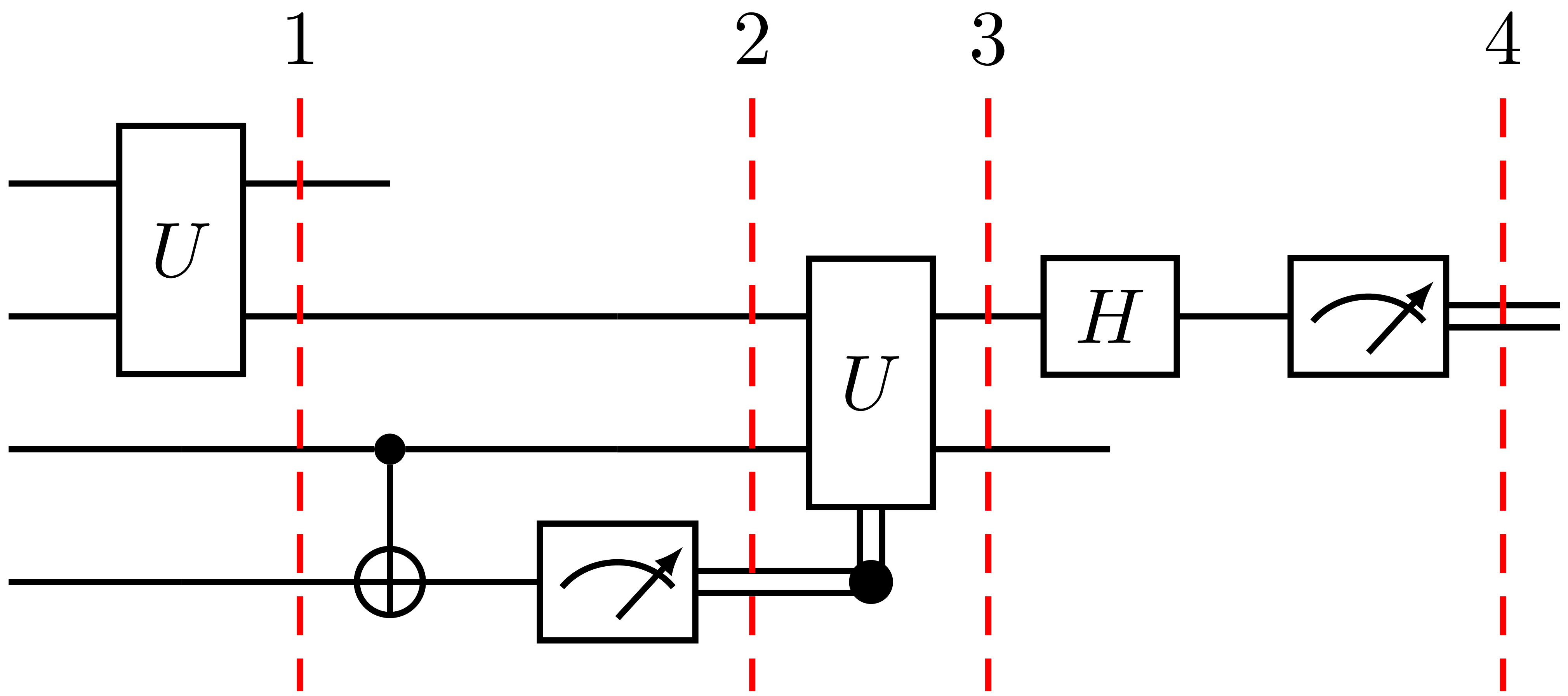
Quantum message passing algorithm: BPQM



- Traverse the tree from W leaves to root
- Associate a qubit and z parameter to each node
- At = nodes: Apply unitary $U(z_1, z_2)$, discard 2nd qubit. Set param to $z_1 z_2$.
- At + nodes: Apply CNOT, measure 2nd qubit $\rightarrow k$. Discard 2nd qubit.
  Reset $z_2 \leftarrow (-1)^k z_2$ and set param to $\dfrac{z_1 + z_2}{1 + z_1 z_2}$.
- Measure last qubit in $\hat{x}$ basis.

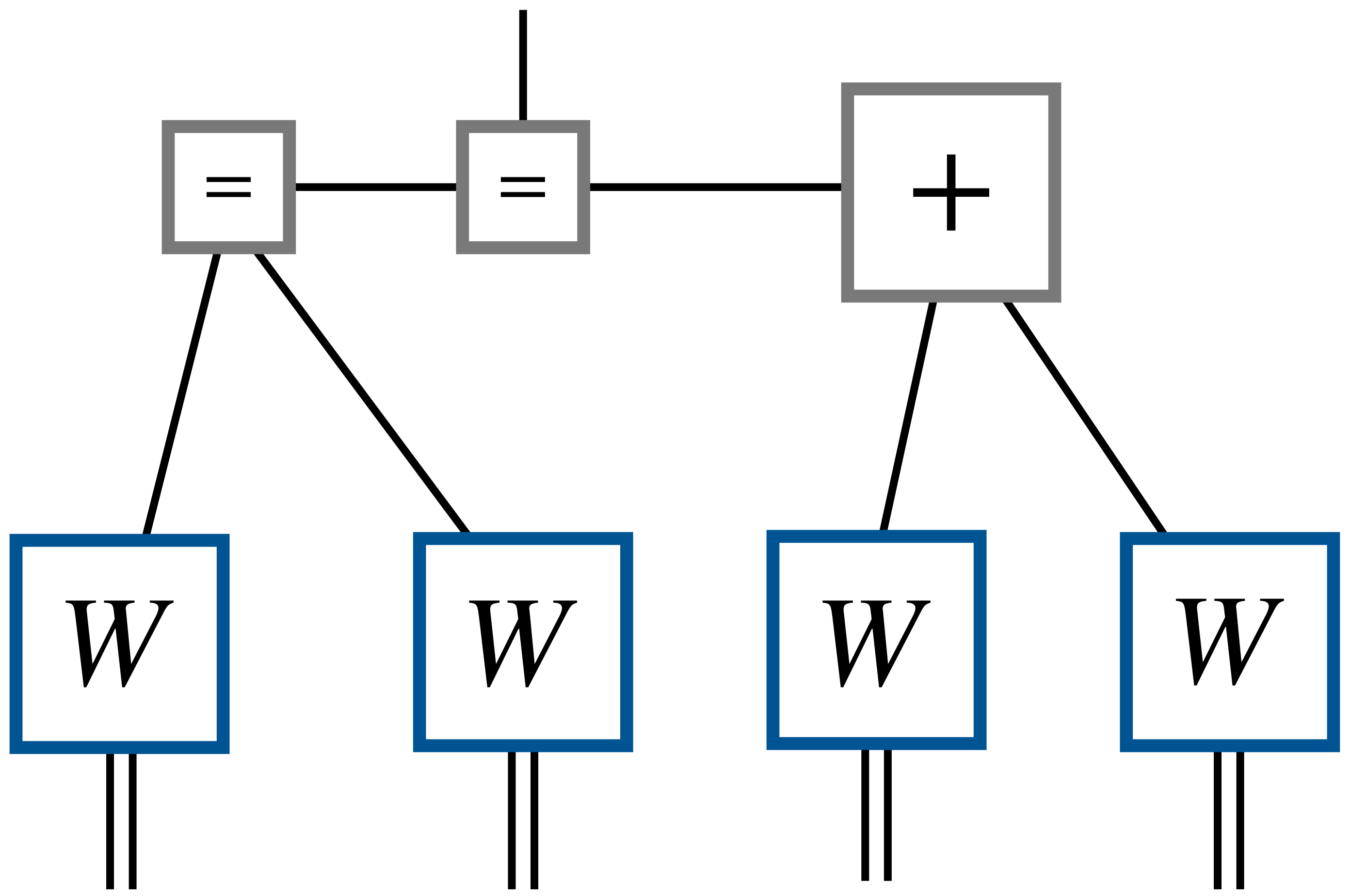
Quantum message passing algorithm: BPQM





- =: Apply unitary $U(z_1, z_2)$, discard 2nd qubit. Set param to $z_1 z_2$.
- +: Apply CNOT, measure 2nd qubit $\rightarrow k$. Discard 2nd qubit. Reset $z_2 \leftarrow (-1)^k z_2$ and set param to $\dfrac{z_1 + z_2}{1 + z_1 z_2}$.
- Measure last qubit in $\hat{x}$ basis.

BPQM implements the bitwise optimal measurement



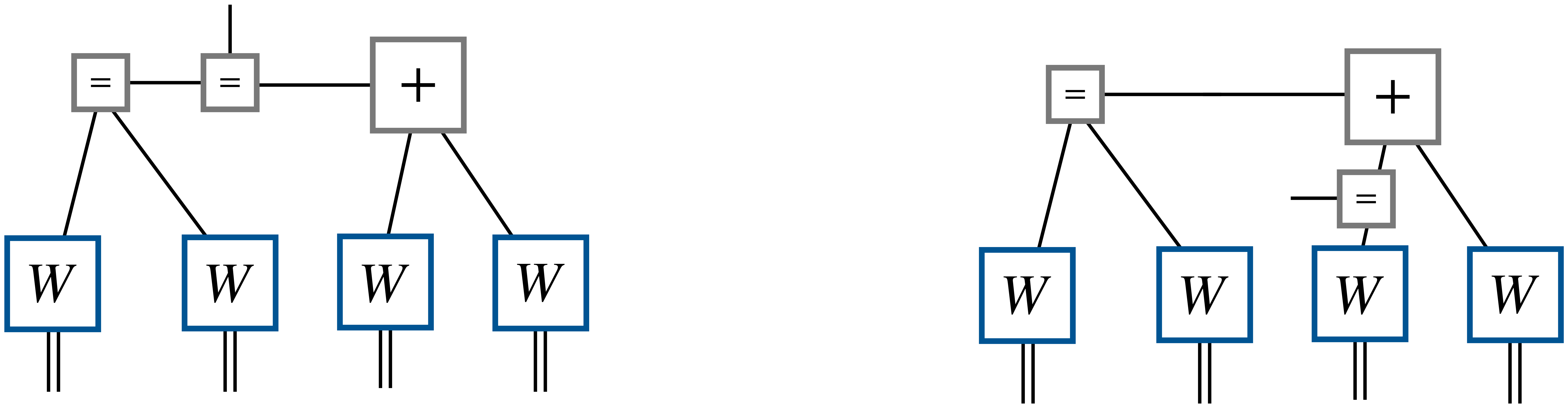
- Consider channel from any node to its leaves
- = node output: $|\varphi_0\rangle|\varphi_0'\rangle$ or $|\varphi_1\rangle|\varphi_1'\rangle$. Repackage into a single qubit with appropriate unitary: $U(z_1, z_2)$.
- + node output: After CNOT, the output becomes $\sum_{k\in\{0,1\}} p_k\,\varphi(k)_x \otimes |k\rangle\langle k|$, with the state parameter as in the algorithm and some probabilities p_k.
- Recursively simplify the factor graph

Outline

- Factor graphs and belief propagation
- BPQM: Passing quantum messages for single bit estimation
- Successive BPQM for entire codewords
- Applications: Efficient decoders at capacity
- Summary and open questions

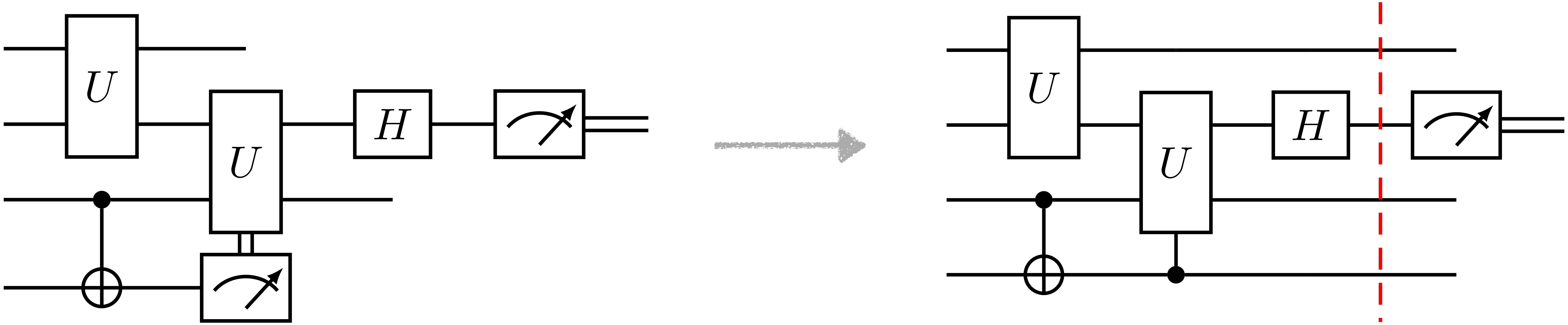
Successive BPQM for decoding entire codeword

- Run bitwise decoders sequentially
- Hope bitwise error probability low;
  use union bound for entire codeword.
- Will work for capacity-achieving codes!



Successive BPQM for decoding entire codeword

- Problem: Intermediate measurements.
  Solution: Perform BPQM coherently.
  Rewind the circuit after measuring the output qubit.



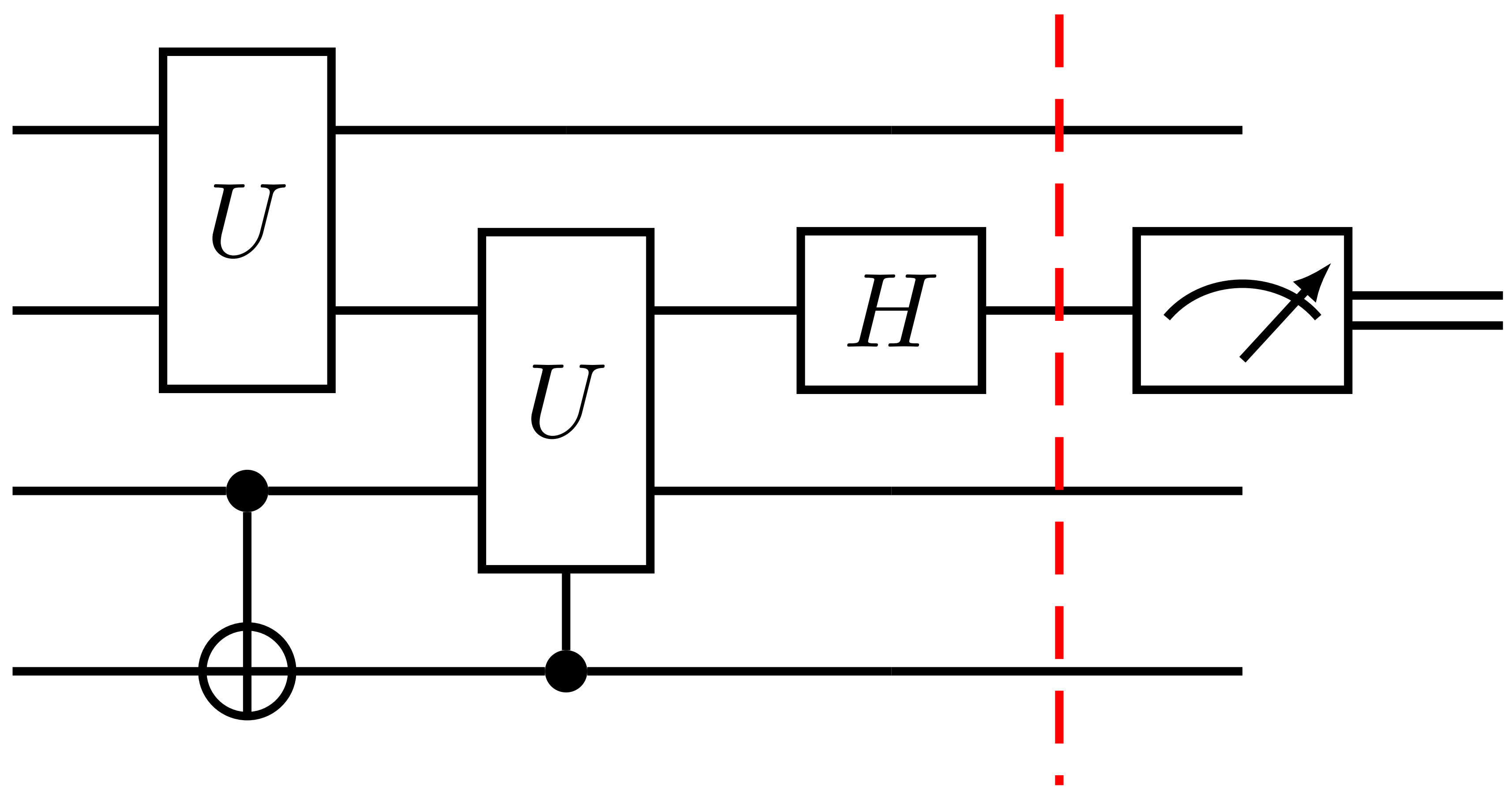
Successive BPQM for decoding entire codeword

- Problem: Intermediate measurements.
  Solution: Perform BPQM coherently.
  Rewind the circuit after decoding each bit.
- Problem: Exponential overhead from + controls.
  Solution: Quantize z register. Uncompute after use.
- Problem: Need infinite dimensions.
  Solution: Discretize to finite precision.
  For target error ε, register size only $O(\log 1/\varepsilon)$.
- All messages passed are now quantum!

BPQMv2: Blockwise optimality

BPQMv2: Adjust measured qubit before rewinding & use updated factor graph

Rengaswamy *et al*. arXiv:2003.04356



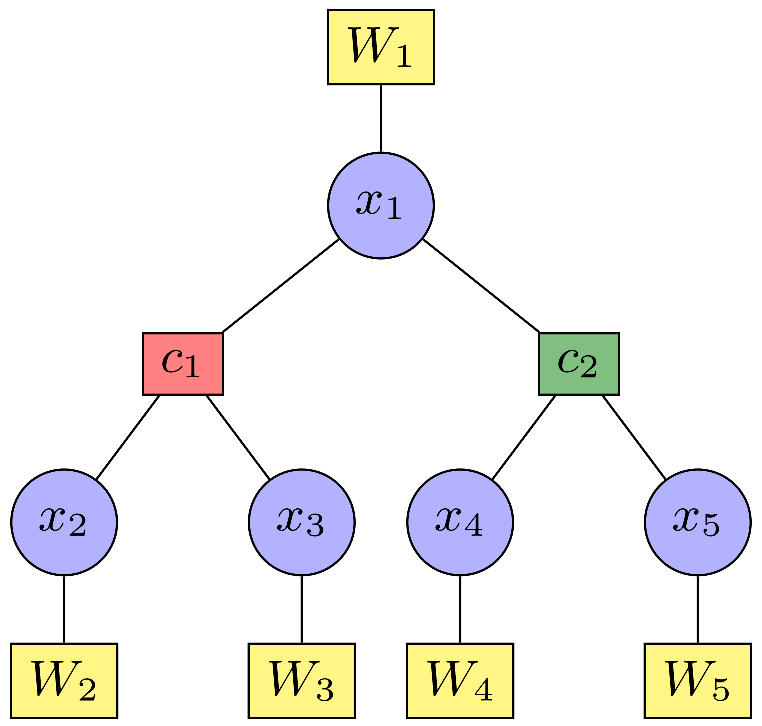
BPQMv2: Blockwise optimality

BPQMv2: Adjust measured qubit before rewinding & use updated factor graph

Rengaswamy *et al*. arXiv:2003.04356

Advantages:

1. Simplifies decoding of subsequent bits.
2. Appears to implement the block optimal measurement!



$$H = \begin{array}{c c} & \begin{array}{ccccc} x_1 & x_2 & x_3 & x_4 & x_5 \end{array} \\ \begin{array}{c} c_1 \\ c_2 \end{array} & \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array}$$

FIG. 2. Factor graph and parity-check matrix for the 5-bit linear code in the running example.

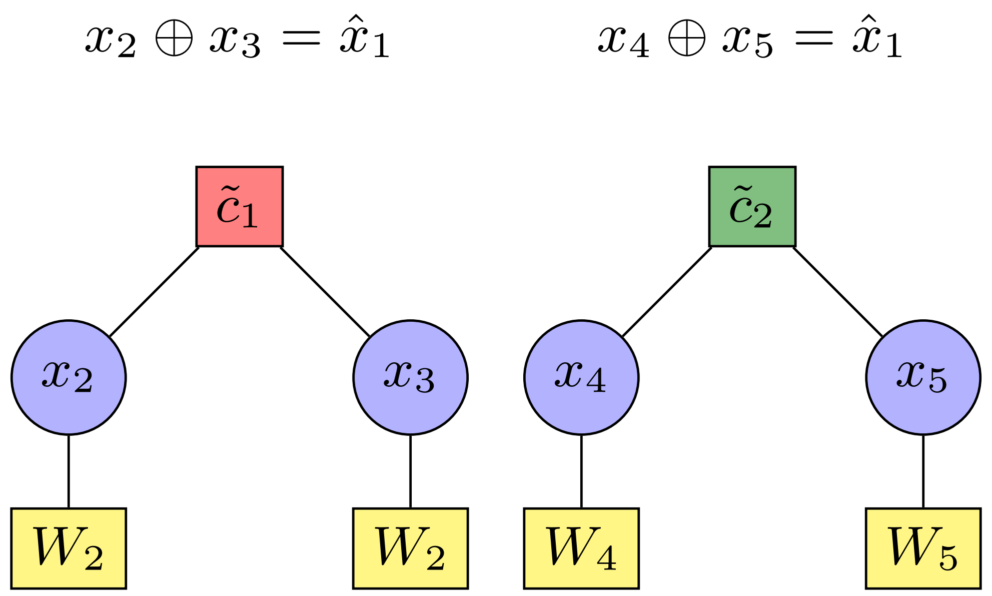
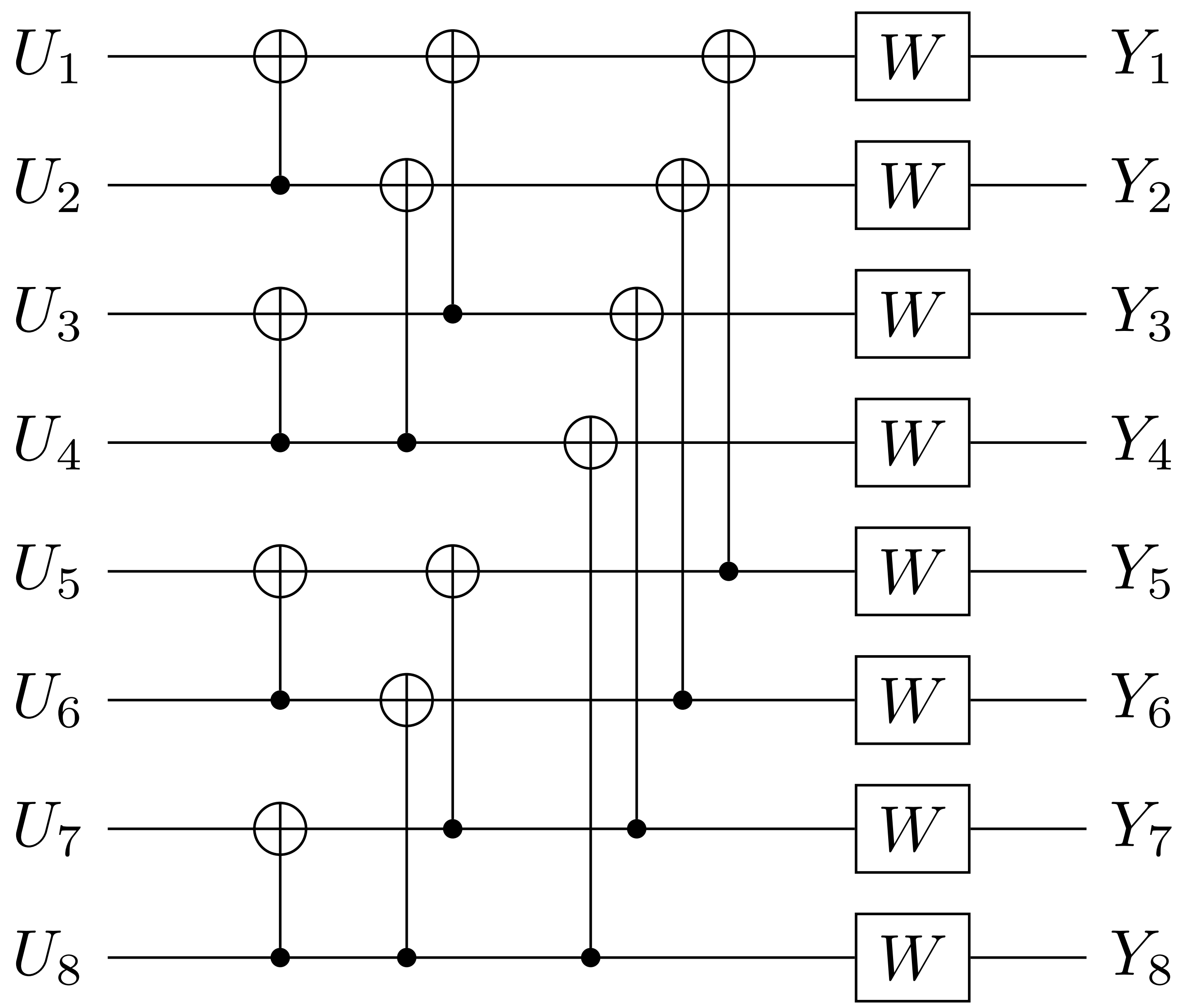


FIG. 7. The reduced factor graph after estimating bit 1 to be $\hat{x}_1$.

Outline

- Factor graphs and belief propagation
- BPQM: Passing quantum messages for single bit estimation
- Successive BPQM for entire codewords
- Applications: Efficient decoders at capacity
- Summary and open questions

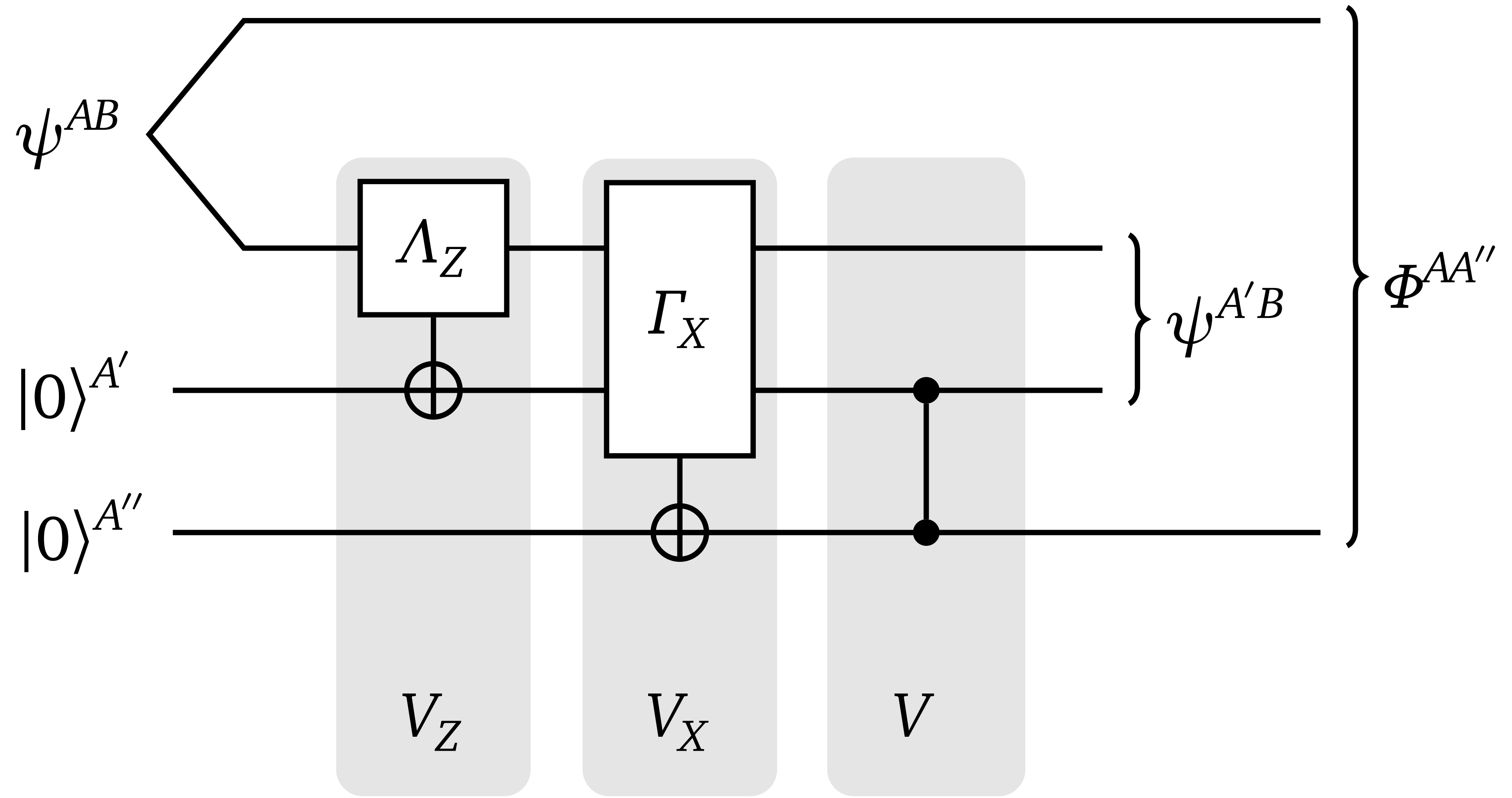
Applications of BPQM: Classical communication



- Polar codes have a tree structure
- BPSK on pure loss Bosonic channel
- BPQM: BPSK-capacity-achieving decoder

Applications of BPQM: Quantum communication

- Quantum polar codes use classical-input polar codes as subroutines
- Classical "pieces" of amplitude damping:
    1. Classical Z channel
    2. Heralded pure state output channel
- BPQM gives an efficient capacity-achieving decoder!



Outline

- Factor graphs and belief propagation
- BPQM: Passing quantum messages for single bit estimation
- Successive BPQM for entire codewords
- Applications: Efficient decoders at capacity
- Summary and open questions

Summary & Open questions

- BPQM: efficient bitwise-optimal quantum message passing decoder
- BPQMv2: Blockwise optimal?
- LDPC codes?
- BPQM for factor graphs with loops?
- BPQM for mixed state output channels?