Trapping Sets of **Iterative Decoders** for Quantum Low-Density Parity Check Codes

Bane Vasić
Nithin Raveendran

Error Correction Laboratory
Center for Quantum Networks
Superconducting Quantum Materials and Systems Center
Department of ECE
University of Arizona

Funded by the NSF under Grants:
NSF-ERC 1941583, CIF-1855879, CIF-2106189, CCF-2100013, and ECCS/CCSS-2027844

THE UNIVERSITY OF ARIZONA.
TUCSON ARIZONA

More in our recent paper

N. Raveendran and B. Vasic, "Trapping Sets of Quantum LDPC Codes," Quantum 5, 562, Oct. 2021. also at arXiv:2012.15297 [cs.IT]

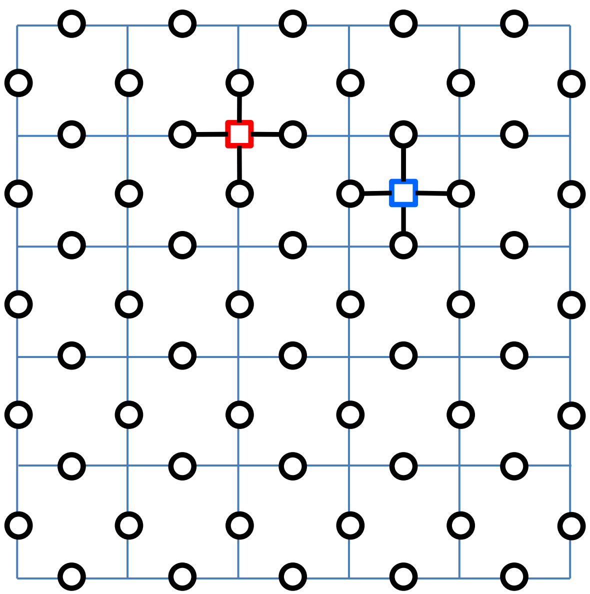
NITHIN RAVEENDRAN

XIN XIAO

ANKUR RAINA

NARAYANAN RENGASWAMY

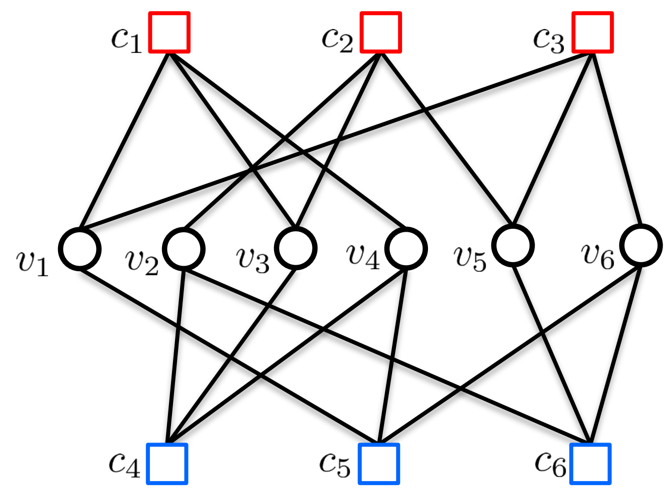
Surface codes and LDPC codes



- vertex checks (H_X)
- plaquette checks (H_Z)

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

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



Why quantum LDPC codes?

- Renewed interest in quantum LDPC (QLDPC) codes
  - Promises fault tolerant computation with constant overhead [15-17]
- Decoded efficiently using low-complexity iterative decoding [18]
- Involve stabilizer (parity) checks of bounded and low weight

[15] D. Gottesman, "Fault-Tolerant Quantum Computation with Constant Overhead," arXiv:1310.2984, 2014.

[16] A. Kovalev and L. Pryadko, "Improved quantum hypergraph-product LDPC codes," in 2012 IEEE ISIT, July 2012, pp. 348–352.

[17] O. Fawzi, A. Grospellier, and A. Leverrier, "Constant overhead quantum fault-tolerance with quantum expander codes," arXiv:1808.03821, 2018

[18] O. Fawzi, A. Grospellier, and A. Leverrier, "Efficient decoding of random errors for quantum expander codes," arXiv:1711.08351, 2017

Why quantum LDPC codes?

- Finite (nonzero) asymptotic rate
  - Surface codes: As code length $n \rightarrow \infty$, surface code rate $\rightarrow 0$.
- Minimum distance scaling better than square root of the code length [19-21]. Better than surface codes.

[19] P. Panteleev and G. Kalachev, "Quantum LDPC codes with almost linear minimum distance," arXiv:2012.04068, 2020.

[20] N. P. Breuckmann and J. N. Eberhardt, "Balanced Product Quantum Codes", arXiv: 2012.09271, 2020.

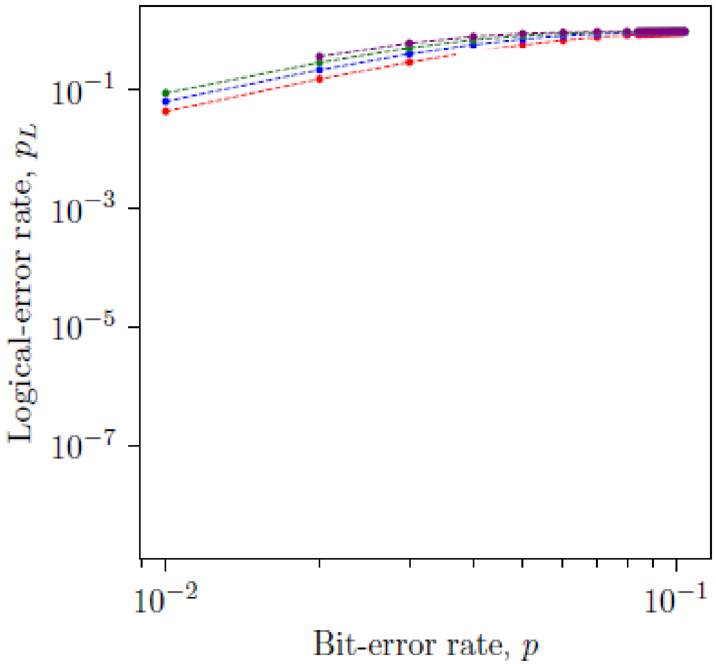
[21] P. Panteleev and G. Kalachev, "Asymptotically Good Quantum and Locally Testable Classical LDPC Codes", arXiv:2111.03654, 2021.

QLDPC decoding problem is still open

- The existing QLDPC code literature primarily focuses on:
  - constructing asymptotically good code families with improved d_{min} scaling with the block length n and higher code rates R
  - designing better iterative decoding algorithms
- QLDPC codes implemented in practical QEC systems will be of finite length and their iterative decoders will propagate finite-precision messages.
  - Performance degradation due to convergence issues
- The convergence failure manifests itself as an error floor of the decoding probability of error at low physical error rate levels.
  - Error floor is observed in all state-of-the-art iterative message-passing decoders: bit-flipping, belief propagation (BP), min-sum algorithm (MSA) and their variants

Example

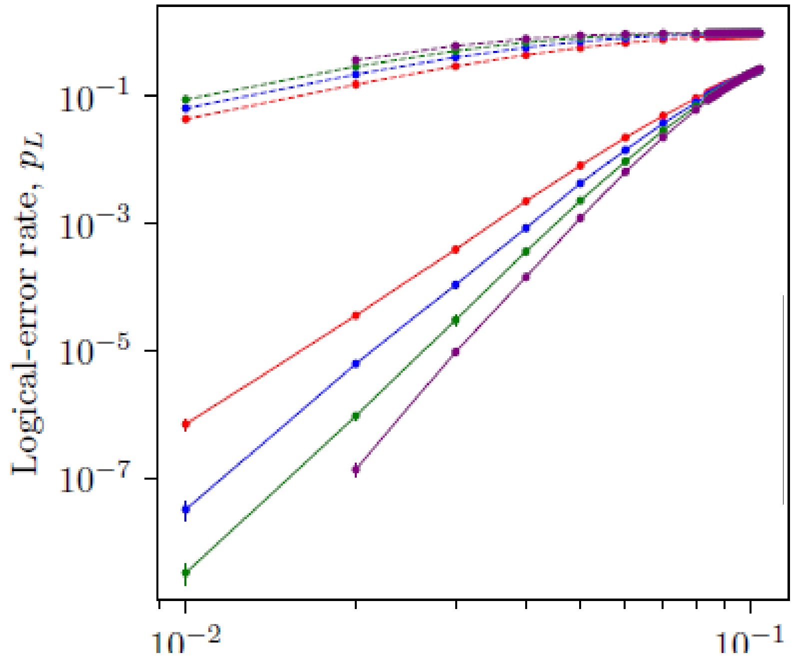
- BP performance (logical error rate) on Toric codes with $k = 2$ and d_{min}: 9, 11, 13, 15.
- Major drawback of BP is inability to handle small row weight of the parity-check matrix present due to the stabilizer commutativity constraint.



Roffe et al. May 2020. arXiv:2005.07016 [quant-ph]

BP with post-processing

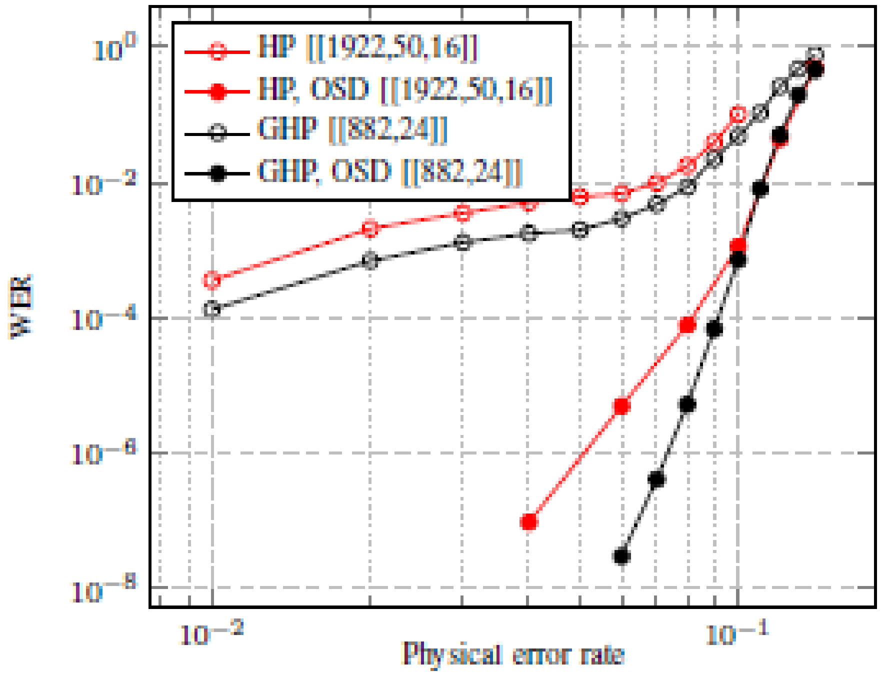
- A typical way to deal with error floor is post-processing with the Ordered Statistics Decoding (OSD).



Roffe et al. May 2020. arXiv:2005.07016 [quant-ph]

BP with post-processing

- However, the complexity of OSD is exponential in the code dimension k. In addition to the BP complexity.



P. Panteleev and G. Kalachev, arXiv:1904.02703, 2019.

Our general objective

- Achieve BP-OSD performance but use message-passing, i.e. only local processing.

General objectives of our research

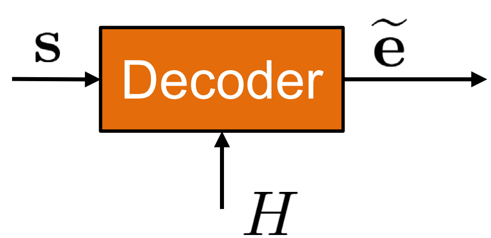
- For a given QLDPC code, design a decoder that will guarantee correction of all error patterns up to weight t.
- For a given decoder, construct QLDPC code that guarantees correction of all error patterns up to weight t.
- Understand precisely what causes error floor:
  - *Degenerate errors*: errors with the same non-zero syndrome
  - *Short cycles: in the Tanner graph: quantum code imposes s*ymplectic product/commutativity constraints on Tanner graph
- Other constraints
  - Convergence
  - Complexity
  - Fault tolerance

Our current projects

- Classical projects:
  - NSF CCF-2100013 Small: Learning To Correct Errors
  - NSF ECCS/CCSS-2027844: Neural Network Nonlinear Iterative LDPC Decoders with Guaranteed Error Performance and Fast Convergence
  - NSF-CCSS-2052751: Collaborative Research: Secure and Efficient Post-quantum Cryptography: from Coding Theory to Hardware Architecture

The problem setting

- A $[\![n, \ell, d]\!]$ stabilizer code given by the $m \times n$ binary parity check matrix $H = \begin{pmatrix} H_\mathrm{X} & H_\mathrm{Z} \end{pmatrix}$, where
$$H_\mathrm{X} H_\mathrm{Z}^\mathrm{T} + H_\mathrm{Z} H_\mathrm{X}^\mathrm{T} = \mathbf{0}$$
- Channel is a depolarizing channel. On each qubit, flips occur independently with probability α , resulting in the error pattern $\mathbf{e} = (\mathbf{e}_\mathrm{X}, \mathbf{e}_\mathrm{Z})$.
- Syndrome measurement of the quantum state $E\,|\psi\rangle$ results in the syndrome $\mathbf{s}$.
- Decoder is a quantum syndrome decoder.



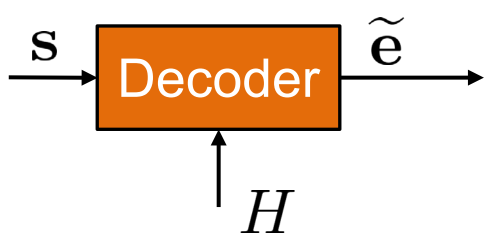
THE UNIVERSITY OF ARIZONA. TUCSON ARIZONA

Decoders for CSS codes

- The parity check matrix is in the form (the rows are linearly independent)

$$H = \begin{pmatrix} H_{\mathrm{X}} & 0 \\ 0 & H_{\mathrm{Z}} \end{pmatrix} \qquad H_{\mathrm{X}} H_{\mathrm{Z}}^{\mathrm{T}} = 0$$

- The errors $\mathbf{e}_{\mathrm{X}}$ and $\mathbf{e}_{\mathrm{Z}}$ are independent, thus two decoders - one operating on H_{Z} to correct $\mathbf{e}_{\mathrm{X}}$ and one operating on H_{X} to correct $\mathbf{e}_{\mathrm{Z}}$ can be independently run.



CSS codes with H_X=H_Z

- The two codes can be chosen to be the same

$$H_{\mathrm{X}} = H_{\mathrm{Z}} = H$$

- The condition $HH^{\mathrm{T}} = 0$ means that the code $\mathcal{C}$ is dual-containing $\mathcal{C}^{\perp} \subset \mathcal{C}$.
- Codewords of $\mathcal{C}$ include the rowspace of *H* but also 2^{ℓ} other codewords.

$$\ell = \dim(\mathcal{C}) - \dim(\mathcal{C}^{\perp})$$

$$G = \begin{pmatrix} H_{(n-k)\times n} \\ L_{\ell\times n} \end{pmatrix}$$

Example (Steane Code)

$\mathcal{C}$

0 0 0 0 0 0 0
1 0 1 0 1 0 1
0 1 1 0 0 1 1
1 1 0 0 1 1 0
0 0 0 1 1 1 1
1 0 1 1 0 1 0
0 1 1 1 1 0 0
1 1 0 1 0 0 1
1 1 1 0 0 0 0
0 1 0 0 1 0 1
1 0 0 0 0 1 1
0 0 1 0 1 1 0
1 1 1 1 1 1 1
0 1 0 1 0 1 0
1 0 0 1 1 0 0
0 0 1 1 0 0 1

$$\longleftarrow \quad G = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

$$H = G^{\perp} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \longrightarrow$$

$\mathcal{C}^{\perp}$

0 0 0 0 0 0 0
1 0 1 0 1 0 1
0 1 1 0 0 1 1
1 1 0 0 1 1 0
0 0 0 1 1 1 1
1 0 1 1 0 1 0
0 1 1 1 1 0 0
1 1 0 1 0 0 1

$\text{rowspace}(H)$

Example (Steane Code) – dual code

$\mathcal{C}$

0 0 0 0 0 0 0
1 0 1 0 1 0 1
0 1 1 0 0 1 1
1 1 0 0 1 1 0
0 0 0 1 1 1 1
1 0 1 1 0 1 0
0 1 1 1 1 0 0
1 1 0 1 0 0 1
1 1 1 0 0 0 0
0 1 0 0 1 0 1
1 0 0 0 0 1 1
0 0 1 0 1 1 0
1 1 1 1 1 1 1
0 1 0 1 0 1 0
1 0 0 1 1 0 0
0 0 1 1 0 0 1

$$\longleftarrow \quad G = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

$$H = G^{\perp} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \longrightarrow$$

$\mathcal{C}^{\perp}$

0 0 0 0 0 0 0
1 0 1 0 1 0 1
0 1 1 0 0 1 1
1 1 0 0 1 1 0
0 0 0 1 1 1 1
1 0 1 1 0 1 0
0 1 1 1 1 0 0
1 1 0 1 0 0 1

$\mathrm{rowspace}(H)$

Example (Steane Code) - cosets

$\mathcal{C}$

0 0 0 0 0 0 0
1 0 1 0 1 0 1
0 1 1 0 0 1 1
1 1 0 0 1 1 0
0 0 0 1 1 1 1
1 0 1 1 0 1 0
0 1 1 1 1 0 0
1 1 0 1 0 0 1
1 1 1 0 0 0 0
0 1 0 0 1 0 1
1 0 0 0 0 1 1
0 0 1 0 1 1 0
1 1 1 1 1 1 1
0 1 0 1 0 1 0
1 0 0 1 1 0 0
0 0 1 1 0 0 1

logical 0

$(0000000) + \text{rowspace}(H)$

logical 1

$(1110000) + \text{rowspace}(H)$

$\mathcal{C}^{\perp}$

0 0 0 0 0 0 0
1 0 1 0 1 0 1
0 1 1 0 0 1 1
1 1 0 0 1 1 0
0 0 0 1 1 1 1
1 0 1 1 0 1 0
0 1 1 1 1 0 0
1 1 0 1 0 0 1

$\text{rowspace}(H)$

Quantum decoder is a coset decoder

- Let $\mathbf{e}$ be a non-zero error vector, resulting in a syndrome $\mathbf{s}$

$$\mathbf{s} = \mathbf{e}H^{\mathrm{T}} \neq \mathbf{0}$$

- As opposed to a *classical syndrome decoder* that tries to find $\mathbf{e}$ for a given observed syndrome, a valid output of a *quantum* decoder is any one of the vectors

$$\widetilde{\mathbf{e}} = \mathbf{e} + \mathbf{h}, \mathbf{h} \in \mathrm{rowspace(H)}$$

- When $\mathbf{e} + \widetilde{\mathbf{e}} \neq \mathbf{0}$, but

$$(\mathbf{e} + \widetilde{\mathbf{e}})H^{\mathrm{T}} = \mathbf{0}$$

$$\mathbf{e} + \widetilde{\mathbf{e}} \in \mathrm{rowspace}(L)$$

then the correction vector $\mathbf{e} + \widetilde{\mathbf{e}}$ is applied to flip bits in the (unobservable) quantum codeword resulting in another codeword, and a logical, undetectable, error occurs.

Checklist of statements to clarify

- What is a *depolarizing* channel, what the X and Z components of an error pattern $\mathbf{e} = (\mathbf{e}_{\mathrm{X}}, \mathbf{e}_{\mathrm{Z}})$ mean?
- Why the parity check matrix has 2*n* columns?

$$H = \begin{pmatrix} H_{\mathrm{X}} & 0 \\ 0 & H_{\mathrm{Z}} \end{pmatrix}$$

- Quantum origin of the condition $H_{\mathrm{X}} H_{\mathrm{Z}}^{\mathrm{T}} + H_{\mathrm{Z}} H_{\mathrm{X}}^{\mathrm{T}} = \mathbf{0}$
  - Discuss LDPC code constructions that satisfy such *commutativity* condition, construction of CSS codes too.
  - Constraints on row weights of *H* due implementation, d_{min}
- Why $\widetilde{\mathbf{e}} = \mathbf{e} + \mathbf{h}, \mathbf{h} \in \mathrm{rowspace(H)}$ is a valid decoding output, why quantum codewords are defined as cosets?
- Misscorrections in the general case $H = \begin{pmatrix} H_{\mathrm{X}} & H_{\mathrm{Z}} \end{pmatrix}$

Quantum questions

- How is it possible to obtain a syndrome without disturbing the quantum state corresponding to a codeword?
- How do we measure a syndrome?
- What are stabilizer codes?
- How do we construct quantum LDPC codes from classical ones?

Summary of a problem statement

- Given the generator matrix

$$G = \begin{pmatrix} H_{(n-k)\times n} \\ L_{\ell\times n} \end{pmatrix}$$

  and the syndrome *s*, find any error pattern such that

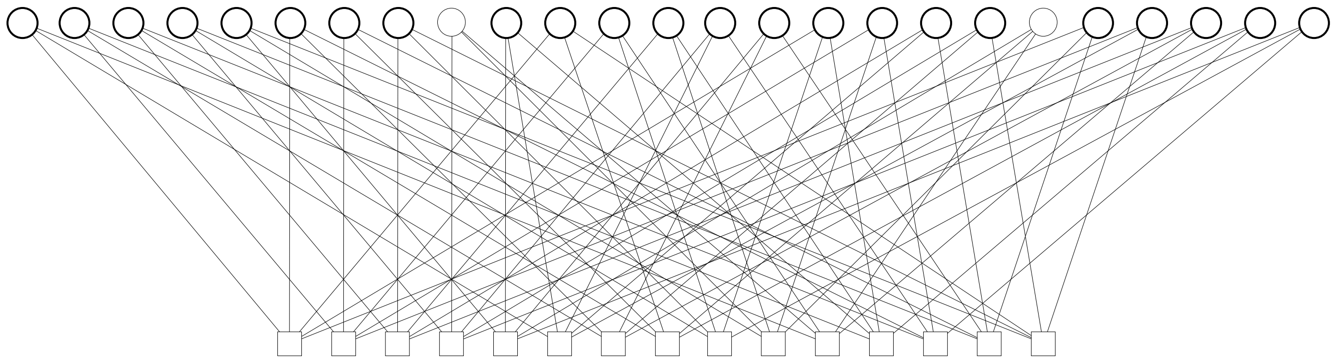
$$\mathbf{e} + \widetilde{\mathbf{e}} \in \mathrm{rowspace}(H) \text{ and } \mathbf{e} + \widetilde{\mathbf{e}} \notin \mathrm{rowspace}(L)$$

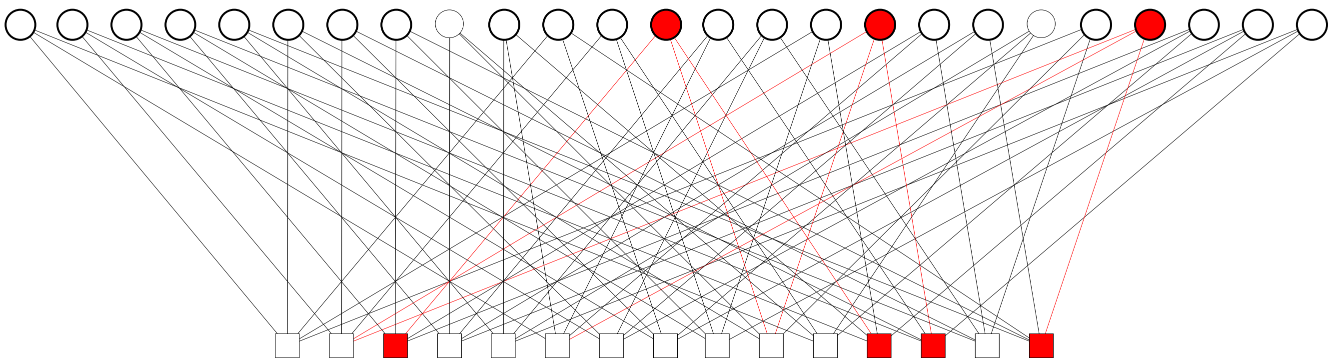
- H is a parity check matrix of an LDPC code
- Decoding is iterative, message passing decoding

Quantum syndrome decoding

- Whether a decoding algorithm $\mathcal{D}$ produces a valid correction vector depends on $\mathrm{wt}(\mathbf{e})$ and $d_{\min}$ of the code, but also on $\mathbf{e}$ and on $\mathcal{D}$.
- For *codes on graphs* (such as QLDPC codes, surface codes, etc.), and ability of a given *iterative* decoder $\mathcal{D}$ to correct an error pattern is determined by specific topologies of subgraphs found in $\mathcal{G} = \mathcal{G}(H)$, the Tanner graph of H.
- We refer to these subgraphs as trapping sets.
- Error floor is attributed to these dense subgraphs present in the Tanner graph.

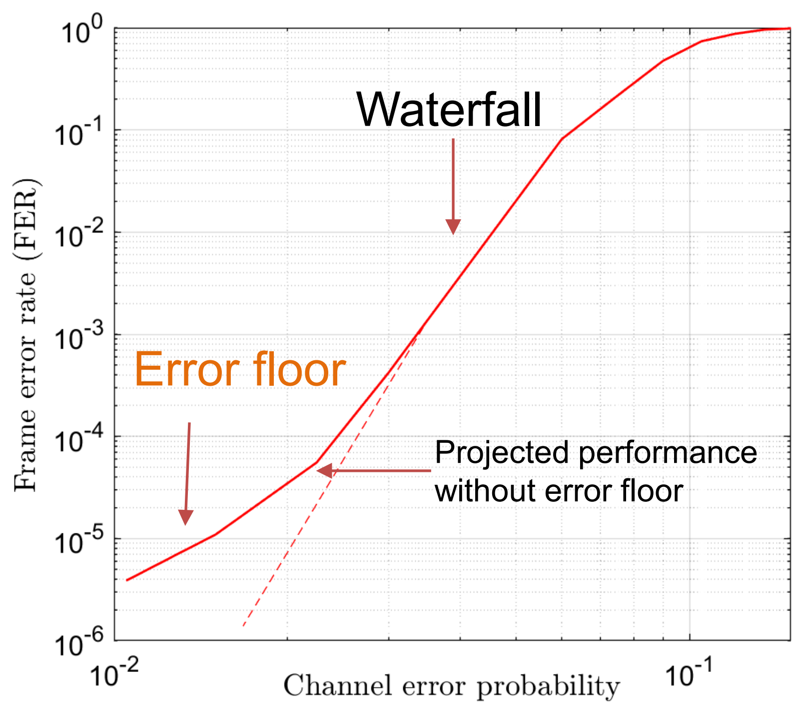
Tanner graph of H and syndrome matching





Typical performance curve of LDPC codes under iterative decoding

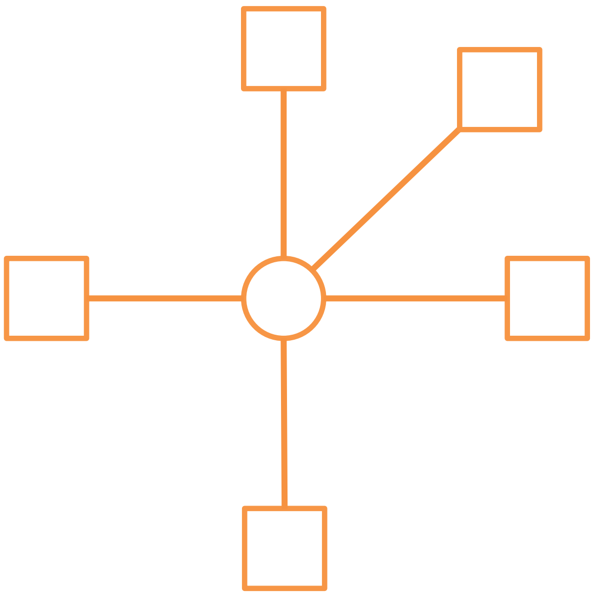
- Waterfall – Frame error rate *drops significantly* with decreasing noise
- Error floor - Abrupt degradation of frame error rate at low channel noise region - curve tends to *floor/flatten.*
  - Dense sub-graphs in the Tanner graph cause iterative decoder to fail for low-weight error patterns: trapping sets.

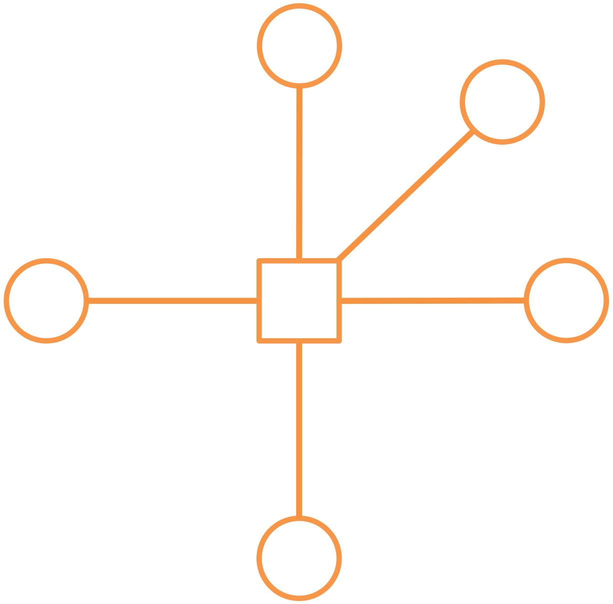


Outline of the talk

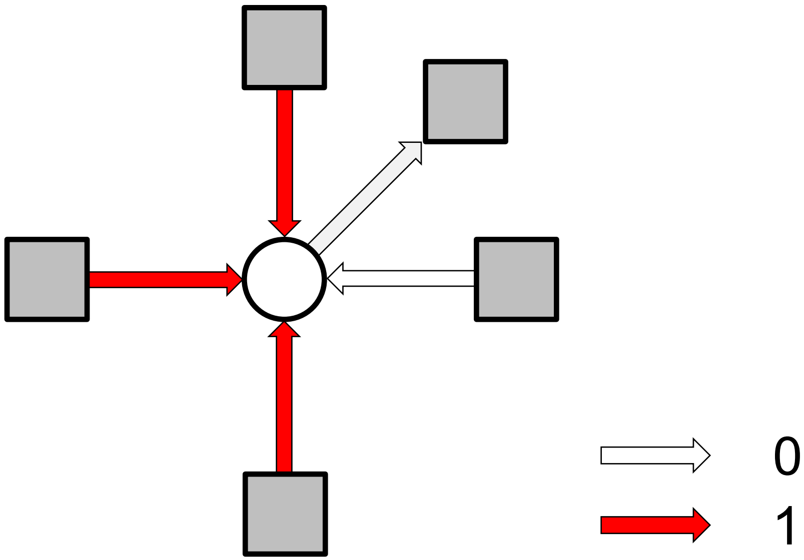
- Finite message precision iterative decoders
  - Gallager B decoder
- Understanding trapping sets in syndrome decoding
- Two types of trapping sets in QLDPC codes
  - Classical-like trapping sets
  - Trapping sets imbedded in symmetric stabilizers
- Using trapping sets to design better QLDPC codes and better decoders
- Enumeration of trapping sets in some known code families: bicycle codes and hypergraph product codes
- An efficient trapping set search algorithm

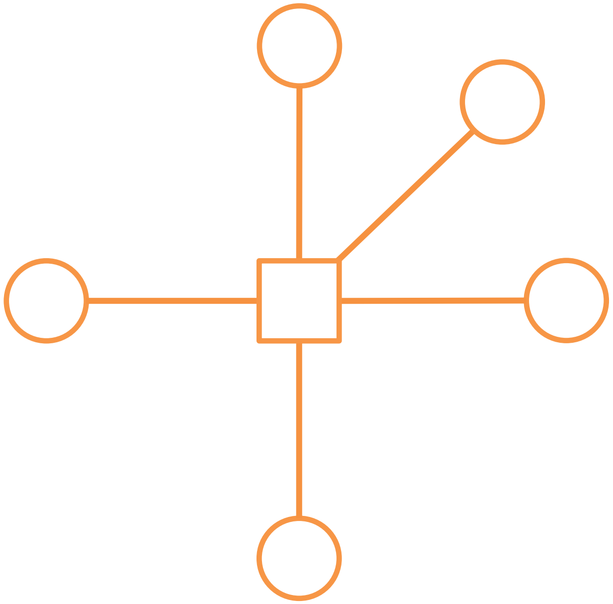
Gallager B algorithm





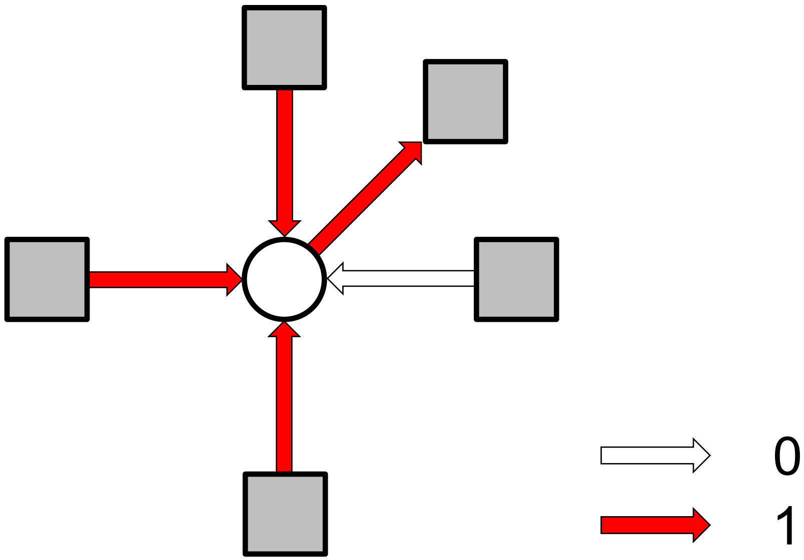
Gallager B
variable message update

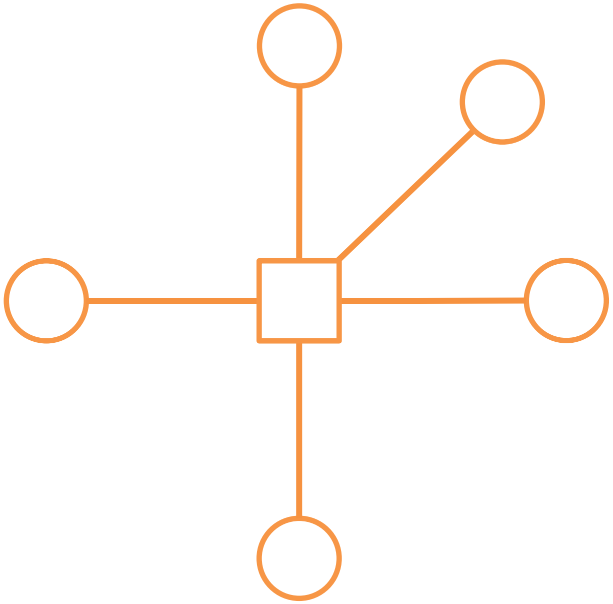




when there is majority

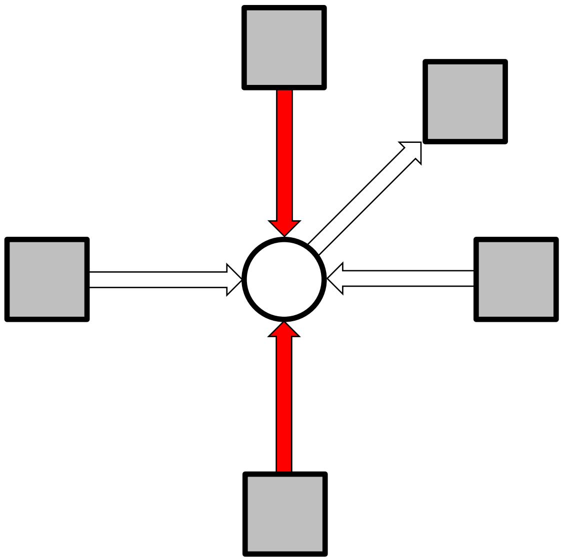
Gallager B
variable message update

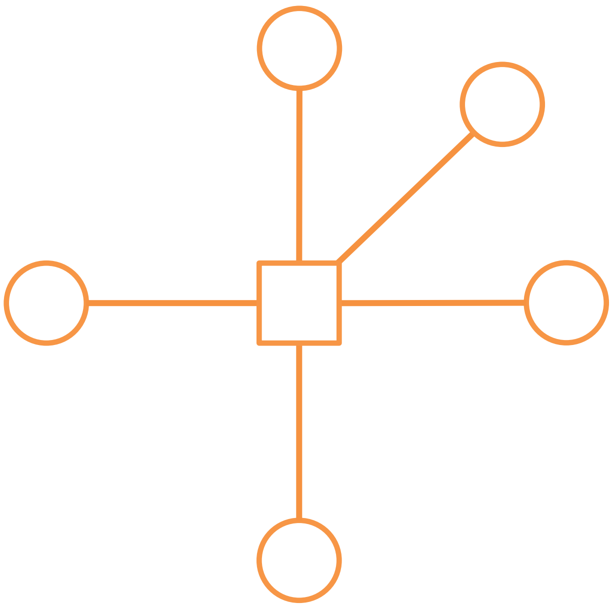




when there is majority, send the majority value

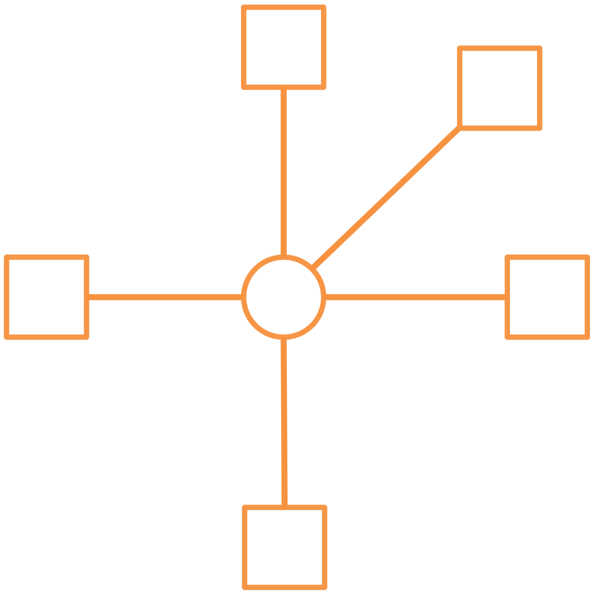
Gallager B
variable message update

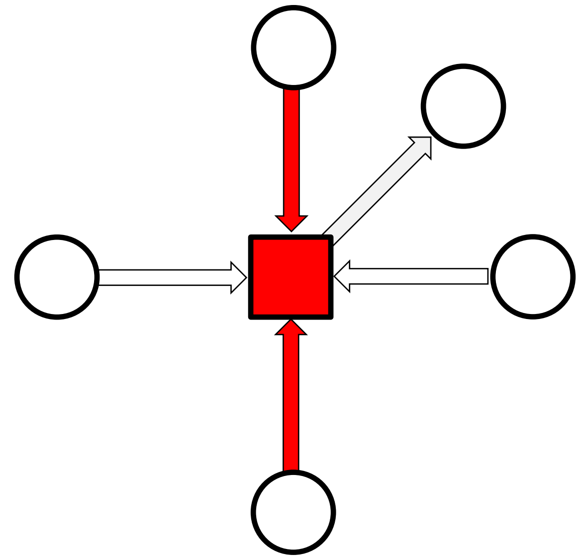




when there is a tie, send the zero value

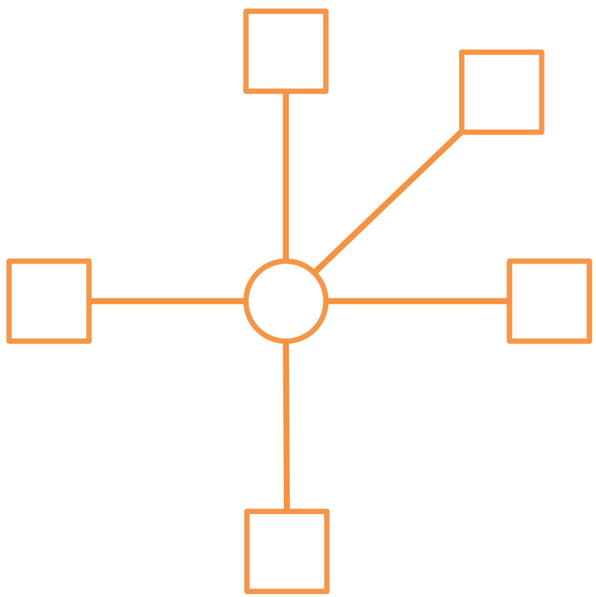
Gallager B
check message update

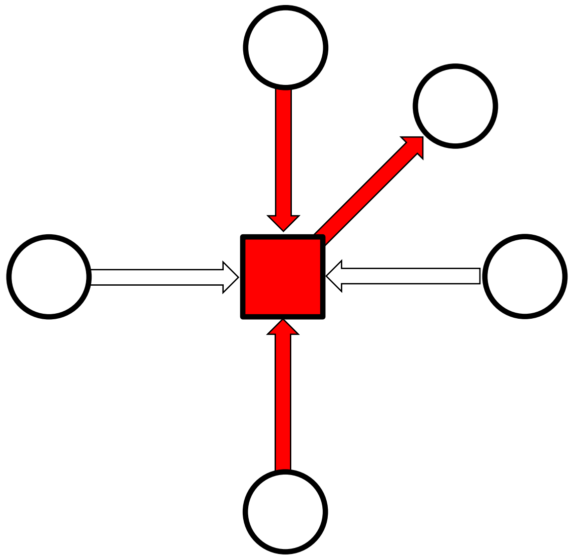




XOR the incoming messages and the check value

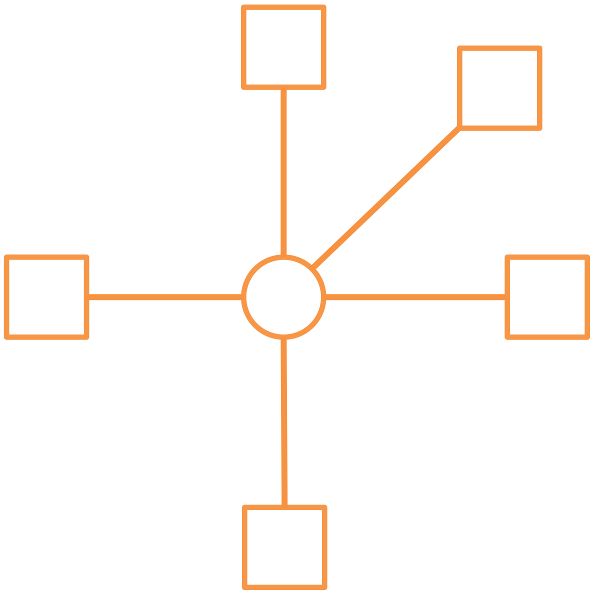
Gallager B
check message update

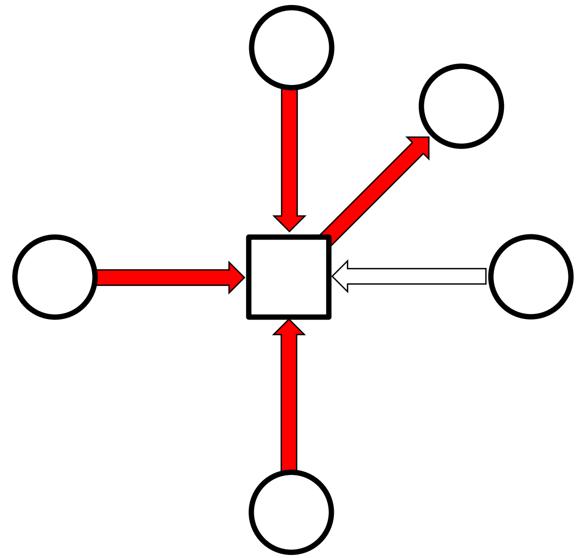




XOR the incoming messages and the check value
- make the number of red colors even

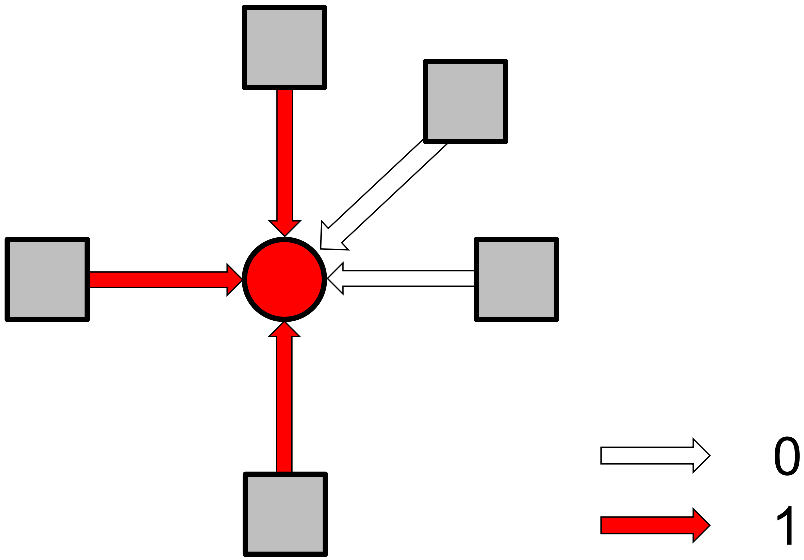
Gallager B
check message update

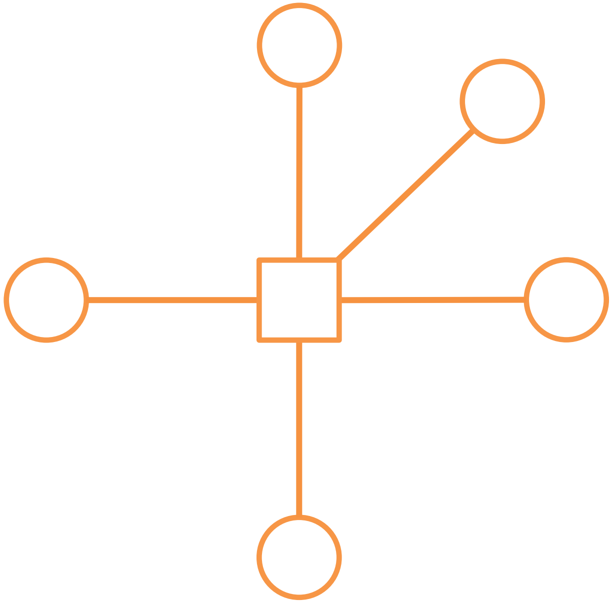




XOR the incoming messages and the check value
- make the number of red colors even

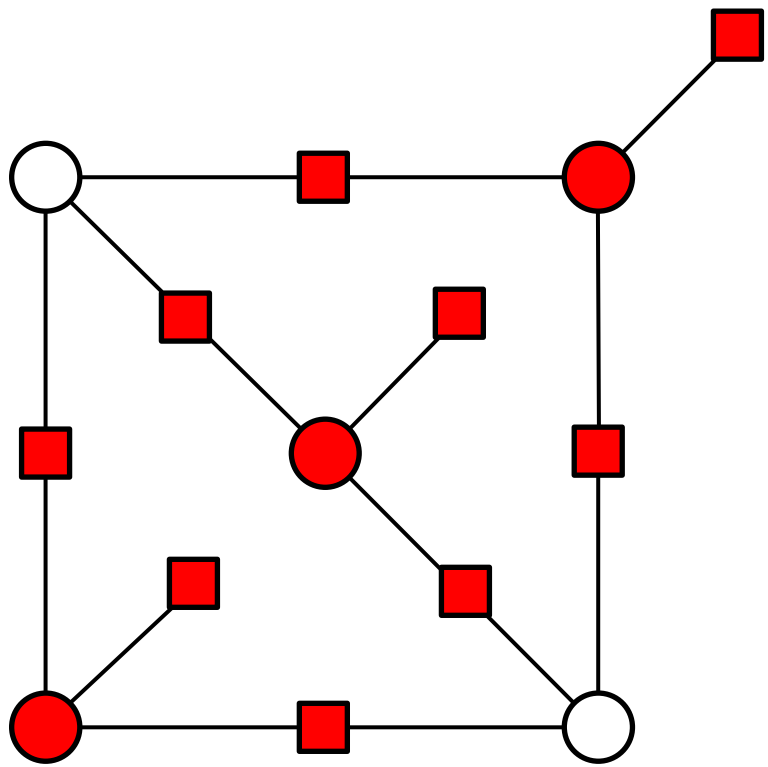
Gallager B decision

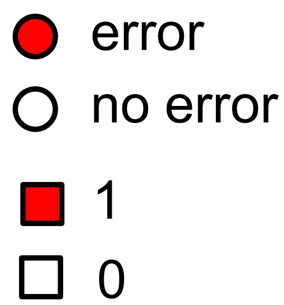




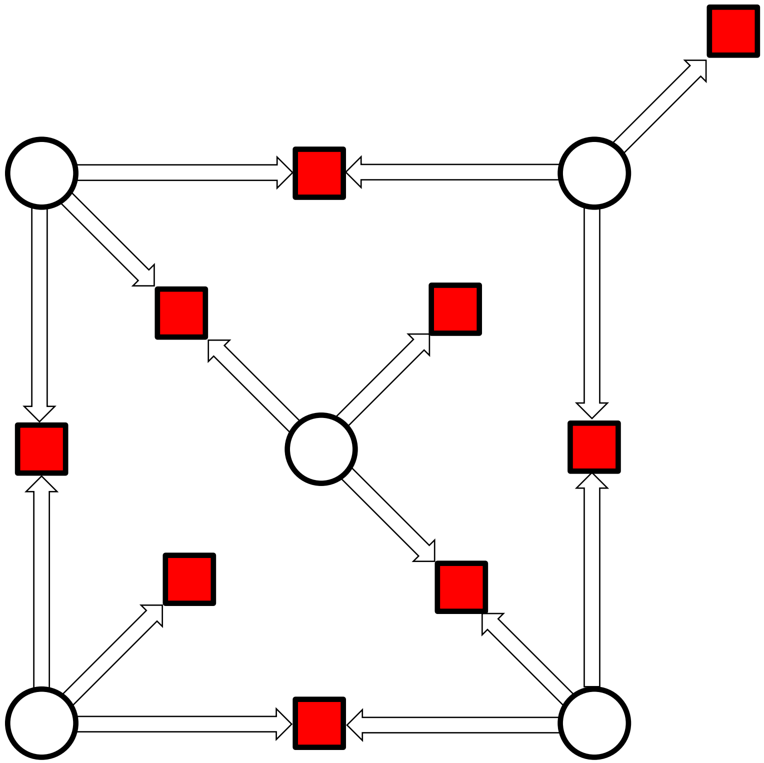
the bit value is decided as
the majority of **all** incoming messages

Trapping set illustration

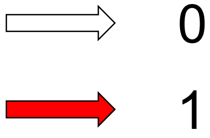




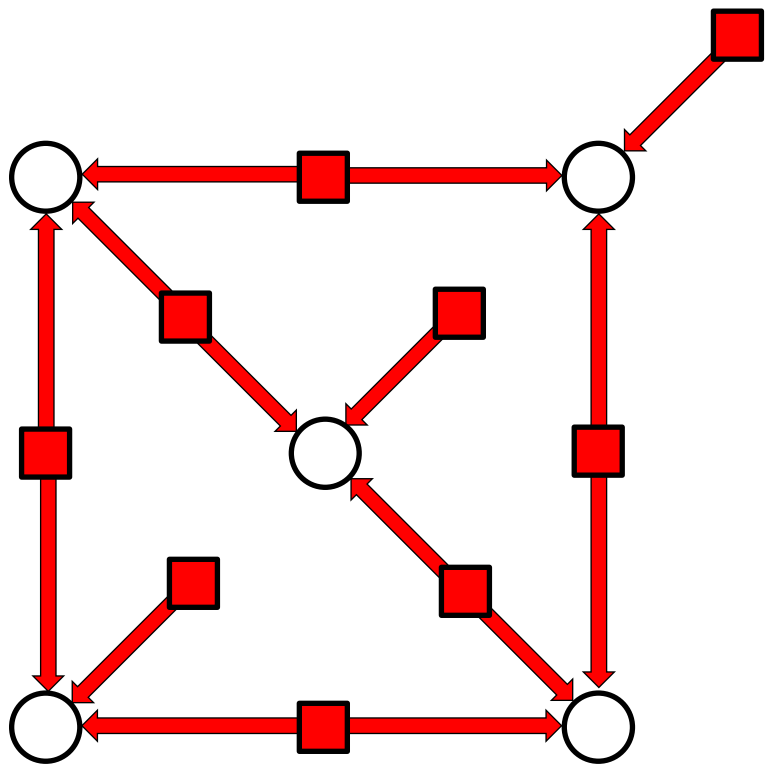
Trapping set illustration



iteration 1 – initialization
all variables send zero

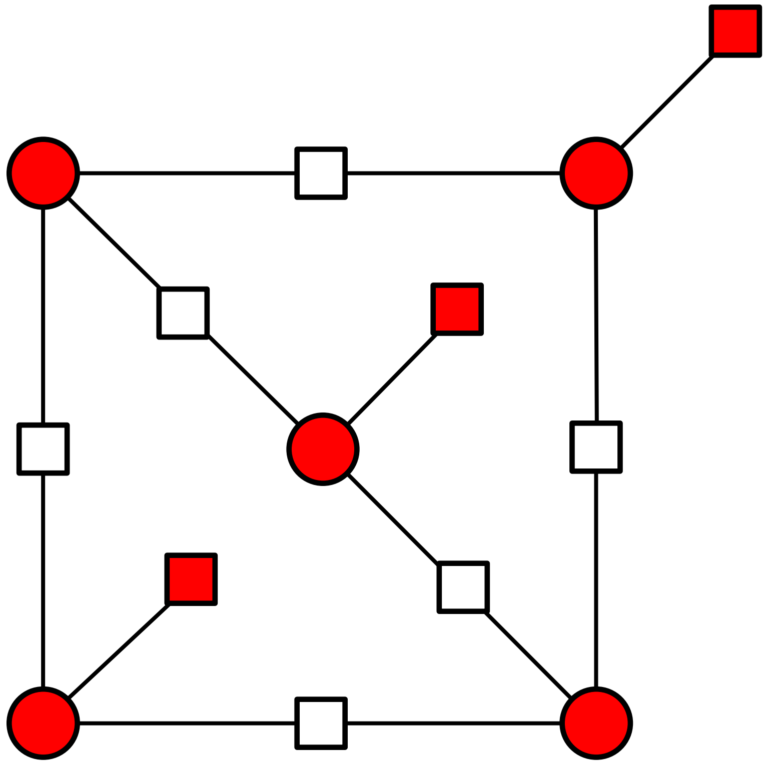


Trapping set illustration



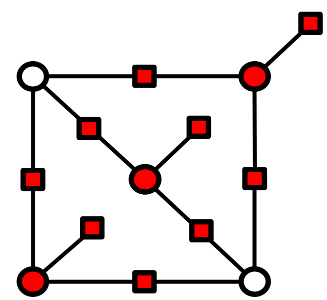
iteration 1 – the second half

Trapping set illustration

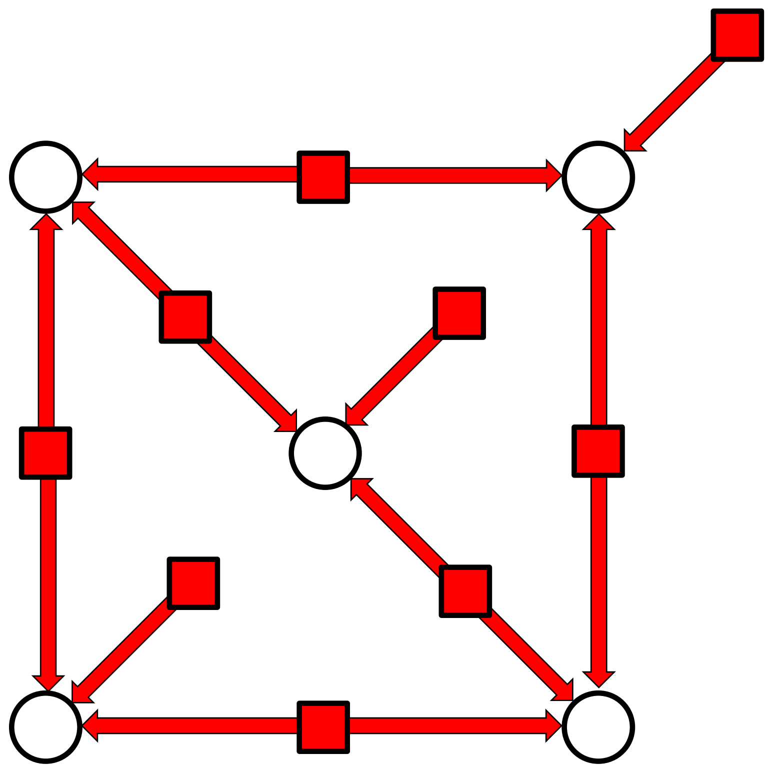


Iteration 1 - decison

syndrome mismatch

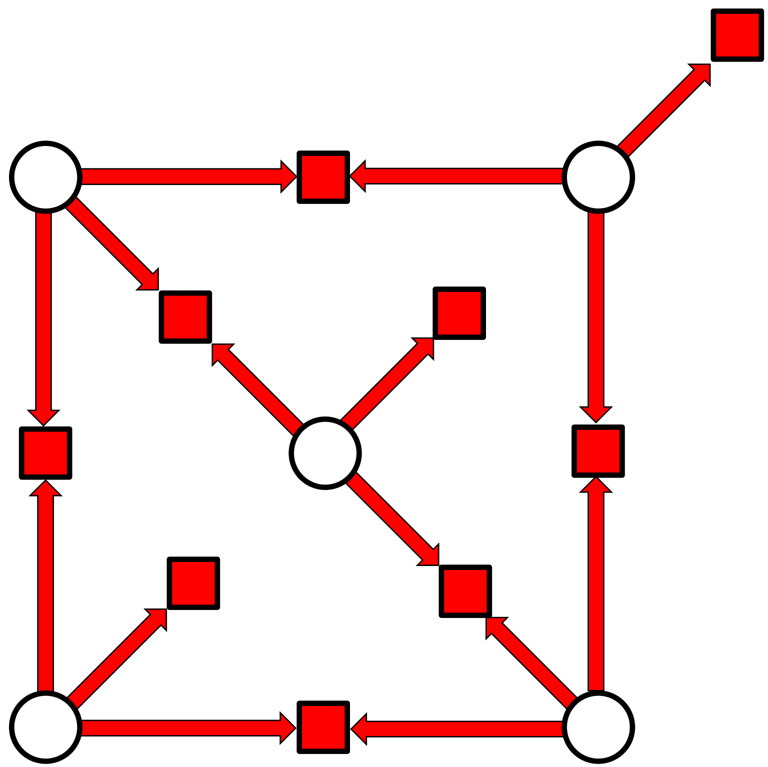


Trapping set illustration

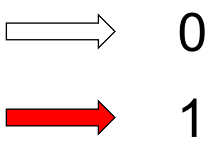


recall what messages were sent to variable nodes

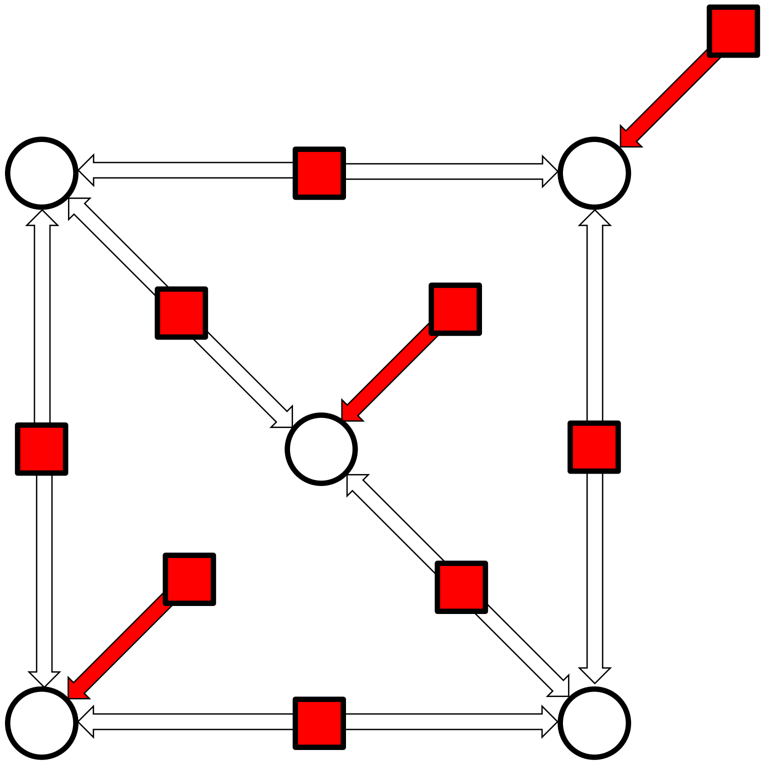
Trapping set illustration



iteration 2 – first half
variables send the majority
of incoming messages

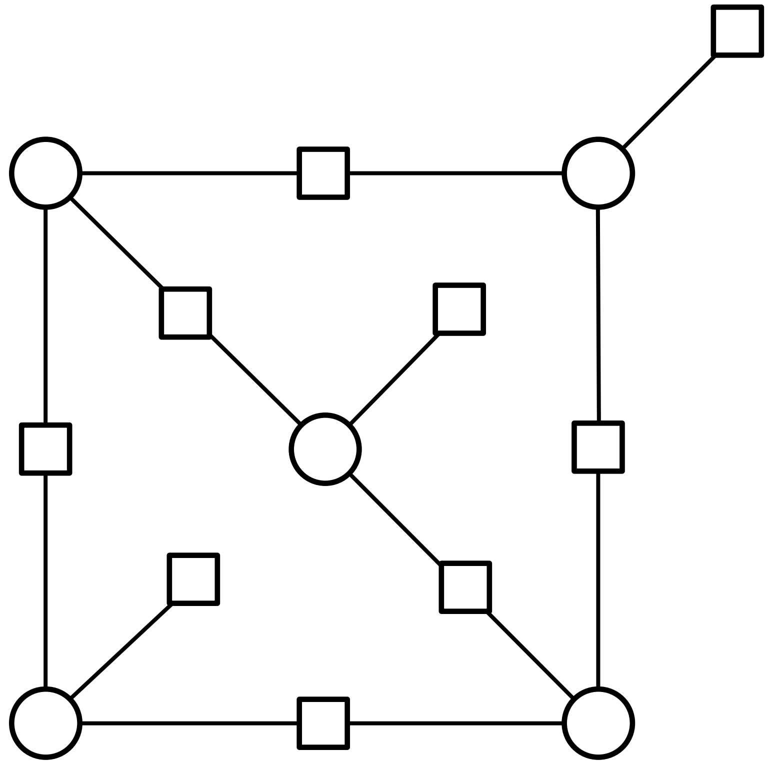


Trapping set illustration

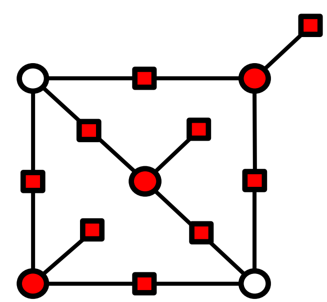


iteration 2 – second half

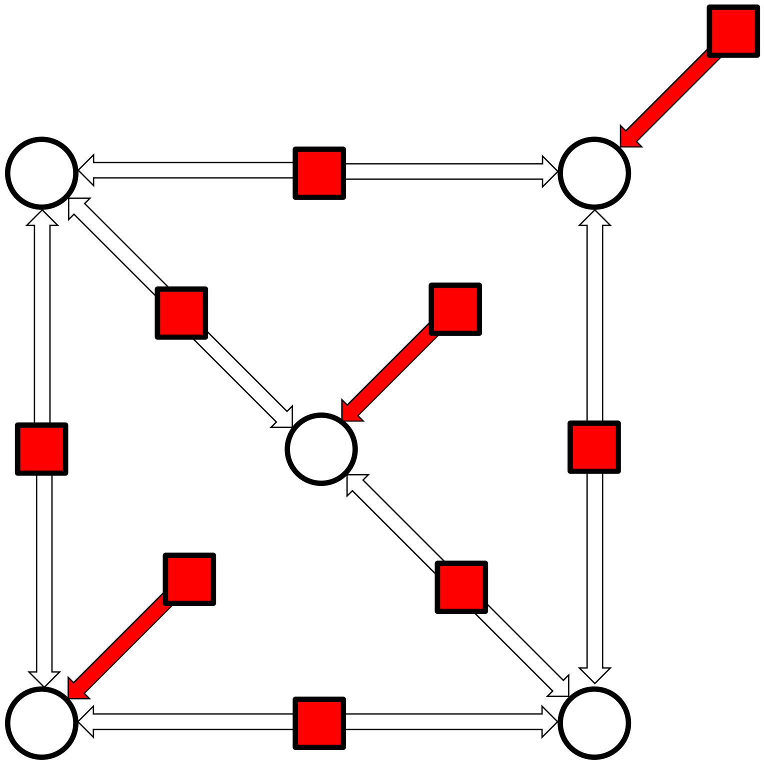
Trapping set illustration



iteration 2 - decision
syndrome mismatch

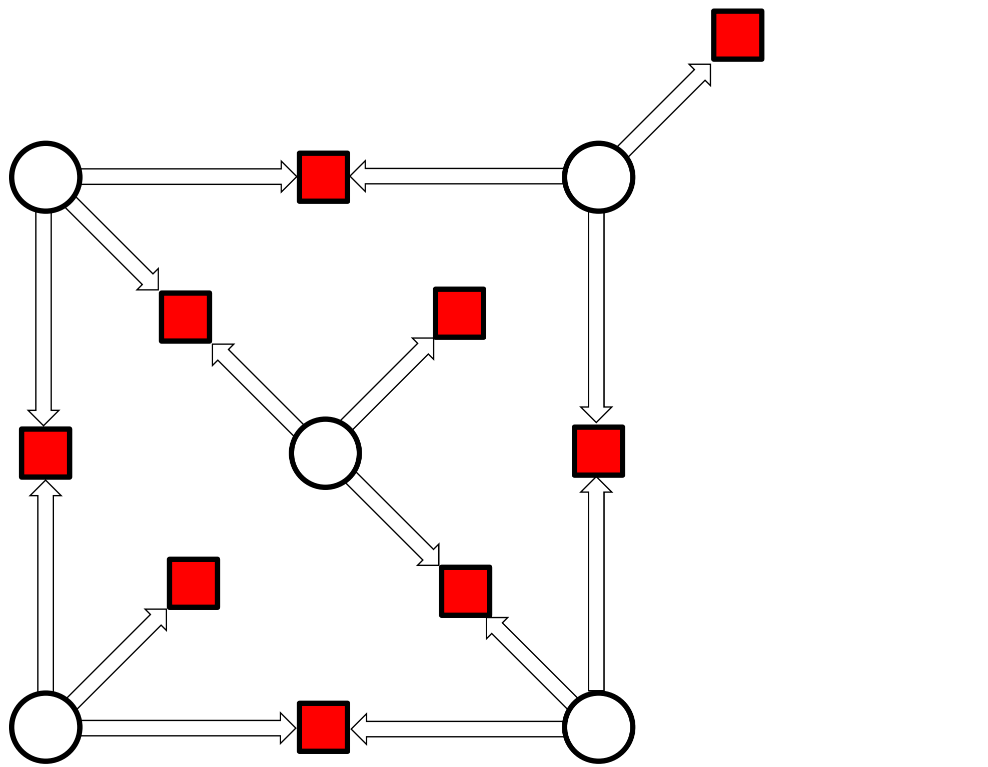


Trapping set illustration



messages sent to variable nodes in previous iteration

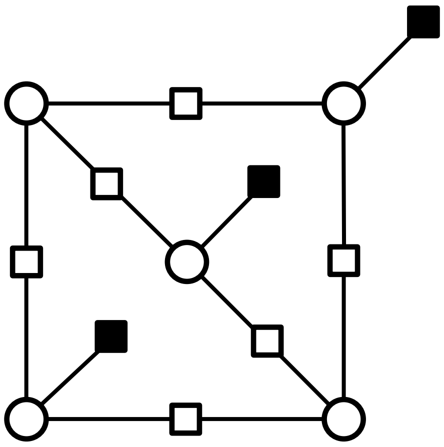
Trapping set illustration



iteration 3 – first half
as when we started

Quantum trapping sets

- Classical definition: An (a, b) trapping set $\mathcal{T}_c$ for an iterative decoder $\mathcal{D}$ is a non-empty set not eventually correct variable nodes in a Tanner graph of size a, inducing a subgraph $\mathcal{G}(\mathcal{T})$ with b odd degree check nodes.
- In a syndrome-based iterative decoders, we keep the (a, b) notation, but the situation is more complex.

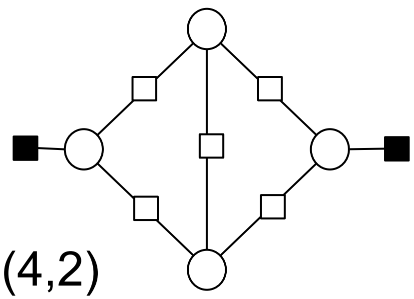


(a, b)= (5,3) trapping set

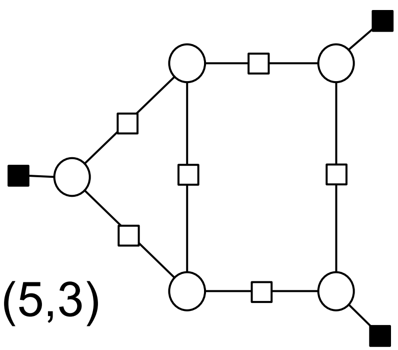
Summary of our findings

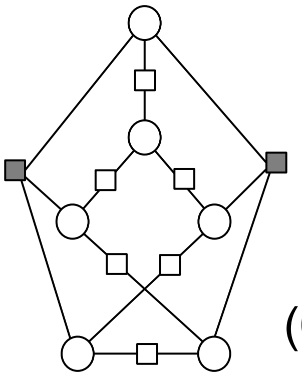
- QLDPC codes have two classes of trapping sets:
- Classical-looking trapping set due to dense subgraphs

(4,2)



(5,3)



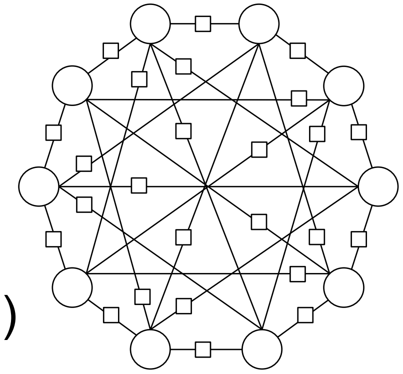


(6,2)

in [[900,36,10]] code

- Inherently-quantum trapping sets due to the symmetry of stabilizers

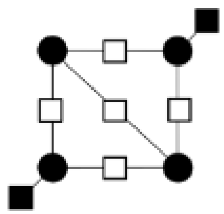
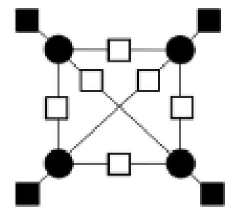
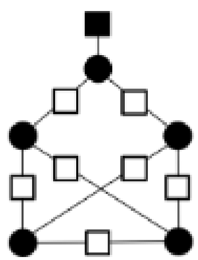
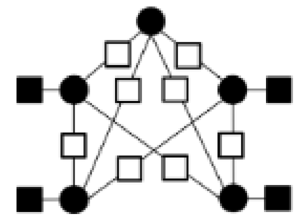
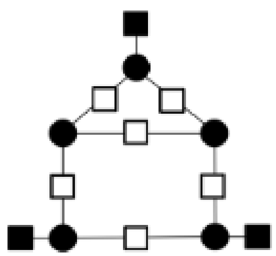
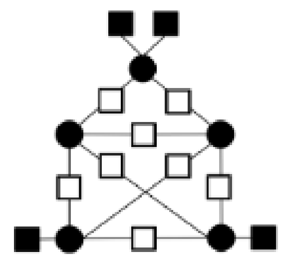
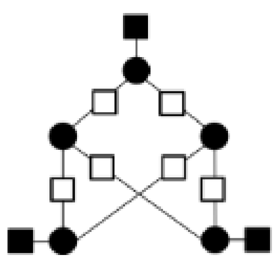
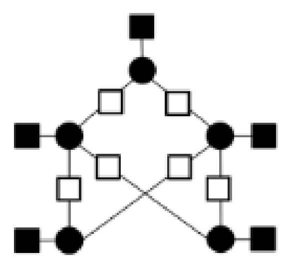
(10,0)



in [[254,28]] code,
“A1” code of Panteleev and Kalatchev

THE UNIVERSITY OF ARIZONA. TUCSON ARIZONA

Trapping sets in [[900,36,10]] HP Code

Quantum TS	Parameters [1] (a,b) CYC(x) Count	Quantum TS	Parameters (a,b) CYC(x) Count
	(4,2) $2x^6 + x^8$ 720		(4,4) $4x^6 + 3x^8$ 72
	(5,1) $2x^6 + 3x^8 + 2x^{10}$ 240		(5,4) $4x^6 + 5x^8 + 4x^{10}$ 36
	(5,3) $x^6 + x^8 + x^{10}$ 4080		(5,4) $5x^6 + 5x^8 + 2x^{10}$ 90
	(5,3) $3x^8$ 360		(5,5) $3x^8$ 5184

Trapping sets due to degeneracy

- Recall: the minimum distance of a code is the minimal weight of codewords in $\mathrm{rowspace}(L)$.
- Code is called degenerate if the minimum distance is much greater than the row weight of *H*.
  - Degenerate errors have weight much smaller than the minimum distance.
- Degenerate errors: Errors $\boldsymbol{e}$ and $\boldsymbol{f}$ that differ by an element in the stabilizer group.

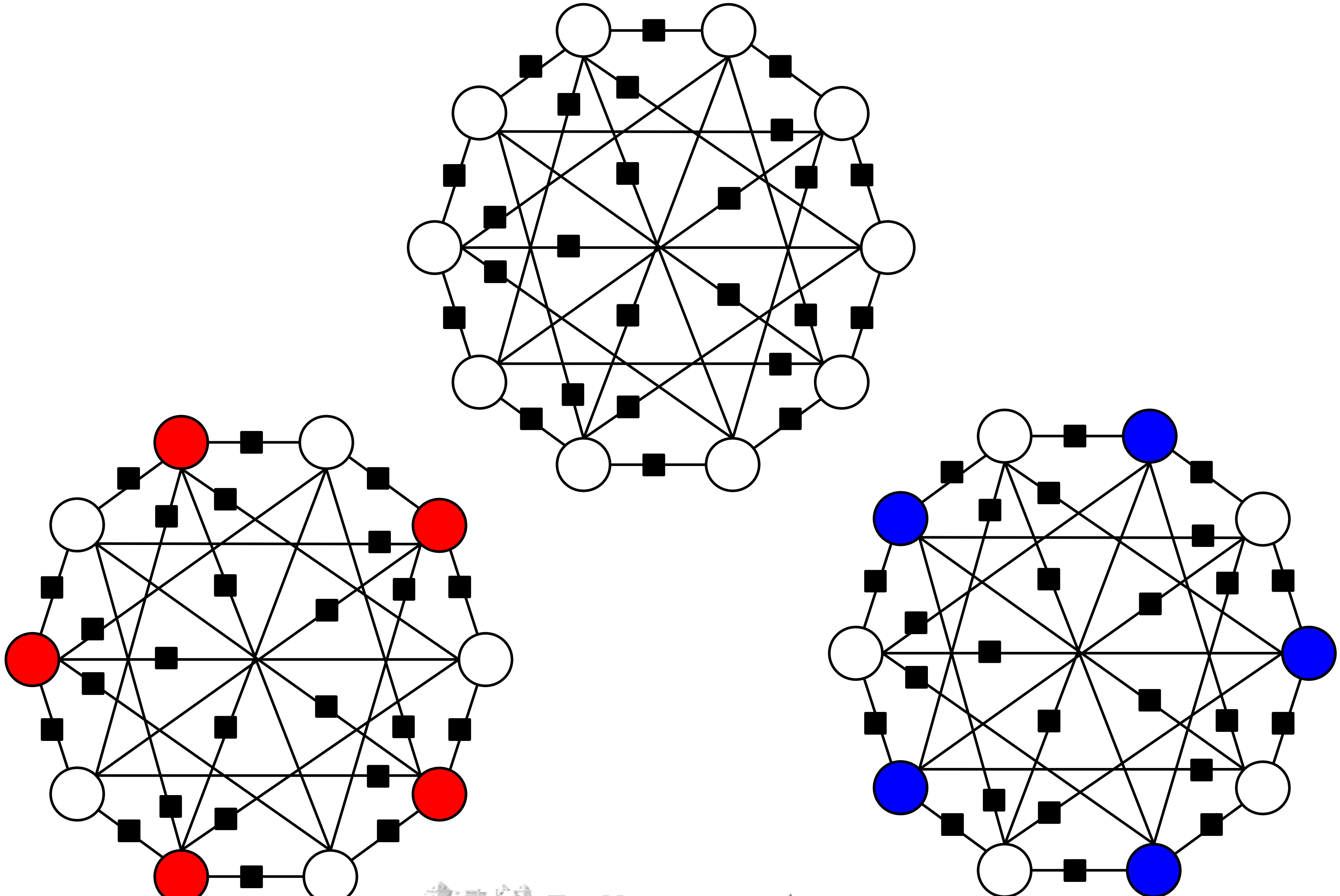
$$\boldsymbol{e} = \boldsymbol{f} + \boldsymbol{h}, \quad \boldsymbol{h} \in \mathrm{rowspace}(H).$$

- Impossible to tell them apart using the syndrome.

Impacts of degeneracy

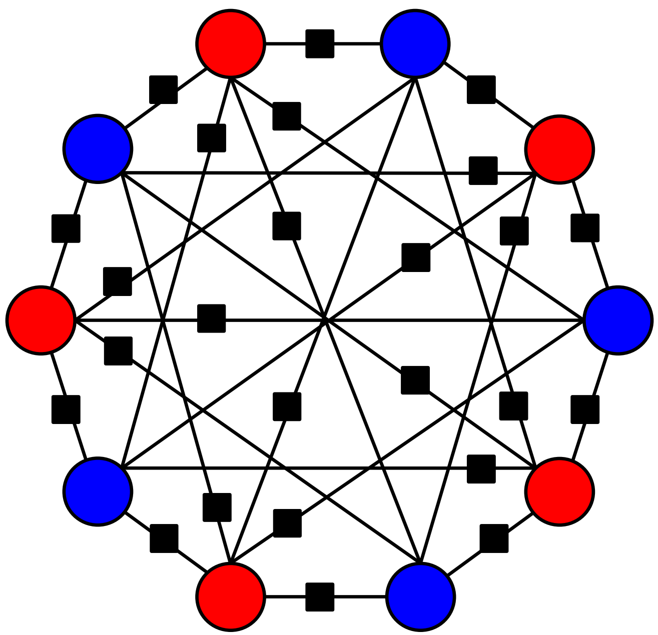
- We typically look for errors of lowest weight.
- H is sparse. If $\boldsymbol{e}$ is a low weight error pattern, then $\boldsymbol{e}+\boldsymbol{h}$, $\boldsymbol{h} \in \mathrm{rowspace}(H)$ may remain low-weight error pattern.
- Typically, there are many low-weight error patterns giving the same syndrome.
- Iterative decoder fails when degeneracy is combined with symmetry!

An example of a degenerate error



An example of a degenerate error

- Error patterns $\boldsymbol{e}$ and $\boldsymbol{f}$ induce a subgraph of a codeword.
- Iterative decoder attempts to converge to both $\boldsymbol{e}$ and $\boldsymbol{f}$ simultaneously leading to decoder failure.

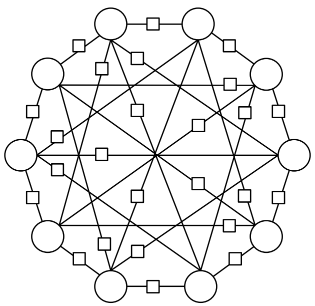


Symmetric stabilizer trapping sets are bad

- Symmetric degenerate errors are harmful for <u>all</u> iterative decoders, even “strong” decoders.
  - The sum and product operations in the sum-product algorithm are symmetric functions, thus messages in a symmetric graph are all equal, and decoder instead of $\boldsymbol{e}$ or $\boldsymbol{f}$ outputs $\boldsymbol{e}+\boldsymbol{f}$.

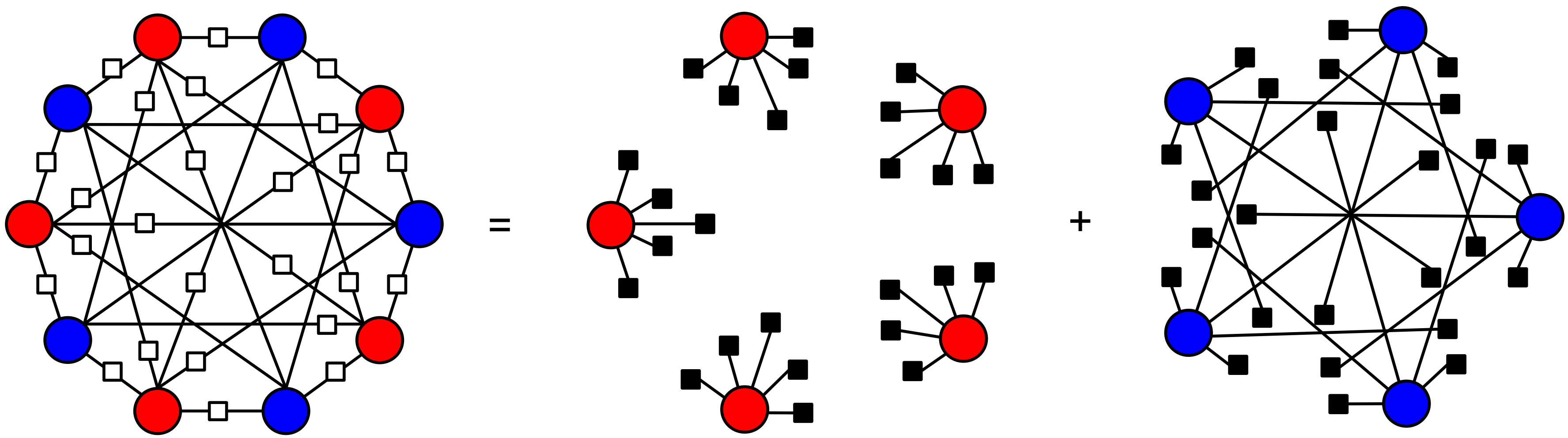
Harmfulness of symmetric stabilizers

- Different from classical-type TSs
- If an iterative decoder can correct half of the error patterns, degeneracy takes care of the rest!
- How degeneracy can be exploited in decoding!



- Lemma: For an $(a, 0)$ symmetric stabilizer TS with any iterative decoder which can correct up to $\frac{a}{2} - 1$ error patterns in the symmetric stabilizer, no error pattern on $\frac{a}{2} + 1$ nodes or more on the symmetric stabilizer is a failure configuration. In Trapping Sets of Quantum LDPC Codes: arxiv:2012.15297

Topology of Symmetric Stabilizer TSs



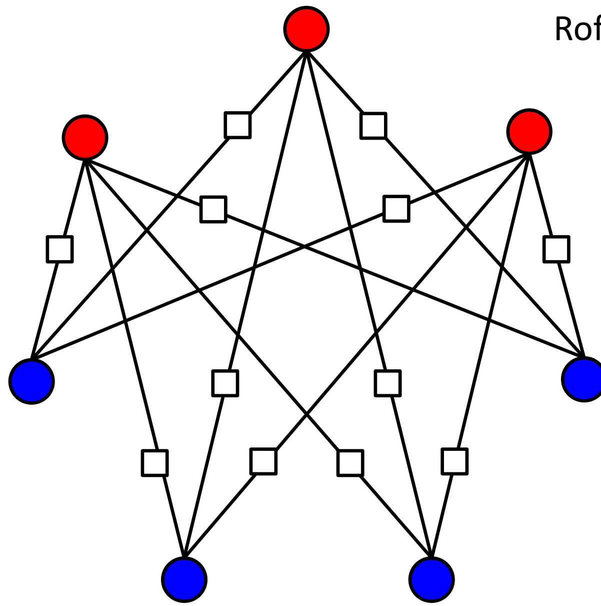
- This specific symmetric stabilizer is present in the A1 code - [[254,28]] code with circulant size $\ell = 127$.
- Non-classical behavior! A decoder fails only for <u>exactly</u> five errors, higher-weight errors are corrected.

P. Panteleev and G. Kalachev, arXiv:1904.02703, 2019.

Many stabilizers are asymmetric

- Asymmetric stabilizer in the [[900,36,10]] HP code

Roffe et al. May 2020. arXiv:2005.07016



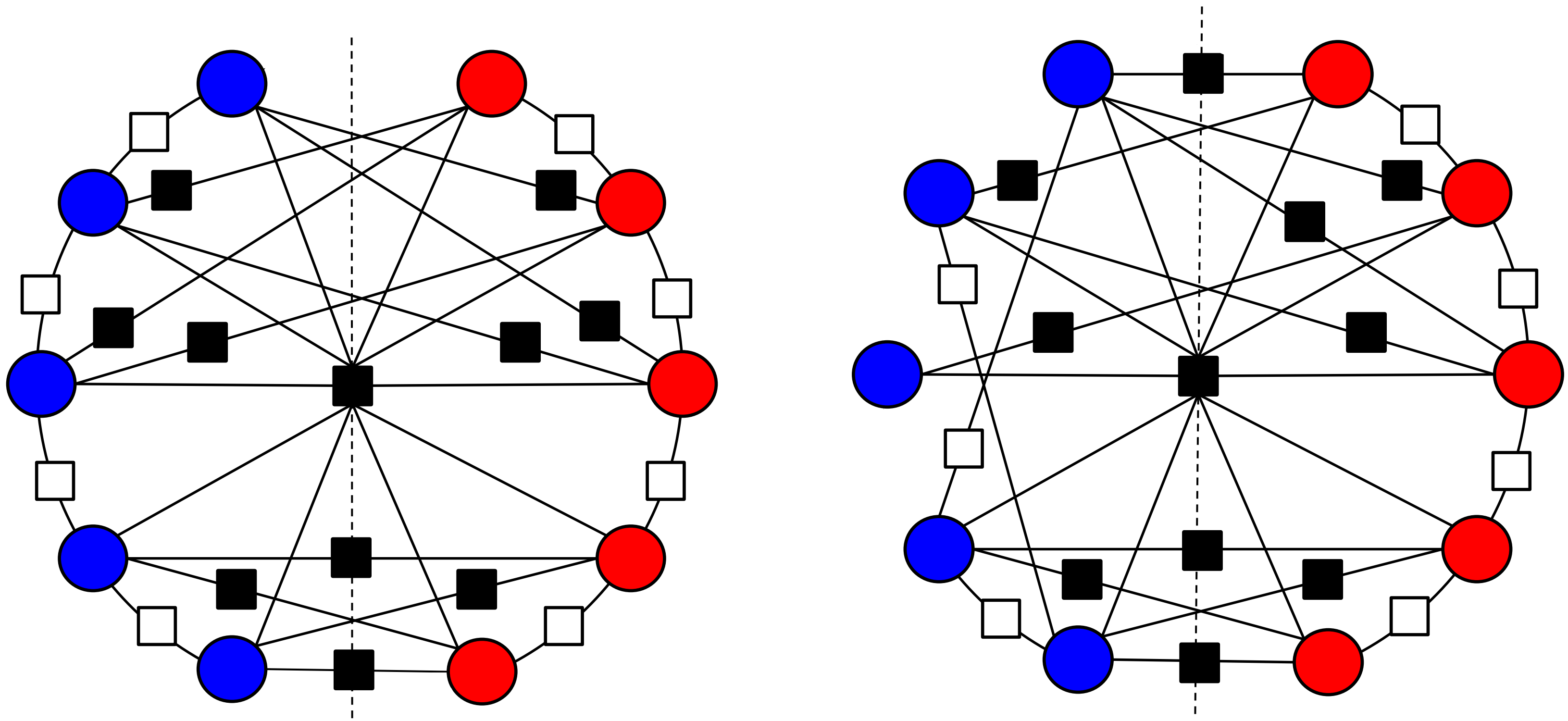
- When a QLDPC code is constructed from classical codes, trapping sets in classical codes remain and multiply in the Tanner graph of the quantum code.
- More structure and elegance, more symmetric stabilizers.

Summary

- Iterative decoders on QLDPC codes fail due to presence of trapping sets - dense subgraphs of specific structure:
  - classical looking (but quite different message dynamic)
  - symmetric stabilizers
- We present the methodology to identify and enumerate trapping sets.
- In code design, increasing d_{min} is not sufficient, the Tanner graph must be also free of small trapping sets.
- BP has a fundamental flaw and fails on dense graphs, but message passing algorithms can be designed using the knowledge of trapping sets.

Implications

- Knowledge of trapping set helps to design better codes and better decoders
- Method I: modify stabilizers to make them asymmetric.



Elimination of small TSs

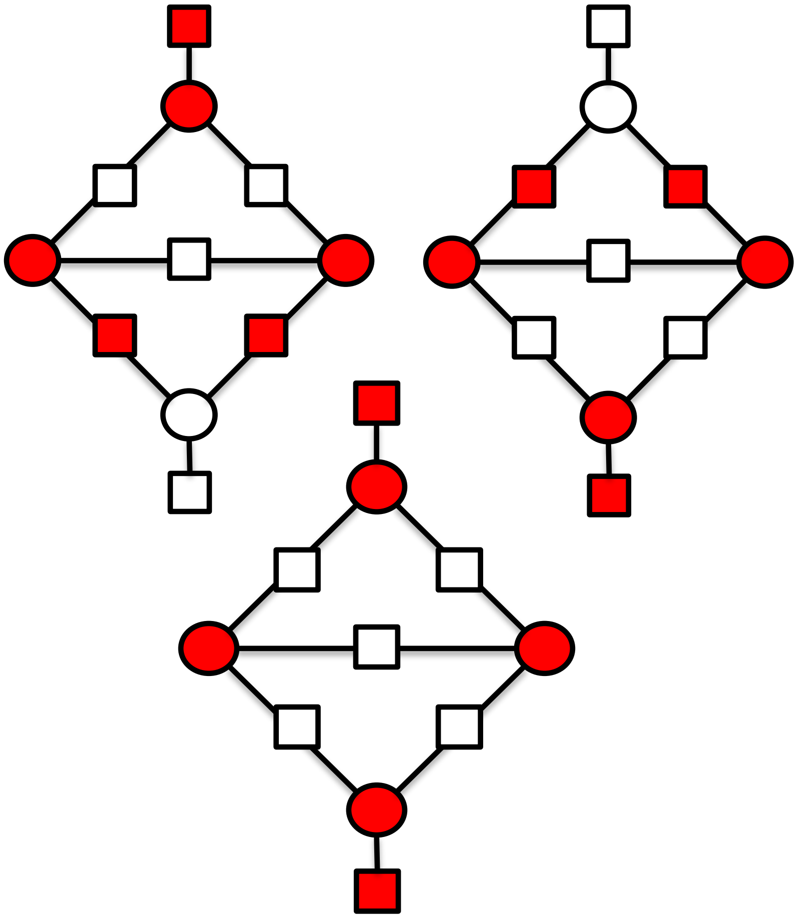
- Method II: Eliminate small trapping sets in code construction.
  - Eliminating TS in constituent codes in HP code construction automatically eliminates them in global code (picture of a HP code here)
- But, which one are more dangerous than the others?
- The answer in the recent Nithin's paper:

N. Raveendran, D. Declercq, and B. Vasić, "A Sub-Graph Expansion-Contraction Method for Error Floor Computation," *IEEE Transactions on Communications*, 2020.

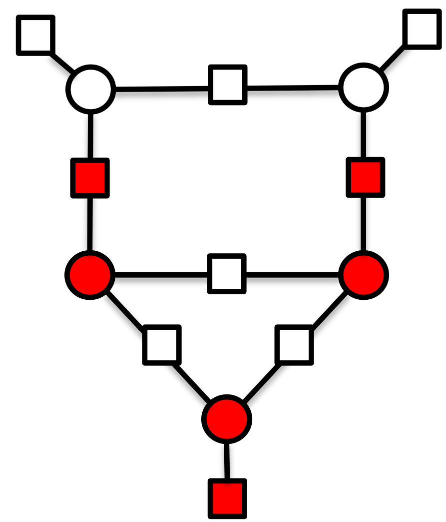
- It is not the size of trapping set that determines harmfulness, it is a critical number and strength of a trapping set.

Harmful syndrome/error patterns: min-sum

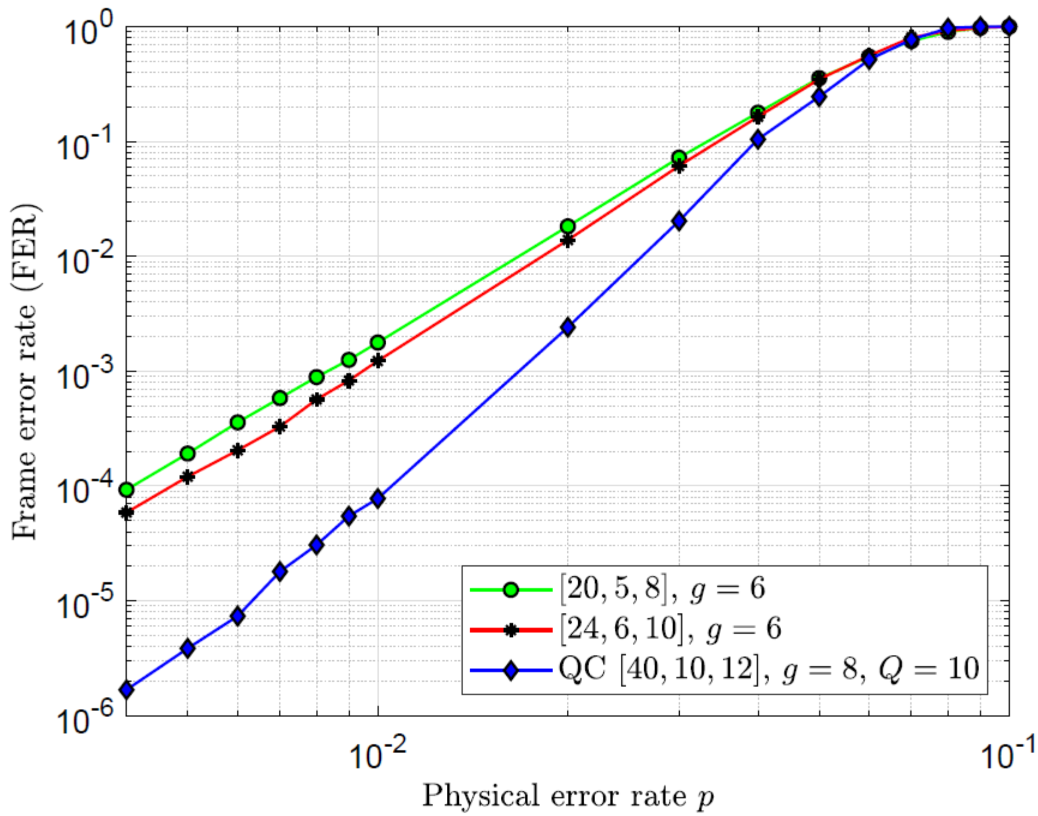
- (4,2) TS



Critical error pattern: (5,3) TS



TS-aware code construction

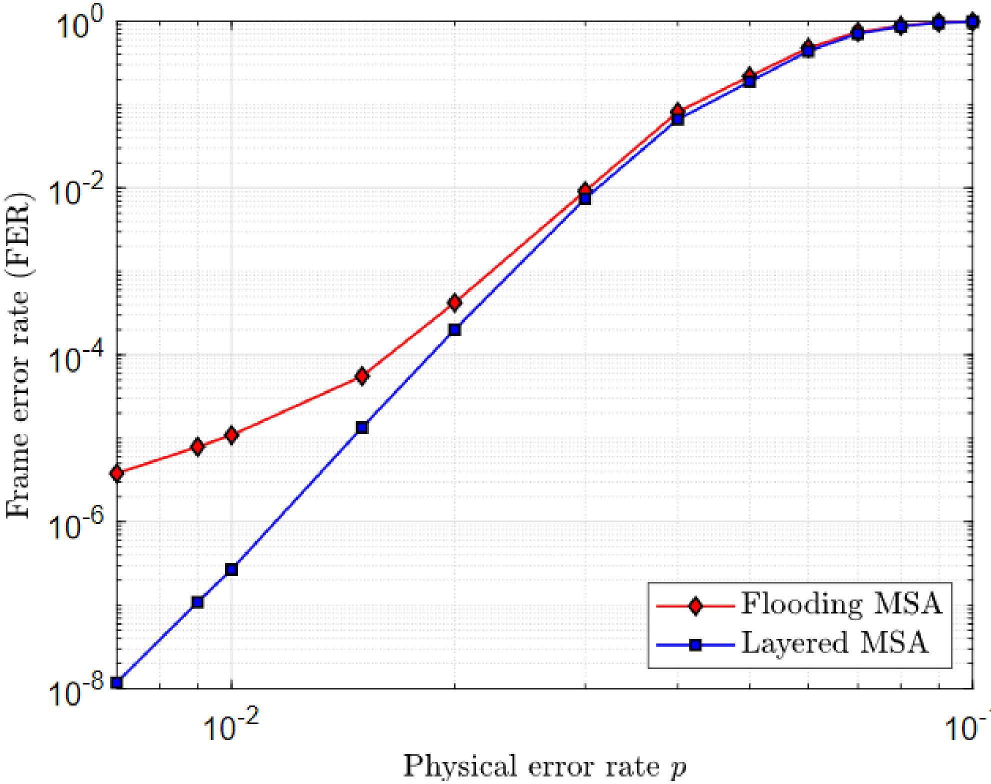


symmetric HP codes constructed using random constituent codes [20,5,8] and [24,6,10] from Roffe et al.
HP code constructed using a trapping set aware QC[40,10,12] code

Roffe et al. May 2020. arXiv:2005.07016 [quant-ph]

Better decoders

- A1 [[254,28]] code decoded by the min-sum algorithm (MSA) for two different schedules:
  - The layered schedule corrects **all** symmetric stabilizer TSs and numerous classical-type TSs.
  - Large unexplored area with potentially big impact.



Thank you!

N. Raveendran and B. Vasic, “Trapping Sets of Quantum LDPC Codes,” arXiv:2012.15297 [cs.IT]

References

1. S. Lin and D. J. Costello. Error control coding, volume 2. Prentice hall, 2001.
2. R. Gallager, Low Density Parity Check Codes. Cambridge, MA: M.I.T. Press, 1963
3. J. Pearl, Probablisitic Reasoning in Intelligent Systems. San Francisco, CA: Kaufmann, 1988
4. F. R. Kschischang, B. J. Frey, and H. A. Loeliger. Factor graphs and the sum-product algorithm, in IEEE Trans. Inform. Theory, 47(2):498–519, Feb. 2001.
5. M. P. C. Fossorier *et al.* "Reduced complexity iterative decoding of low density parity check nodes based on belief propagation" IEEE Trans. on Commun. vol. 47 no. 5 pp. 673-680 May 1999.
6. S. K. Planjery , D. Declercq, L. Danjean, and B. Vasi´c. Finite alphabet iterative decoders, Part I: Decoding beyond belief propagation on the binary symmetric channel, In IEEE Trans. Commun., 61(10):4033–4045, Nov. 2013.
7. T. J. Richardson, “Error floors of LDPC codes,” in Proc. 41st Annual Allerton Conf. on Commun., Control and Computing, 2003, pp. 1426–1435.
8. B. Vasić, D.V. Nguyen, and S. K. Chilappagari, "Chapter 6 - Failures and Error Floors of Iterative Decoders ," Academic Press Library in Mobile and Wireless Communications , 2014.
9. S. K. Chilappagari, S. Sankaranarayanan, and B. Vasić, "Error floors of LDPC codes on the binary symmetric channel," ICC 2006.
10. S. K. Chilappagari, D. V. Nguyen, B. Vasić, and M. W. Marcellin, "Error correction capability of column-weight-three LDPC codes under the Gallager A algorithm - Part II," IEEE Trans. Information Theory, June 2010.
11. S. K. Chilappagari, D. V. Nguyen, B. Vasić, and M. W. Marcellin, "On Trapping Sets and Guaranteed Error Correction Capability of LDPC codes and GLDPC Codes," IEEE Trans. Information Theory, Apr. 2010.
12. B. Vasić, D.V. Nguyen, and S. K. Chilappagari, "Chapter 6 - Failures and Error Floors of Iterative Decoders ," Academic Press Library in Mobile and Wireless Communications , 2014.
13. B. Vasic, S.K. Chilappagari, D.V. Nguyen, and S.K. Planjery. Trapping set ontology. In Proc. 47th Ann. Allerton Conf. on Commun., Control and Computing, Monticello, IL, USA, Sep. 30-Oct. 2 2009.

References

15. D. Gottesman, "Fault-Tolerant Quantum Computation with Constant Overhead," arXiv:1310.2984, 2014.
16. A. Kovalev and L. Pryadko, "Improved quantum hypergraph-product LDPC codes," in 2012 IEEE ISIT, July 2012, pp. 348-352.
17. O. Fawzi, A. Grospellier, and A. Leverrier, "Constant overhead quantum fault-tolerance with quantum expander codes," arXiv:1808.03821, 2018
18. O. Fawzi, A. Grospellier, and A. Leverrier, "Efficient decoding of random errors for quantum expander codes," arXiv:1711.08351, 2017
19. P. Panteleev and G. Kalachev, "Quantum LDPC codes with almost linear minimum distance," arXiv:2012.04068, 2020
20. N. P. Breuckmann and J. N. Eberhardt, "Balanced Product Quantum Codes", arXiv: 2012.09271, 2020
21. J. Roffe *et. al*, "Decoding across the quantum low-density parity-check code landscape" Phys. Rev. Research 2, 043423, 2020

Quantum bit, quantum state

- Qubit: $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ and } |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$a, b \in \mathbb{C}, \quad |a|^2 + |b|^2 = 1$$

- Unlike classical bit having 'state' either 0 or 1, a qubit can be in a linear superposition of states.
- For N–qubit state

$$|\psi\rangle = a_0\,|00\ldots0\rangle + a_1\,|00\ldots1\rangle + \ldots + a_{2^N-1}\,|11\ldots1\rangle$$

$$a_i \in \mathbb{C}, \quad \sum_i |a_i|^2 = 1$$

$$|x_1 x_2 \ldots x_N\rangle \triangleq |x_1\rangle \otimes |x_2\rangle \ldots \otimes |x_N\rangle\,, |x_i\rangle \in \{|0\rangle\,, |1\rangle\}$$

Pauli group and its properties

- Pauli group
  - $\mathcal{G}_1$ with Pauli matrices and multiplicative factors $\pm 1, \pm i,$ closed under matrix multiplication-on single qubit.
  - $\mathcal{G}_n$ is an *n*-fold tensor product of $\mathcal{G}_1$-on *n* qubits.

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad Y = i\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

- Pauli matrices either commute or anti commute - with eigenvalues ± 1 - and have self inverse:
- Eg. $XZ = -ZX$

  $X^2 = I$

$$X^2 = I \quad Y^2 = I \quad Z^2 = I$$

$$XY = iZ \quad YX = iZ$$

$$YZ = iX \quad ZY = iX$$

$$ZX = iY \quad XZ = iY$$

Pauli operators on *n* qubits

$$X|0\rangle = |1\rangle\,, X|1\rangle = |0\rangle\,, X(a|0\rangle + b|1\rangle) = a|1\rangle + b|0\rangle$$
$$Z|0\rangle = |0\rangle\,, Z|1\rangle = -|1\rangle\,, Z(a|0\rangle + b|1\rangle) = a|0\rangle - b|1\rangle$$

$$Y|0\rangle = -i(XZ)|0\rangle = -iX|0\rangle = -i|1\rangle$$
$$Y|1\rangle = -i(XZ)|1\rangle = -iX(-|1\rangle) = i|0\rangle$$

$$A = (X \otimes Z \otimes I \otimes I \otimes Y \otimes Z)$$
$$(X \otimes Z \otimes I \otimes I \otimes Y \otimes Z)|000000\rangle = i|100010\rangle$$

$$(X \otimes Z \otimes I \otimes I \otimes Y \otimes Z)|111111\rangle = -i|010001\rangle$$

Commuting operators (example)

$$AB = (X \otimes Z)(Z \otimes X) = \left(\begin{array}{cc|cc} & & 1 & \\ & & & -1 \\ \hline 1 & & & \\ & -1 & & \end{array}\right)\left(\begin{array}{cc|cc} & 1 & & \\ 1 & & & \\ \hline & & & -1 \\ & & -1 & \end{array}\right)$$

$$AB = \left(\begin{array}{cc|cc} & & & -1 \\ & & 1 & \\ \hline & 1 & & \\ -1 & & & \end{array}\right)$$

$$BA = (Z \otimes X)(X \otimes Z) = \left(\begin{array}{cc|cc} & 1 & & \\ 1 & & & \\ \hline & & & -1 \\ & & -1 & \end{array}\right)\left(\begin{array}{cc|cc} & & 1 & \\ & & & -1 \\ \hline 1 & & & \\ & -1 & & \end{array}\right)$$

$$BA = \left(\begin{array}{cc|cc} & & & -1 \\ & & 1 & \\ \hline & 1 & & \\ -1 & & & \end{array}\right)$$

$$A = (X \otimes Z)$$

$$B = (Z \otimes X)$$

$$\mathrm{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \mathrm{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \mathrm{Y} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad \mathrm{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Commuting Operators

$$
\begin{array}{cccccccc}
 & & 1 & 2 & 3 & 4 & 5 & 6 \\
 & & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
A & = & (X \otimes & Z \otimes & I \otimes & I \otimes & Y \otimes & Z) \\
| & & | & \vdots_1 & | & | & \vdots_2 & | \\
B & = & (X \otimes & X \otimes & X \otimes & Y \otimes & Z \otimes & Z)
\end{array}
$$

$$
\begin{array}{cccccccccccccc}
 & & 1_X & 2_X & 3_X & 4_X & 5_X & 6_X & 1_Z & 2_Z & 3_Z & 4_Z & 5_Z & 6_Z \\
 & & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
\mathbf{a} & = & (1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1) \\
| & & & & & & & & & & & & & \\
\mathbf{b} & = & (1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1)
\end{array}
$$

1 2

$$AB = BA$$

Simplectic product

- Simplectic product

$$\mathbf{a} = (\mathbf{a}_{\mathrm{X}}, \mathbf{a}_{\mathrm{Z}})$$

$$\mathbf{b} = (\mathbf{b}_{\mathrm{X}}, \mathbf{b}_{\mathrm{Z}})$$

$$\mathbf{a}_{\mathrm{X}}\mathbf{b}_{\mathrm{Z}}^{\mathrm{T}} + \mathbf{a}_{\mathrm{Z}}\mathbf{b}_{\mathrm{X}}^{\mathrm{T}}$$

$$(\mathbf{a}_X, \mathbf{a}_Z) \odot (\mathbf{b}_X, \mathbf{b}_Z)^{\mathrm{T}} = \mathbf{a}_X\mathbf{b}_Z^{\mathrm{T}} + \mathbf{a}_Z\mathbf{b}_X^{\mathrm{T}}$$

Entanglement as coding resource

- Superposition versus entangled states

$$|\psi\rangle = \begin{pmatrix} a \\ b \end{pmatrix} \qquad |\phi\rangle = \begin{pmatrix} c \\ d \end{pmatrix} \qquad \underset{\substack{\uparrow \\ \text{superposition}}}{(|\psi\rangle \otimes |\phi\rangle)} = \begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} ac \\ ad \\ bc \\ bd \end{pmatrix}$$

superposition

$$\frac{1}{\sqrt{2}}(|00\rangle + |01\rangle) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

entanglement

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}} \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \neq |\psi\rangle \otimes |\phi\rangle$$

Quantum code - illustration

0000000
0001111
0110011
0111100
1010101
1011010
1100110
1101001

$$\frac{1}{\sqrt{8}}\left(|0000000\rangle + |0001111\rangle + |0110011\rangle + |0111100\rangle + |1010101\rangle + |1011010\rangle + |1100110\rangle + |1101001\rangle\right)$$

codeword

0010110
0011001
0100101
0101010
1000011
1001100
1110000
1111111

$$\frac{1}{\sqrt{8}}\left(|0010110\rangle + |0011001\rangle + |0100101\rangle + |0101010\rangle + |1000011\rangle + |1001100\rangle + |1110000\rangle + |1111111\rangle\right)$$

another codeword

Stabilizer formalism for quantum codes

- Stabilizer Group $\mathcal{S}$: Subgroup of *n*-qubit Pauli group $\mathcal{G}_n$ that leaves a non-trivial code state invariant.
  - Let $|\psi\rangle$ be any codeword state, then $\forall\ S \in \mathcal{S}$, $S\,|\psi\rangle = |\psi\rangle$.
  - All codewords have eigenvalue $+1$ for stabilizers S in $\mathcal{S}$.
  - All elements of $\mathcal{S}$ commute.
  - $-I \notin \mathcal{S}$.
- An [[*n*,*k*]] stabilizer code (in terms of stabilizers)
  - Vector space $V_{\mathcal{S}}$ stabilized by subgroup $\mathcal{S}$ of $\mathcal{G}_n$ such that $-I \notin \mathcal{S}$
  - $V_{\mathcal{S}}$ is the intersection of the subspaces fixed by each operator in $\mathcal{S}$
  - Example: S={I, Z_1Z_2, Z_2Z_3, Z_1Z_3}.

The subspace fixed by Z_1Z_2 is spanned by $|000\rangle$,$|001\rangle$,$|110\rangle$ and $|111\rangle$

The subspace fixed by Z1Z3 is spanned by $|000\rangle$,$|011\rangle$,$|101\rangle$ and $|111\rangle$

The subspace fixed by Z2Z3 is spanned by $|000\rangle$,$|011\rangle$,$|100\rangle$ and $|111\rangle$

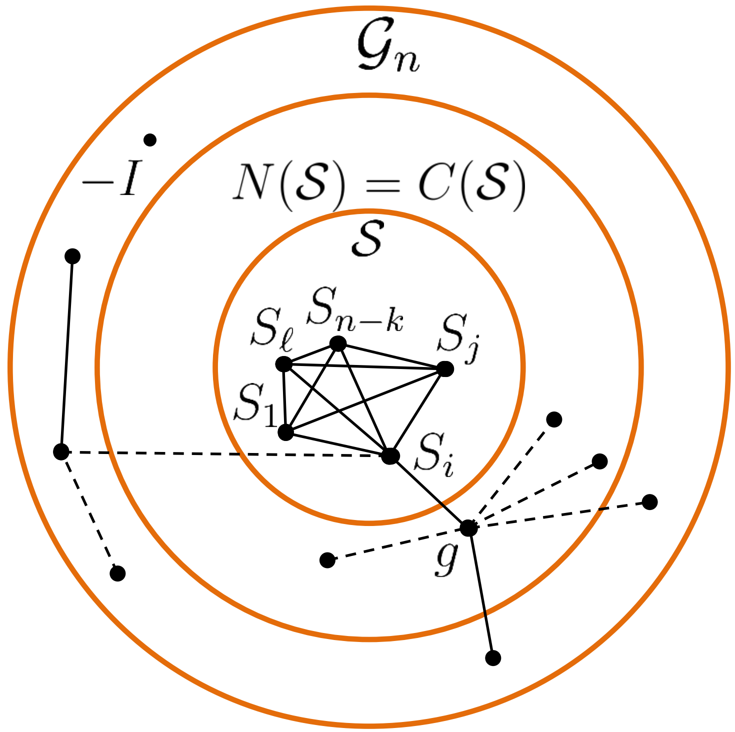
Intersection: $|000\rangle$ and $|111\rangle$

Stabilizer generators

- The code is specified by subspaces stabilized by all operators in the subgroup $\mathcal{S}$.
- $\mathcal{S}$ can be compactly represented by its generators.
- Let $\mathcal{S} = \langle g_1, g_2, \ldots, g_{n-k} \rangle$, where g_i are the generators, then every element of $\mathcal{S}$ can be written as a product of its generators.
- Example. The set of generators of the subgroup

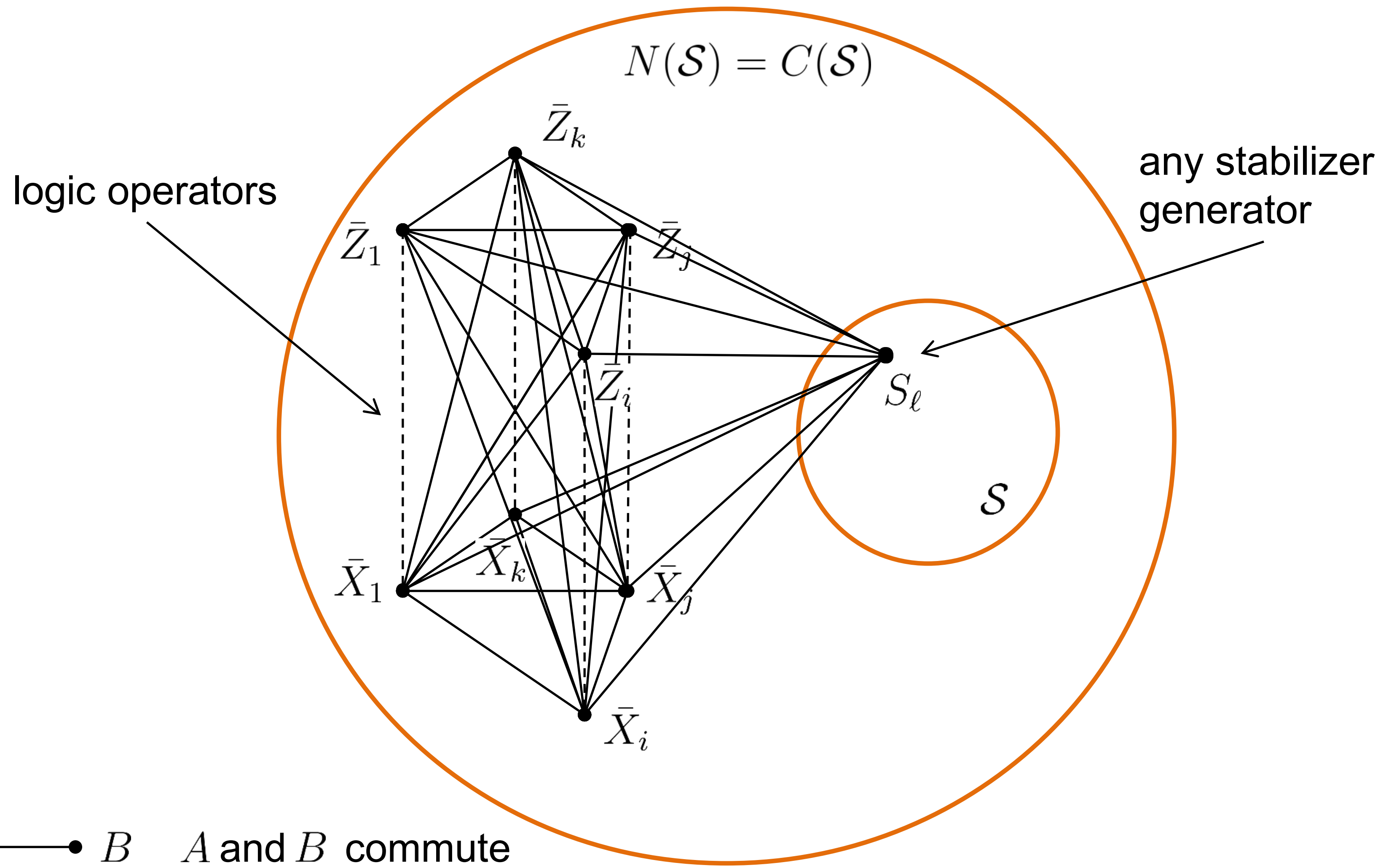
  $S=\{I, Z_1Z_2, Z_2Z_3, Z_1Z_3\}$ is $\langle Z_1Z_2, Z_2Z_3 \rangle$ as $Z_1Z_3=(Z_1Z_2)(Z_2Z_3)$, and $I=(Z_1Z_2)^2$.

Commuting Diagram



A ●——● B	A and B commute
A ●- - -● B	A and B anti-commute

Logical Operators and Normalizer



A •——• B A and B commute

A •- - - -• B A and B anti-commute

Syndrome measurement

- Nature prevents us from learning anything about the probability amplitudes α and β and
- Nature only allows us to measure *observables*.
  - Observable is a Hermitian operator
  - Measurement outcome is one of the eigenvalues of the operator (real number)
  - Quantum state after measurement is eigenvector corresponding to that eigenvalue
- Examples of qubit observables: the Pauli operators X, Y, and Z.
- Measurement – projection to an eigenvector.
- Idea: choose measurements so that encoded state is an eigenvector corresponding to eigenvalue +1.

A Classical Equivalent of Stabilizer Codes

- Recall the definition of a symplectic product $\odot$ of vectors

$$(\mathbf{x},\mathbf{z}) \odot (\mathbf{u},\mathbf{v})^{\mathrm{T}} = \mathbf{x}\mathbf{v}^{\mathrm{T}} + \mathbf{u}\mathbf{z}^{\mathrm{T}}$$

- Since the stabilizer generators commute, any two rows

$$\mathbf{a} = (\mathbf{a}_{\mathrm{X}}, \mathbf{a}_{\mathrm{Z}}) \qquad \mathbf{b} = (\mathbf{b}_{\mathrm{X}}, \mathbf{b}_{\mathrm{Z}})$$

of the parity check matrix $H = \begin{pmatrix} H_{\mathrm{X}} & H_{\mathrm{Z}} \end{pmatrix}$ must satisfy

$$\mathbf{a} \odot \mathbf{b}^{\mathrm{T}} = 0$$

$$(\mathbf{a}_{\mathrm{X}}, \mathbf{a}_{\mathrm{Z}}) \odot (\mathbf{b}_{\mathrm{X}}, \mathbf{b}_{\mathrm{Z}})^{\mathrm{T}} = \mathbf{a}_{\mathrm{X}}\mathbf{b}_{\mathrm{Z}}^{\mathrm{T}} + \mathbf{b}_{\mathrm{X}}\mathbf{a}_{\mathrm{Z}}^{\mathrm{T}} = 0$$

- This leads the condition

$$H_{\mathrm{X}}H_{\mathrm{Z}}^{\mathrm{T}} + H_{\mathrm{Z}}H_{\mathrm{X}}^{\mathrm{T}} = \mathbf{0}$$

CSS Codes

- The parity check matrix is in this form

$$H = \begin{pmatrix} H_{\mathrm{X}} & 0 \\ 0 & H_{\mathrm{Z}} \end{pmatrix}$$

where $H_{\mathrm{X}} H_{\mathrm{Z}}^{\mathrm{T}} = 0.$

- The syndrome has two components

$$\mathbf{e} \odot H^{\mathrm{T}} = (\mathbf{e}_{\mathrm{X}}, \mathbf{e}_{\mathrm{Z}}) \begin{pmatrix} H_{\mathrm{X}}^{\mathrm{T}} & 0^{\mathrm{T}} \\ 0^{\mathrm{T}} & H_{\mathrm{Z}}^{\mathrm{T}} \end{pmatrix} = (\mathbf{s}_{\mathrm{X}}, \mathbf{s}_{\mathrm{Z}})$$

where

$$\begin{aligned} \mathbf{s}_{\mathrm{X}} &= \mathbf{e}_{\mathrm{X}} 0^{\mathrm{T}} + \mathbf{e}_{\mathrm{Z}} H_{\mathrm{X}}^{\mathrm{T}} \\ \mathbf{s}_{\mathrm{Z}} &= \mathbf{e}_{\mathrm{X}} H_{\mathrm{Z}}^{\mathrm{T}} + \mathbf{e}_{\mathrm{Z}} 0^{\mathrm{T}} \end{aligned}$$

thus

$$\begin{aligned} \mathbf{s}_{\mathrm{X}} &= \mathbf{e}_{\mathrm{Z}} H_{\mathrm{X}}^{\mathrm{T}} \\ \mathbf{s}_{\mathrm{Z}} &= \mathbf{e}_{\mathrm{X}} H_{\mathrm{Z}}^{\mathrm{T}} \end{aligned}$$

Quantum channel

- Evolution of a *closed* quantum system is described by a *unitary* transformation E $(EE^\dagger = I)$

$$|\psi\rangle \xrightarrow{E} E\,|\psi\rangle$$

- Discretization of errors:
  - We do not have to separately correct a continuum of possible errors, but only a discrete, finite set of errors.
  - If a code can correct operators in the set $\{E_k\}$, then it can correct any linear sum of these operators.

Error correction

- Must be done without learning the state

$$|\psi\rangle \xrightarrow{E} E\,|\psi\rangle \xrightarrow{\hat{E}^\dagger} \hat{E}^\dagger E\,|\psi\rangle$$

correction

$$\hat{E}$$

error estimation

Syndrome measurement

- Nature prevents us from learning anything about the probability amplitudes α and β and
- Nature only allows us to measure *observables*.
  - Observable is a Hermitian operator
  - Measurement outcome is one of the eigenvalues of the operator (real number)
  - Quantum state after measurement is eigenvector corresponding to that eigenvalue
- Examples of qubit observables: the Pauli operators X, Y, and Z.
- Measurement – projection to an eigenvector.
- Idea: choose measurements so that encoded state is an eigenvector corresponding to eigenvalue +1.

Quantum syndrome decoding

- Let $\mathbf{e}$ be a non-zero error vector, resulting in a syndrome $\mathbf{s}$

$$\mathbf{s} = \mathbf{e} \odot H^{\mathrm{T}} \neq \mathbf{0}$$

$\odot$ symplectic product

- As opposed to a *classical syndrome decoder* that tries to find $\mathbf{e}$ for a given observed syndrome, a valid output of a *quantum* decoder is any one of the vectors

$$\widetilde{\mathbf{e}} = \mathbf{e} + \mathbf{h}, \mathbf{h} \in \mathrm{rowspace(H)}$$

- When $\mathbf{e} + \widetilde{\mathbf{e}} \neq \mathbf{0}$, but

$$(\mathbf{e} + \widetilde{\mathbf{e}}) \odot H^{\mathrm{T}} = \mathbf{0}$$

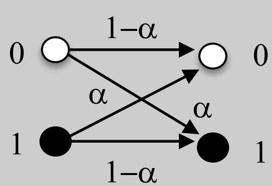
then the correction vector $\mathbf{e} + \widetilde{\mathbf{e}}$ is applied to flip bits in the (unobservable) quantum codeword is also a codeword, and a logical, undetectable, error occurs.

Stabilizer generators

- An [[n,k]] stabilizer code (in terms of stabilizers)
  - Vector space V_S stabilized by subgroup $\mathcal{S}$ of $\mathcal{G}_n$ such that $-I \notin \mathcal{S}$.
  - $\mathcal{S}$ has n-k independent and commuting generators $S_1, \ldots, S_{n-k}$.
  - A codeword is a simultaneous eigenstate of all generators of $\mathcal{S}$ with eigenvalue +1.
- Let $\mathcal{S} = \langle g_1, g_2, \ldots, g_{n-k} \rangle$ be generated by n-k independent generators, and $-I \notin \mathcal{S}$.

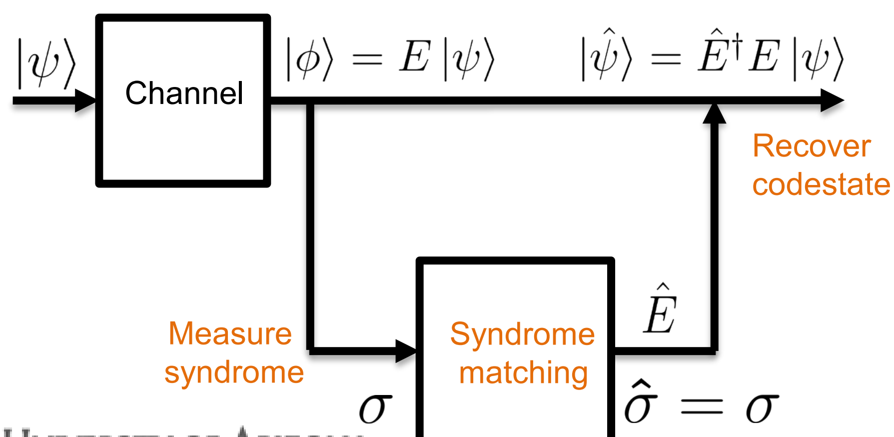
  Then there exists $g \in \mathcal{G}_n$ such that

  $$g g_i g^\dagger = -g_i \text{ and } g g_j g^\dagger = g_j \text{ for all } j \neq i .$$
- Since the generators in $\mathcal{S}$ must commute, addition of each generator cuts the dimension of V_S by half.
- Therefore, V_S is 2^k -dimensional.

CLASSICAL ERROR CORRECTION	QUANTUM ERROR CORRECTION	
Linear code $\mathcal{C} = [N, K, d]$	Stabilizer code $\mathcal{Q} = [[N, K, d]]$	
Information bits (K) - $\mathbf{m}$	Logical qubits (K) - $\vert\psi\rangle_L \in \mathbb{C}^{2^K}$	
Coded bits (N) - $\mathbf{x}$	Physical qubits (N) - $\vert\boldsymbol{\psi}\rangle \in \mathbb{C}^{2^N}$	
Parity check matrix H $\mathbf{x}H^{\mathrm{T}} = \mathbf{0}$	Generators of commutative stabilizer group $\mathcal{S}$ Fixes physical qubits/code state $S\vert\boldsymbol{\psi}\rangle = \vert\boldsymbol{\psi}\rangle, \quad \forall S \in \mathcal{S}$	
$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$ for [7,4,3] code	$H_{\mathrm{P}} = \begin{bmatrix} \mathrm{X} & \mathrm{I} & \mathrm{I} & \mathrm{X} & \mathrm{I} & \mathrm{X} & \mathrm{X} \\ \mathrm{I} & \mathrm{X} & \mathrm{I} & \mathrm{X} & \mathrm{X} & \mathrm{X} & \mathrm{I} \\ \mathrm{I} & \mathrm{I} & \mathrm{X} & \mathrm{I} & \mathrm{X} & \mathrm{X} & \mathrm{X} \\ \mathrm{Z} & \mathrm{I} & \mathrm{I} & \mathrm{Z} & \mathrm{I} & \mathrm{Z} & \mathrm{Z} \\ \mathrm{I} & \mathrm{Z} & \mathrm{I} & \mathrm{Z} & \mathrm{Z} & \mathrm{Z} & \mathrm{I} \\ \mathrm{I} & \mathrm{I} & \mathrm{Z} & \mathrm{I} & \mathrm{Z} & \mathrm{Z} & \mathrm{Z} \end{bmatrix}$ for [[7,1,3]]	
Bit flip error modelled as a BSC(α) 	Bit flip error – X, Phase flip error – Z Bit and phase flip – Y No error - I	Continuous nature of qubit errors discretized to correcting Pauli X, Y, Z errors.

THE UNIVERSITY OF ARIZONA.
TUCSON ARIZONA

CLASSICAL ERROR CORRECTION	QUANTUM ERROR CORRECTION
Direct access to channel output $\mathbf{r}$	Cannot directly access erroneous code state but can measure syndrome.
Syndrome for error detection For $\mathbf{r} \notin \mathcal{C}$ $\boldsymbol{\sigma} = \{\sigma_1, \dots, \sigma_M\} = \mathbf{r}H^T \neq 0$	Measure syndrome Measure the $M = N - K$ stabilizer generators Eigen value $(-1^{\sigma_i}) \rightarrow$ binary syndrome σ_i $\boldsymbol{\sigma} = \{\sigma_1, \dots, \sigma_M\}$
Error detected when $\boldsymbol{\sigma} \neq \mathbf{0}$	Error detected when $\boldsymbol{\sigma} \neq \mathbf{0}$
Decoding Decoder uses H to recover the correct codeword $\mathbf{x}$ from channel output $\mathbf{r}$.	**Syndrome-based Decoding** 1. Syndrome matching - Find corresponding error that matches the syndrome. 2. Error recovery: Reverse the error to get back the codestate $\vert\psi\rangle$



CLASSICAL ERROR CORRECTION	QUANTUM ERROR CORRECTION
No Classical Analog	**Degenerate error patterns** Errors $\mathbf{E}$ and $\mathbf{F}$ have same non-trivial syndrome $\boldsymbol{\sigma}$ if they differ by a stabilizer (check) $\mathbf{F} = \mathcal{S}\mathbf{E}$, where $\mathcal{S} \in \mathcal{S}$
	Degeneracy property of QLDPC codes allows syndrome based decoders to match the syndrome with $\boldsymbol{E}$ or $\boldsymbol{F}$ as degenerate errors Any error upto a stabilizer $\mathbf{F} = \mathcal{S}\mathbf{E}$, where $\mathcal{S} \in \mathcal{S}$

- Our quantum trapping set study investigates effect of
  - Degeneracy of QLDPC codes
  - Syndrome-based iterative decoding
  - QLDPC code constraints from commutativity of stabilizers

Approach

- We want to find all low weight uncorrectable error patterns of a given iterative decoder $\mathcal{D}$.
- If a decoding algorithm is local, such as a bit-flipping or message passing decoding, then a computationally efficient algorithm for finding all low-weight error patterns exists. Induced subgraphs are trapping sets.

N. Raveendran, D. Declercq, and B. Vasić, "A Sub-Graph Expansion-Contraction Method for Error Floor Computation," *IEEE Transactions on Communications*, 2020.

- Consequences:
  - on a depolarizing channel with probability α, we can accurately compute decoding probability of error for low values of channel error rates.
  - The knowledge of trapping sets allows us to design better codes and better decoders!

Trapping sets due to degeneracy

- Recall: the minimum distance of a code is the minimal weight of operators that commute with all the stabilizers but are not in the stabilizer group. Also, the minimal weight of logical operators.
- Code is called degenerate if the minimum distance is much greater than the weight of the stabilizers.
  - Degenerate errors have weight much smaller than the minimum distance.
- Degenerate errors: Errors $\boldsymbol{e}$ and $\boldsymbol{f}$ that differ by an element in the stabilizer group.

$$\boldsymbol{e} = \boldsymbol{f} + \boldsymbol{h}, \quad \boldsymbol{h} \in \mathrm{rowspace}(H).$$

- Impossible to tell them apart using the syndrome.

Generalized bicycle codes

- $H_X = [A\ B]$ and $H_Z = [B^T A^T]$
  - Commuting binary matrices A and B, i.e, $AB = BA$
  - Introduced by Kovalev and Pryadko, called Kronecker Sum-Product Codes,
  - Generalization of MacKay et. al. Bicycle Codes, where $A = B$
- Stabilizer commutativity satisfied: $H_X H_Z^T = AB + BA = \mathbf{0}$
- Panteleev and Kalachev use circulant matrices.
  - A1 code - [[254,28]] code with circulant size $\ell = 127$
  - $A := a(x) = 1 + x^{15} + x^{20} + x^{28} + x^{66}$,
  - $B := b(x) = 1 + x^{58} + x^{59} + x^{100} + x^{121}$.

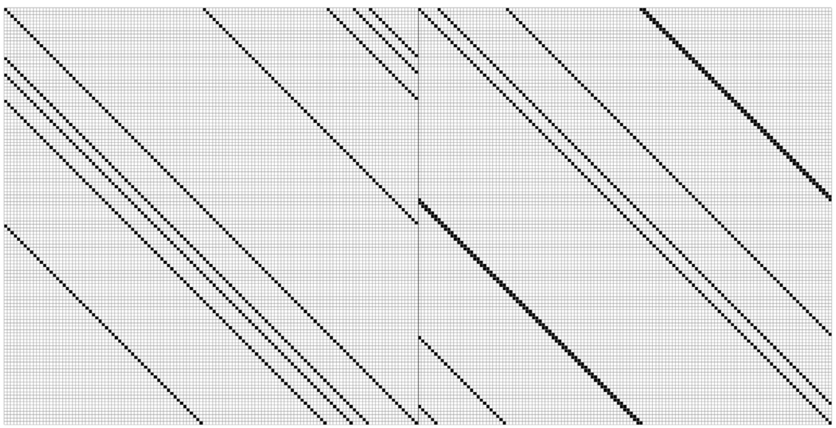
A. A. Kovalev and L. P. Pryadko, Phys. Rev. A 88, 012311, 2013.

P. Panteleev and G. Kalachev, arXiv:1904.02703, 2019.

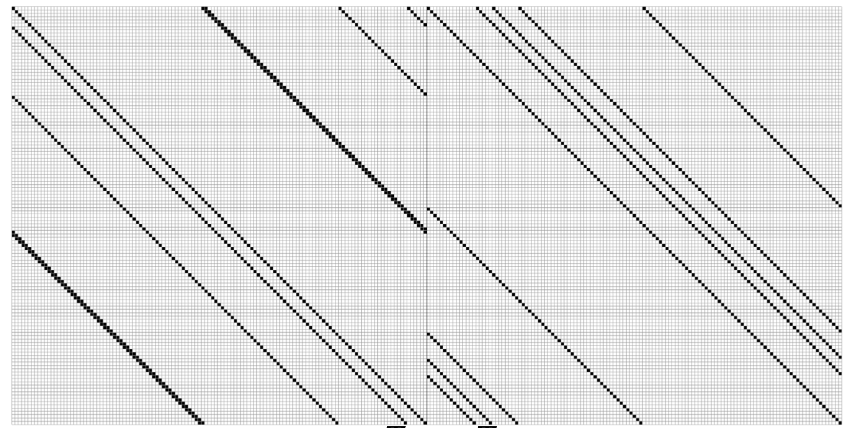
D. J. C. MacKay, et al. , IEEE Trans. Inf. Theory, 50, 10, 2315–2330, 2004.

[[254,28]] generalized bicycle code

- A1 code - [[254,28]], $\ell = 127$
  - $A := a(x) = 1 + x^{15} + x^{20} + x^{28} + x^{66}$,
  - $B := b(x) = 1 + x^{58} + x^{59} + x^{100} + x^{121}$.



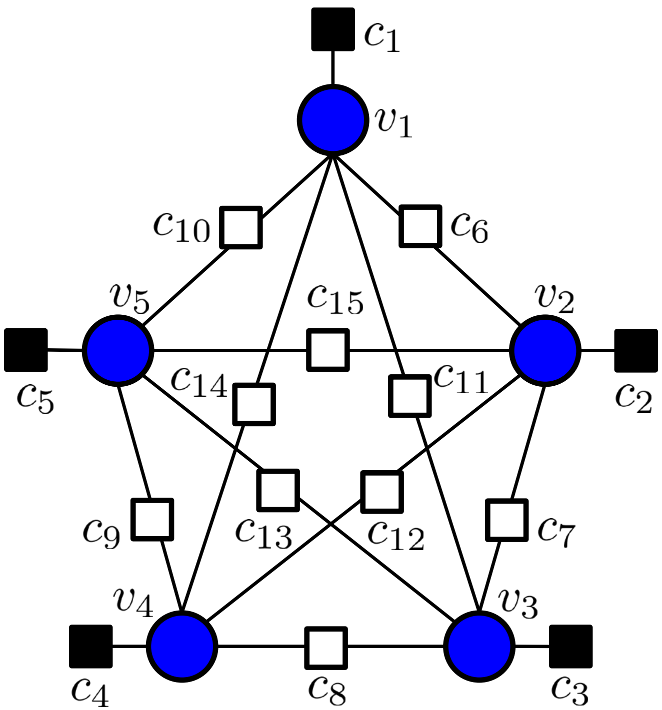
$H_X = [A \mid B]$



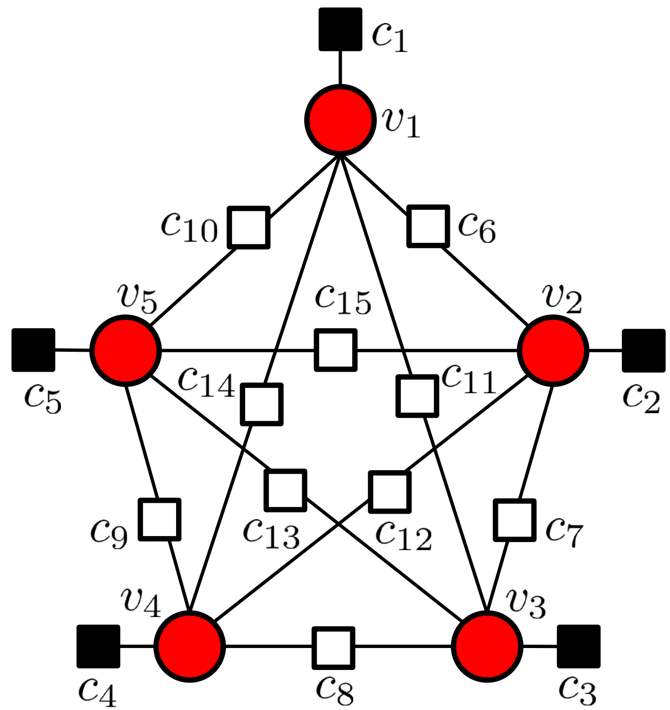
$H_Z = [B^T \mid A^T]$

P. Panteleev and G. Kalachev, arXiv:1904.02703, 2019.

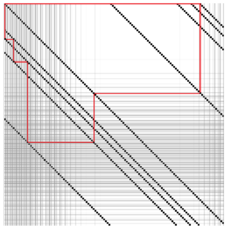
(5,5) TS in A1 code



more harmful - in circulant A

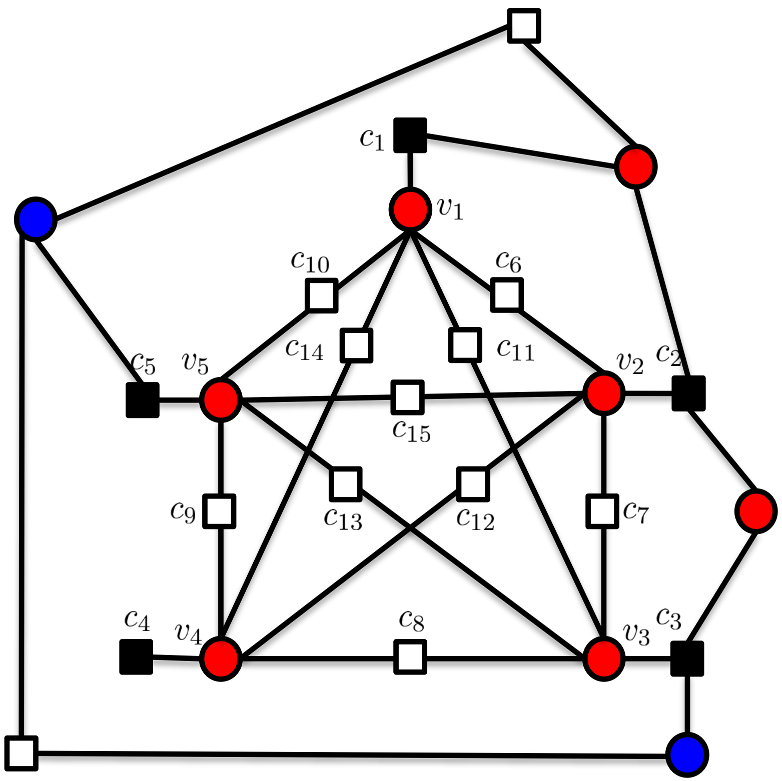


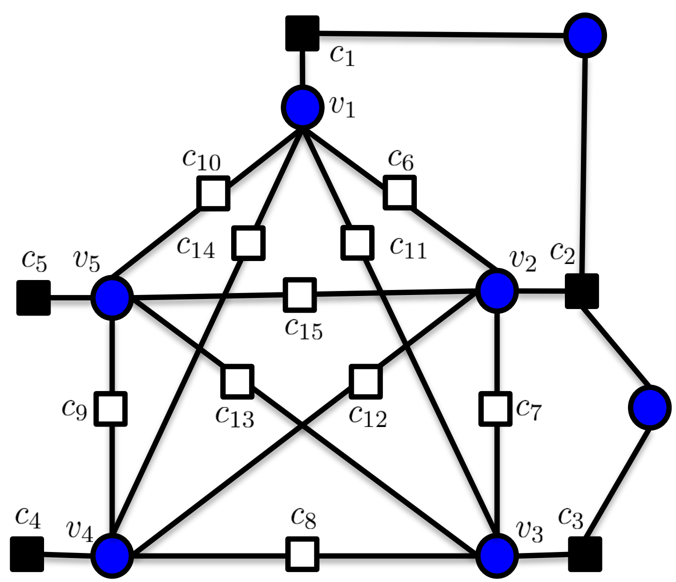
less harmful - in circulant B



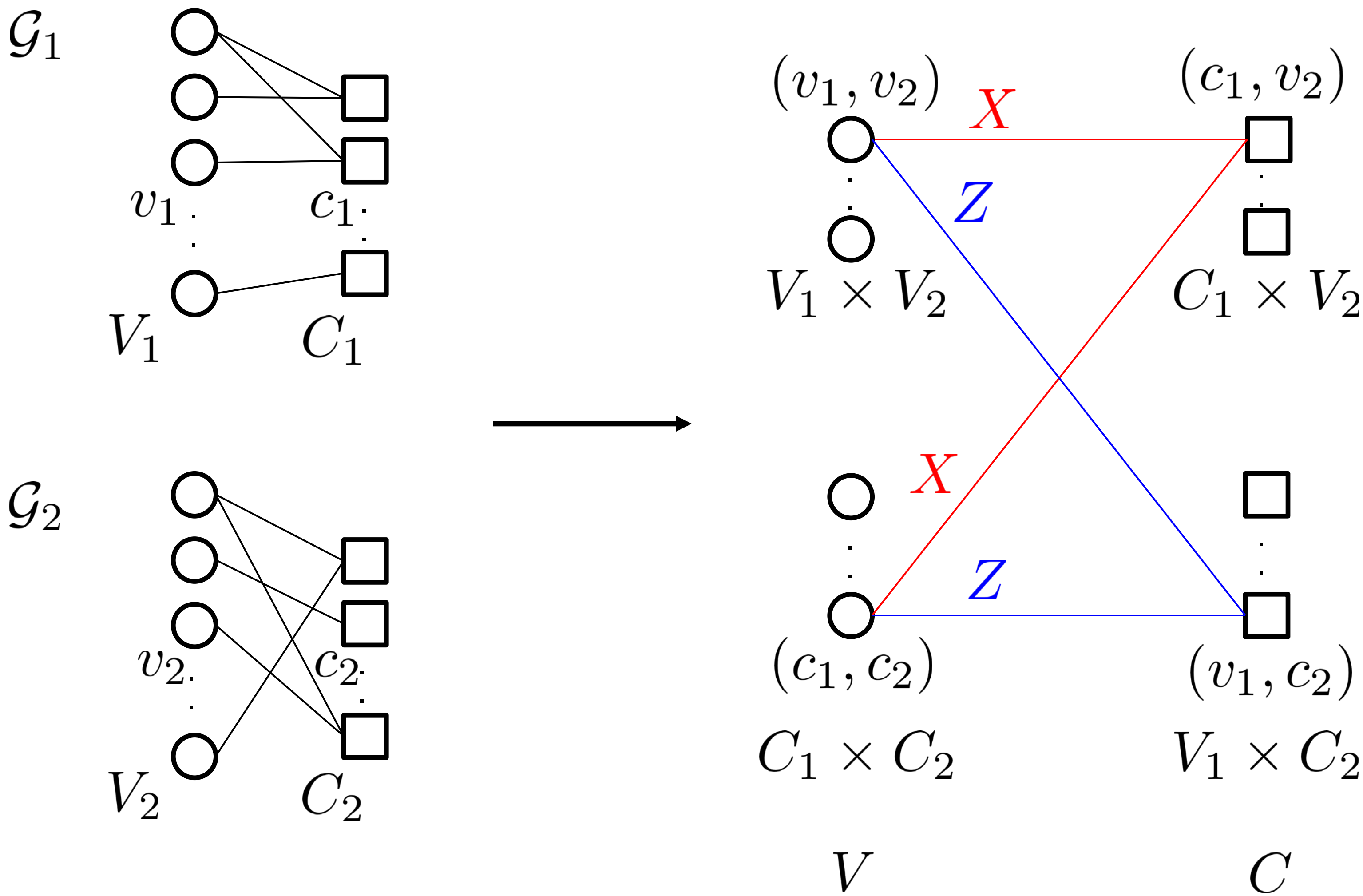
Neighborhood of (5,5)TS is different

- The blue trapping set is more harmful - both floating point min-sum and BP fail on it.

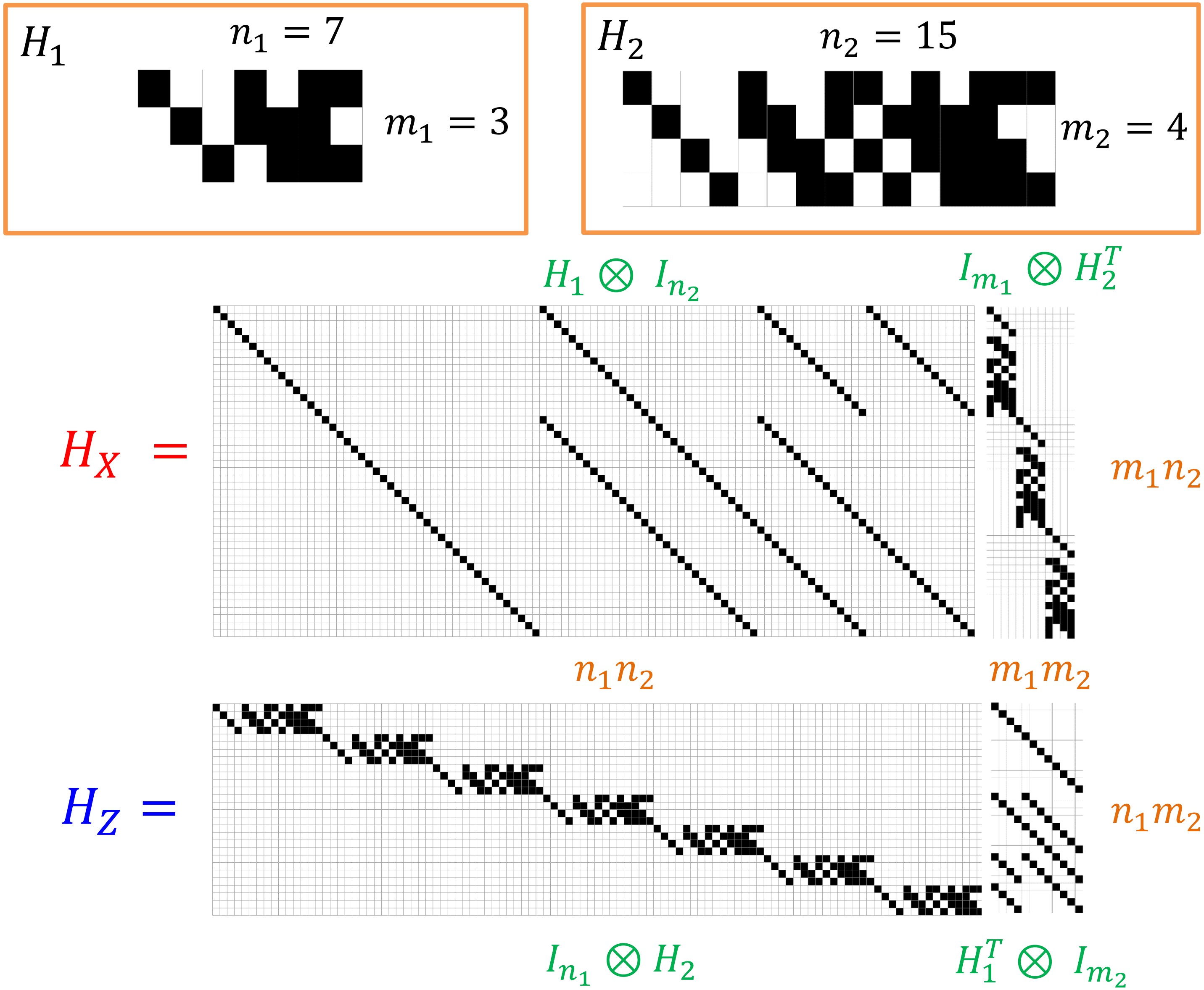




Hyper-graph product (HP) codes



HP code with two classical Hamming codes



Recent example

- Constant rate (4,7) QLDPC code family from (3,4)-LDPC codes

$\mathcal{C}_H$	$\mathcal{C}_H^T$	$\mathcal{HGP}(\mathcal{C}_H)$	$R = k/n$	$\bar{w}$
$[16, 4, 6]$	$[12, 0, \infty]$	$[[400, 16, 6]]$	0.04	7.0
$[20, 5, 8]$	$[15, 0, \infty]$	$[[625, 25, 8]]$	0.04	7.0
$[24, 6, 10]$	$[18, 0, \infty]$	$[[900, 36, 10]]$	0.04	7.0

Roffe et al. May 2020. arXiv:2005.07016 [quant-ph]

Symmetric HP codes

- When C_{H_1} and C_{H_2} are the same - C_H with parameters $[n, k, d]$, the result is a symmetric HP code
  - $H_X = [\, H \otimes \mathrm{I}_n \;\; | \;\; \mathrm{I}_m \otimes H^T]$
  - $H_Z = [\, \mathrm{I}_n \otimes H \;\; | H^T \otimes \mathrm{I}_m]$
  - H is a classical parity check matrix of size $m \times n$.
- $HP(C_H)$ has code parameters:
  - $[[n^2 + m^2 , \; k^2 + (k^T)^2, \min(d, d^T)]]$
  - k^T and d^T are code parameters of the transpose code C_{H^T}
- Commutativity constraint $H_Z \cdot H_X^T = 0$ is satisfied for all binary parity check matrices, thus any classical code to be converted to a quantum code.

J. Tillich and G. Zemor, IEEE Trans. Inf. Theory, 60, 2, 1193–1202, 2014.