2025/12/19

A Brief Overview of some Quantum Erasure Decoding Algorithms

Nicholas Connolly

Nagoya University Quantum Error Correction Theory Workshop 2025

Some Erasure Decoding Papers We Will Discuss

	Paper	Decoder	Code Type
[1]	Luby, M. G., Mitzenmacher, M., Shokrollahi, M. A., & Spielman, D. A. (2002). Efficient erasure correcting codes. *IEEE Transactions on Information Theory*, *47*(2), 569-584.	Peeling decoder	Classical linear LDPC codes
[2]	Connolly, N., Londe, V., Leverrier, A., & Delfosse, N. (2024). Fast erasure decoder for hypergraph product codes. *Quantum*, *8*, 1450.	Pruned peeling and VH decoder	CSS codes and HGP codes
[3]	Delfosse, N., & Zémor, G. (2020). Linear-time maximum likelihood decoding of surface codes over the quantum erasure channel. *Physical Review Research*, *2*(3), 033042.	Surface code spanning-tree peeling decoder	Surface codes
[4]	Delfosse, N., & Nickerson, N. H. (2021). Almost-linear time decoding algorithm for topological codes. *Quantum*, *5*, 595.	Union-find decoder	Surface codes (gen. to qLDPC)
[5]	Nishio, S., Connolly, N., Piparo, N. L., Munro, W. J., Scruby, T. R., & Nemoto, K. (2025). Multiplexed quantum communication with surface and hypergraph product codes. *Quantum*, *9*, 1613.	Surface peeling and VH decoder (**Application*)	Multiplexed toric and HGP codes
[6]	Yao, H., Gökduman, M., & Pfister, H. D. (2025). Cluster decomposition for improved erasure decoding of quantum ldpc codes. *IEEE Journal on Selected Areas in Information Theory*.	Cluster decoder	Any quantum LDPC codes

(Classical) Tanner Graph Representation of a Code

Recall: A (classical binary linear) code C of length n is a subspace of F_2^n.

- C is the kernel of a parity check matrix H.
- The vectors of this subspace are the codewords of C.
- The length n of C is the number of "bits" per codeword.
- The dimension is $k = n - rank(H)$.

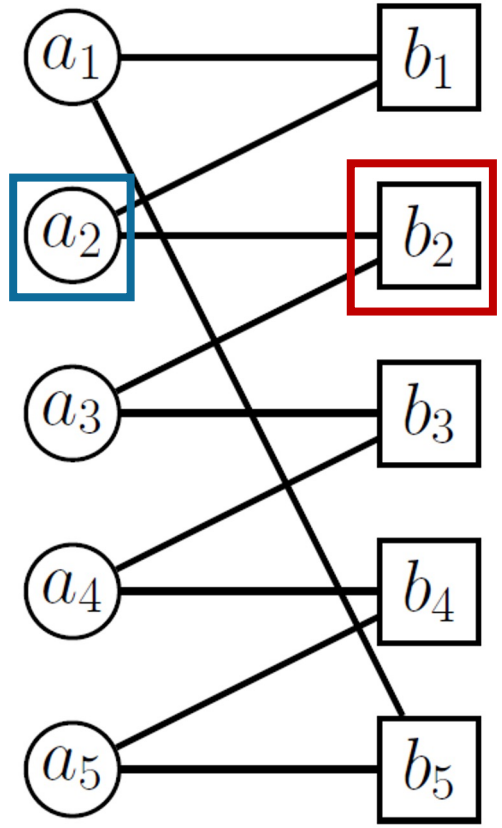
bit a_2

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

check b_2

$$\ker(H) = \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \right\}$$

$T(H) =$

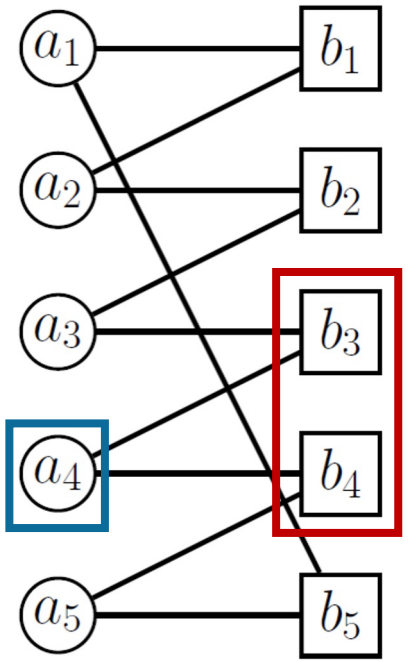


The Tanner Graph of $C = \ker(H)$ is the bipartite graph $T(H)$ defined using H as a bi-adjaceny matrix

- Columns of H are bits of $T(H)$ (circle nodes).
- Rows of H are checks of $T(H)$ (square nodes).

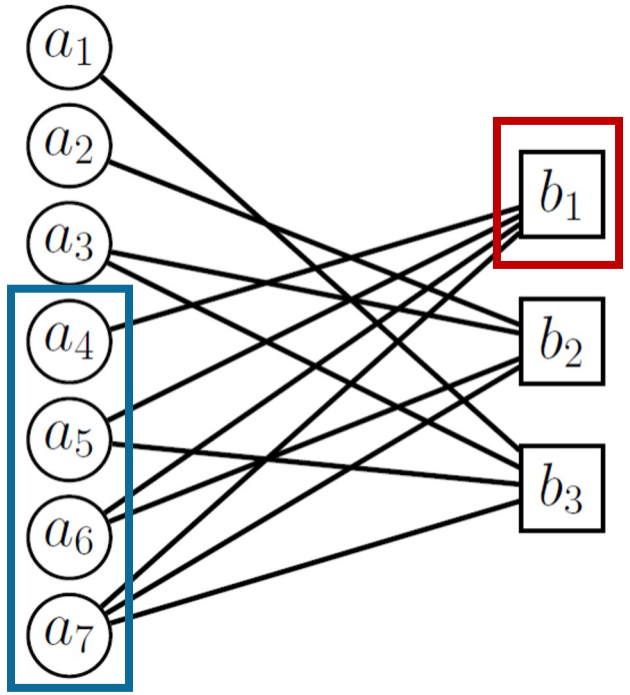
Examples of Parity Check Matrices and Tanner Graphs

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$



5-bit repetition code

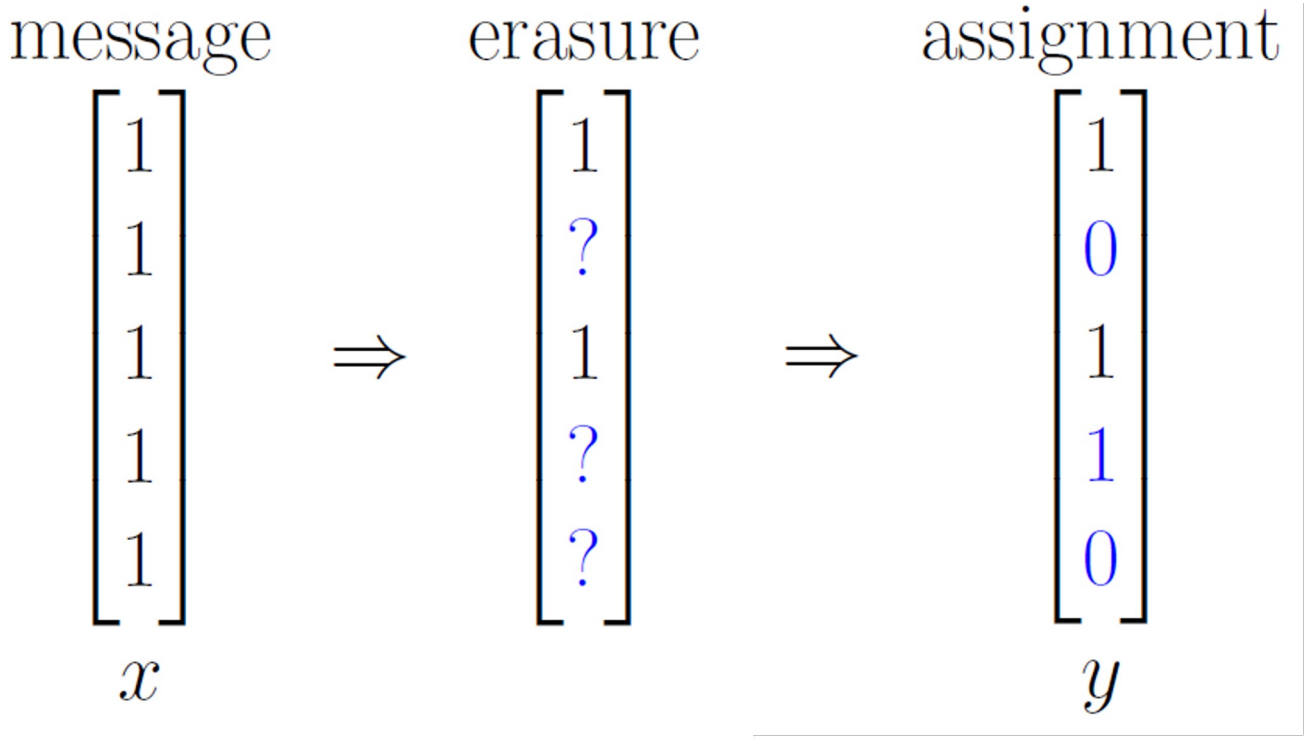
$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$



[7,4,3] Hamming code
(used in Steane code)

(Classical) Binary Erasure Channel and Correction

- A codeword $x \in \ker(H)$ is sent through a noisy erasure channel.
- Each bit in the message x is erased (lost) with probability p.
- Erased bits can be replaced with random bits (0 or 1), yielding corrupted codeword y.
- Corrupted codword y can be corrected using standard (bit-flip) error correction.
- **Additional information:** non-erased bits are known to be correct!



Erasure-Induced Subgraph of the Tanner Graph

- The “erased” part of a codeword $x \in \ker(H)$ can be viewed inside a subcode.
- The submatrix of H corresponding to erased bits in x defines a subgraph of $T(H)$.
- Degree-1 check nodes in the subgraph carry perfect information about adjacent bits.

$$x = \begin{bmatrix} 1 \\ 1 \\ ? \\ ? \\ ? \\ 1 \\ 0 \end{bmatrix}$$

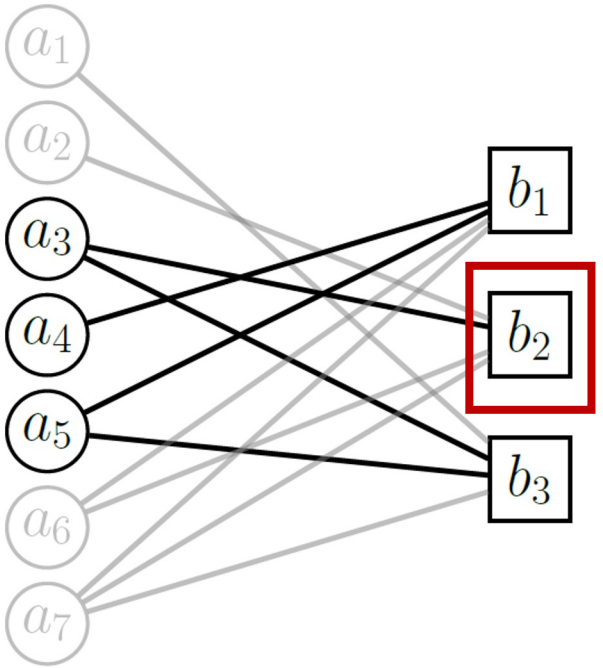
$$x' =$$

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

$$H' =$$

$T(H) =$

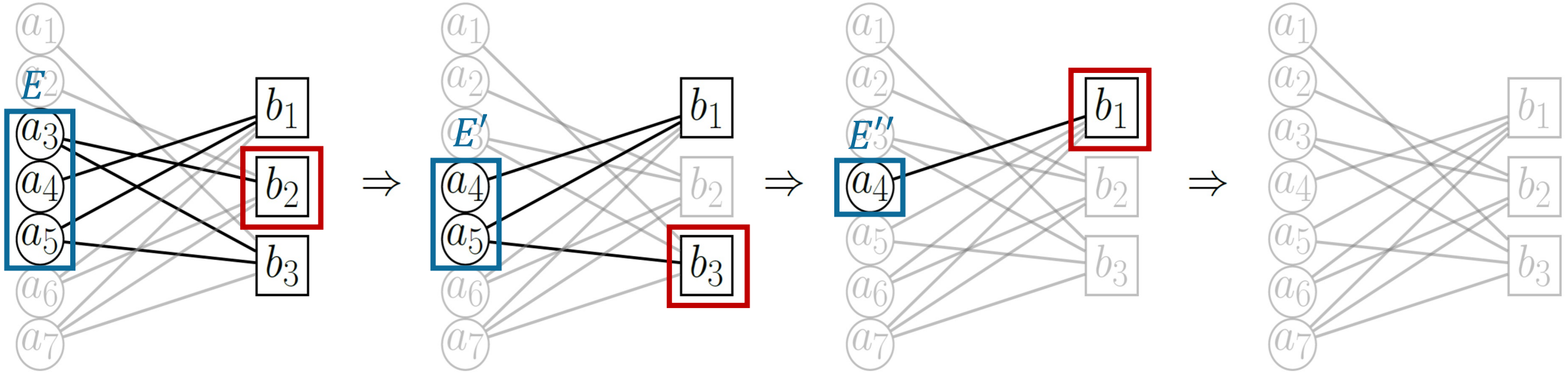
$T(H') =$



Classical Peeling Decoder Algorithm

1. Given erasure vector $E \in F_2^n$, consider the erasure-induced subgraph of $T(H)$.
2. Select a dangling (degree 1) check in this subgraph.
3. Correct the value of the single adjacent erased bit based on the dangling check.
4. Remove this bit from E, shrinking in the erasure ("peeling").
5. Repeat until the erasure is empty, or no dangling checks remain.

$E''' = \emptyset$



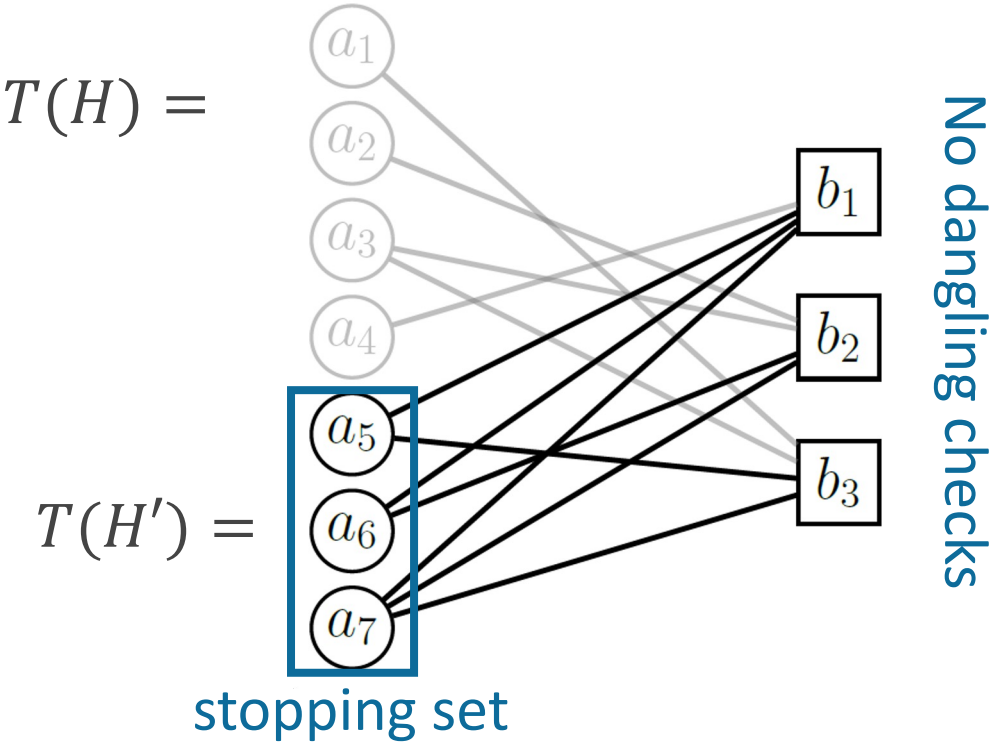
Luby, M. G., Mitzenmacher, M., Shokrollahi, M. A., & Spielman, D. A. (2002). Efficient erasure correcting codes. *IEEE Transactions on Information Theory*, *47*(2), 569-584.

Stopping Sets for the Peeling Decoder

- A decoding success or failure depends *only* on the erasure vector.
- Success occurs if there exists a sequence of dangling checks that peel the whole erasure.
- Failure occurs when there exists a non-empty erasure vector with no dangling checks.
- An erasure-induced subgraph of $T(H)$ with no dangling checks is called a stopping set.
- Tanner graphs for sparse (LDPC) codes have fewer stopping sets.

$$H = \left[\begin{array}{cccc|ccc} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{array}\right]$$

$H' =$



Review: Quantum CSS Codes from Classical Codes

- A quantum code of length N and dimension K is a subspace of a Hilbert space.
  - Vectors are N-qubit states $|\psi\rangle \in C^N$.
  - Errors on $|\psi\rangle$ are represented by N-qubit Pauli operators in $P_N = \{I, X, Y, Z\}^{\otimes N}$.
- A stabilizer code is the space of states left fixed by a subgroup of the Pauli group P_N.
  - Defined by a set of stabilizer generators $\langle g_1, \cdots, g_r \rangle = G \subseteq P_N$
- A CSS code $C = CSS(C_X, C_z)$ is a stablizer code defined from two classical codes.
  - Defined by *commuting* Pauli X- and Z-operators as stabilizer generators.
  - These define the rows of matrices H_X and H_Z satisftying $H_X H_Z^T = 0$ (commutivity).
  - Z- and X-error correction is modeled by classical $C_X = \ker(H_X)$ and $C_Z = \ker(H_Z)$.

Example:
Steane Code

$$H_X = H_Z = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

X-stablizers:	Z-stablizers:	
$IIIXXXX$	$IIIZZZZ$	
$IXXIIXX$	$IZZIIZZ$	$\in P_7$
$XIXIXIX$	$ZIZIZIZ$	

(Quantum) Peeling Decoder for CSS Codes

- For CSS code $C = CSS(C_X, C_Z)$, X- and Z-type Pauli errors can be corrected separately.
- Pauli errors in P_N can be mapped onto binary strings in Z_2^N.

$$E = X_1 Z_1 X_2 Z_4 \in P_4 \quad \Leftrightarrow \quad e_X = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad e_Z = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

- Pauli error correction is modeled by classical error correction with C_X and C_Z.
- For erasure errors, the Peeling Decoder can be directly applied.
- However, it works poorly due to certain quantum features.

Hypergraph Product (HGP) Codes

Theorem (Tillich-Zémor):

The Hypergraph Product code $C = HGP(C_1, C_2)$ of two classical codes $C_1 = \ker(H_1)$ and $C_2 = \ker(H_2)$ is the quantum code $C = CSS(C_X, C_Z)$, where $C_X = \ker(H_X)$ and $C_Z = \ker(H_Z)$ have parity check matrices H_X and H_Z defined from H_1 and H_2.

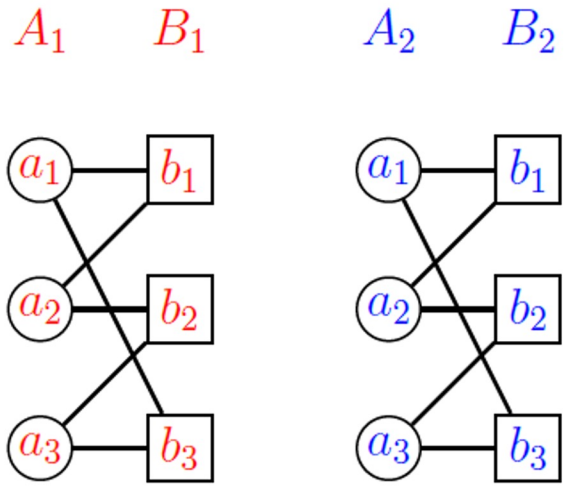
$$H_X = \left[\begin{array}{c|c} H_1 \otimes I & I \otimes H_2^T \end{array} \right]$$

$$H_Z = \left[\begin{array}{c|c} I \otimes H_2 & H_1^T \otimes I \end{array} \right]$$

- The matrices have sizes $H_1 = [r_1 \times n_1], H_2 = [r_2 \times n_1]$, so $H_X = [r_1 n_2 \times (n_1 n_2 + r_1 r_2)], H_Z = [r_2 n_1 \times (n_1 n_2 + r_1 r_2)]$.
- C has length $N = n_1 n_2 + r_1 r_2$ and dimension $K = N - rank(H_X) - rank(H_Z)$.
- C has minimum distance $\min(d_1, d_2)$ where d_1 and d_2 are the minimum distances of C_1 and C_2.

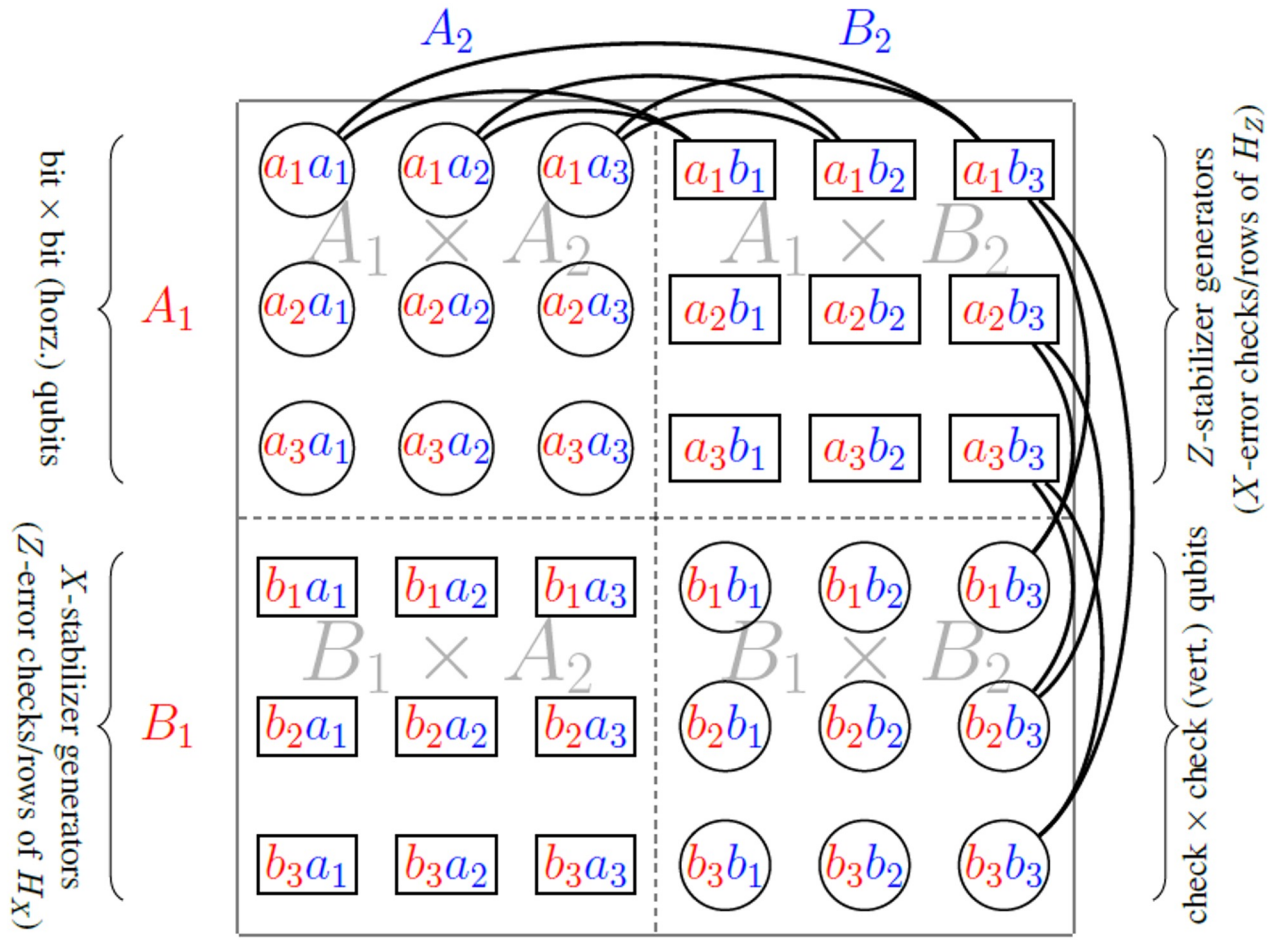
Tillich, J. P., & Zémor, G. (2013). Quantum LDPC codes with positive rate and minimum distance proportional to the square root of the blocklength. *IEEE Transactions on Information Theory*, *60*(2), 1193-1202.

Tanner Graph Structure of Hypergraph Product Codes



$$T(H_1) \cong T(H_2)$$

$$H_1 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} = H_2$$



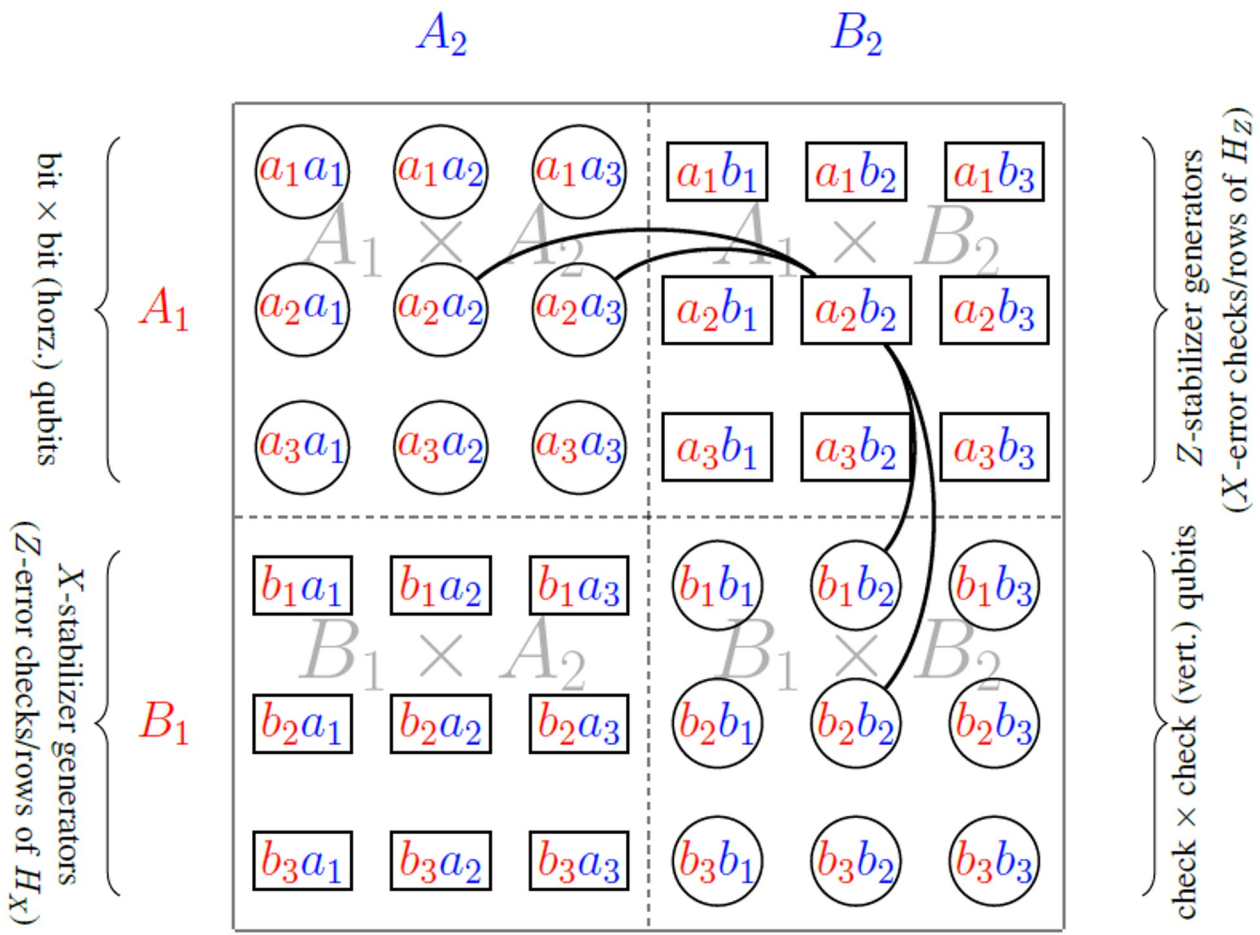
$$H_Z = \begin{bmatrix} 110000000 & 100000100 \\ 011000000 & 010000010 \\ 101000000 & 001000001 \\ 000110000 & 100100000 \\ 000011000 & 010010000 \\ 000101000 & 001001000 \\ 000000110 & 000100100 \\ 000000011 & 000010010 \\ 000000101 & 000001001 \end{bmatrix}$$

$$[\ I \otimes H_2 \mid H_1^T \otimes I\ ]$$

$$H_X = \begin{bmatrix} 100100000 & 101000000 \\ 010010000 & 110000000 \\ 001001000 & 011000000 \\ 000100100 & 000101000 \\ 000010010 & 000110000 \\ 000001001 & 000011000 \\ 100000100 & 000000101 \\ 010000010 & 000000110 \\ 001000001 & 000000011 \end{bmatrix}$$

$$[\ H_1 \otimes I \mid I \otimes H_2^T\ ]$$

HGP Code: Z-Type Stabilizer Generators



$H_Z =$

a_2a_2	a_2a_3	b_1b_2	b_2b_2
1100	0000	1000	0100
0110	0000	0100	0010
1010	0000	0010	0001
0001	1000	1001	0000
0000	1100	0110	0000
0001	0100	0011	1000
0000	0011	0001	0100
0000	0101	0000	1001

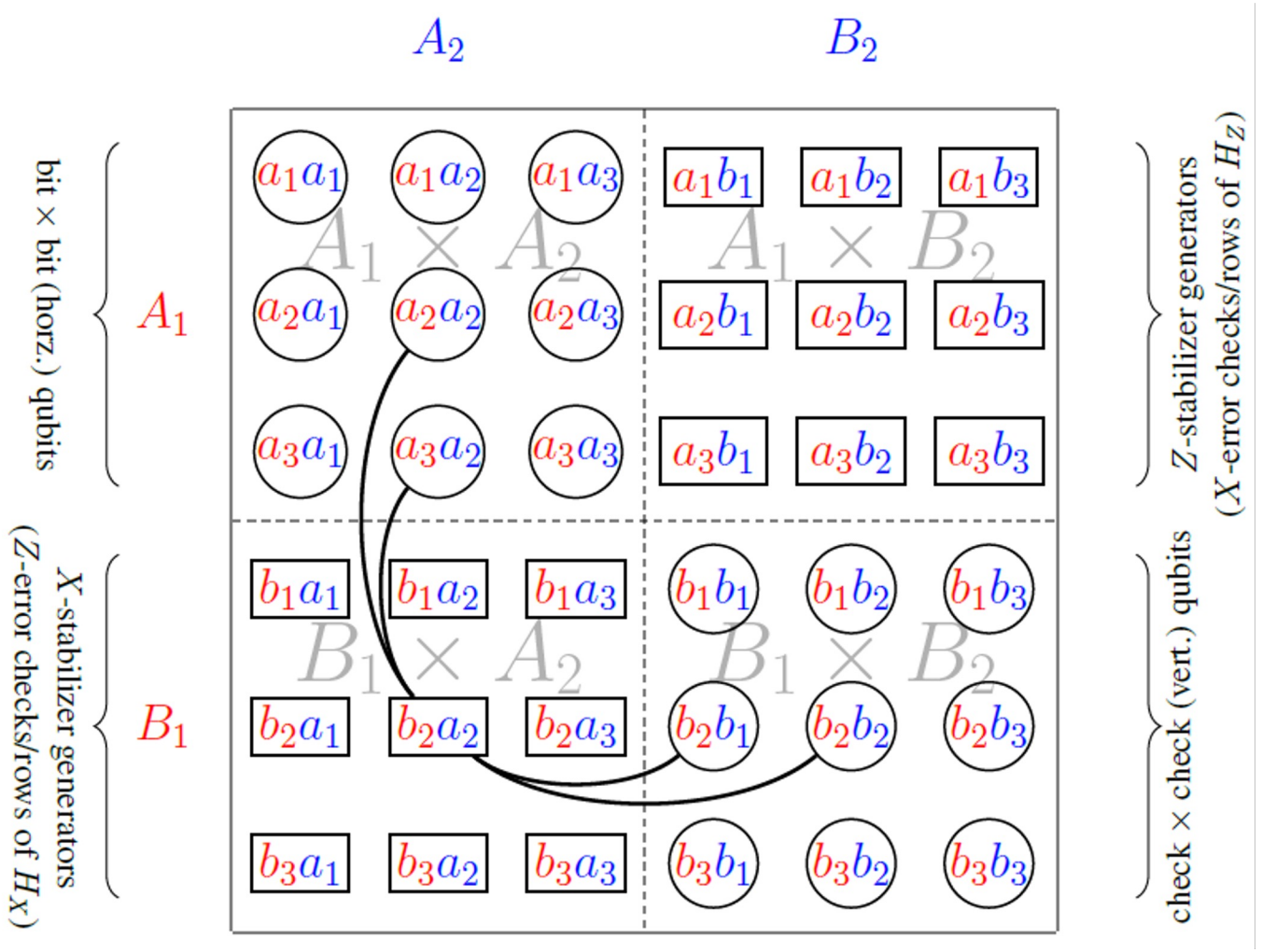
$$\left[\; I \otimes H_2 \;\middle|\; H_1^T \otimes I \;\right]$$

HGP Code: X-Type Stabilizer Generators

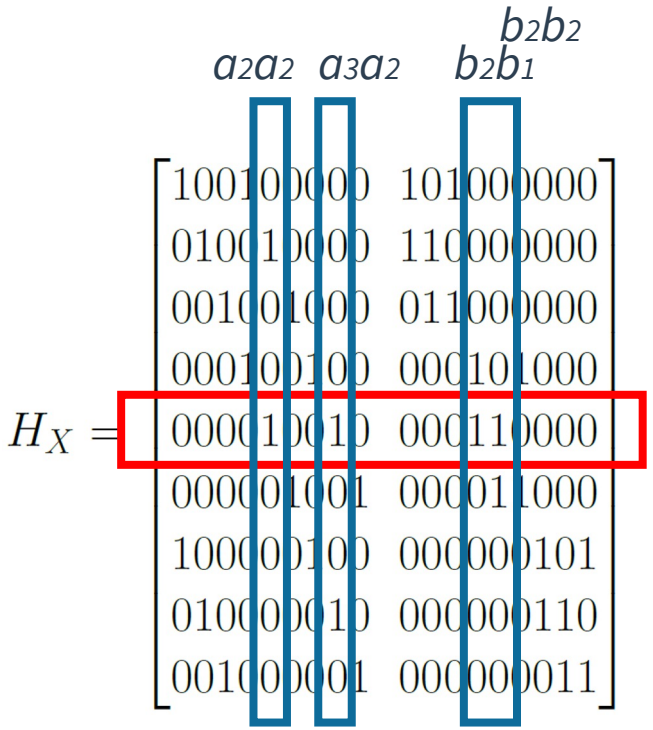
a_2a_2 a_3a_2 b_2b_1 b_2b_2

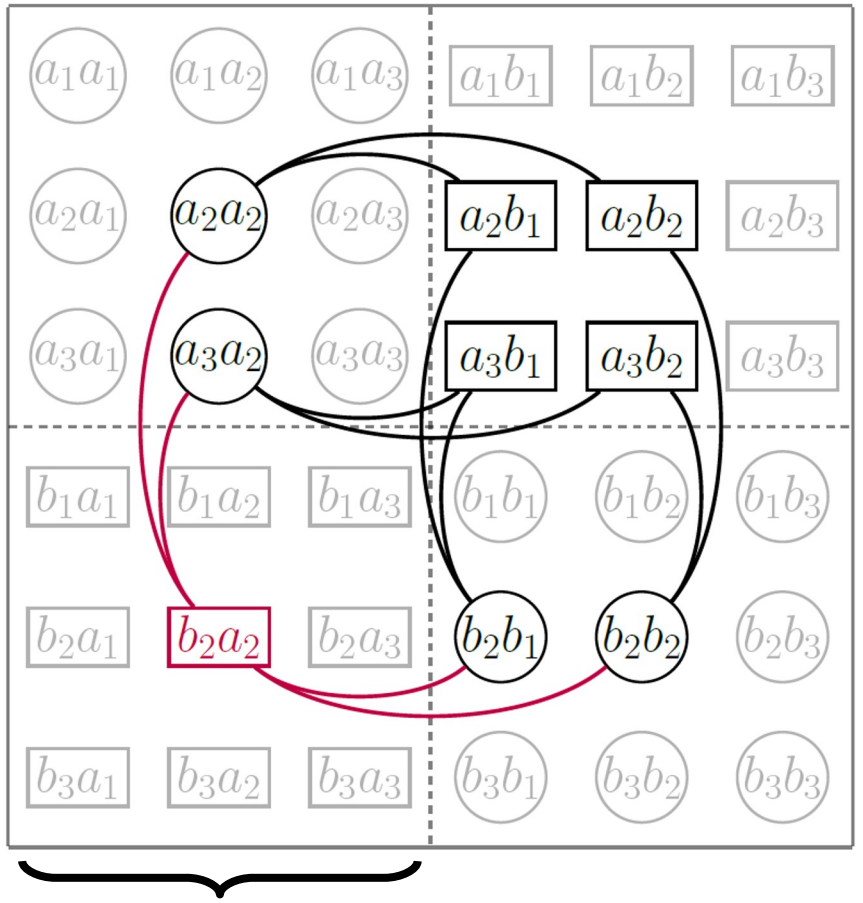
$$H_X = \begin{bmatrix}
1001\,0000\,0 & 1010\,0000\,0 \\
0100\,1000\,0 & 1100\,0000\,0 \\
0010\,0100\,0 & 0110\,0000\,0 \\
0001\,0010\,0 & 0001\,0100\,0 \\
0000\,1001\,0 & 0001\,1000\,0 \\
0000\,0100\,1 & 0000\,1100\,0 \\
1000\,0010\,0 & 0000\,0010\,1 \\
0100\,0001\,0 & 0000\,0011\,0 \\
0010\,0000\,1 & 0000\,0001\,1
\end{bmatrix}$$

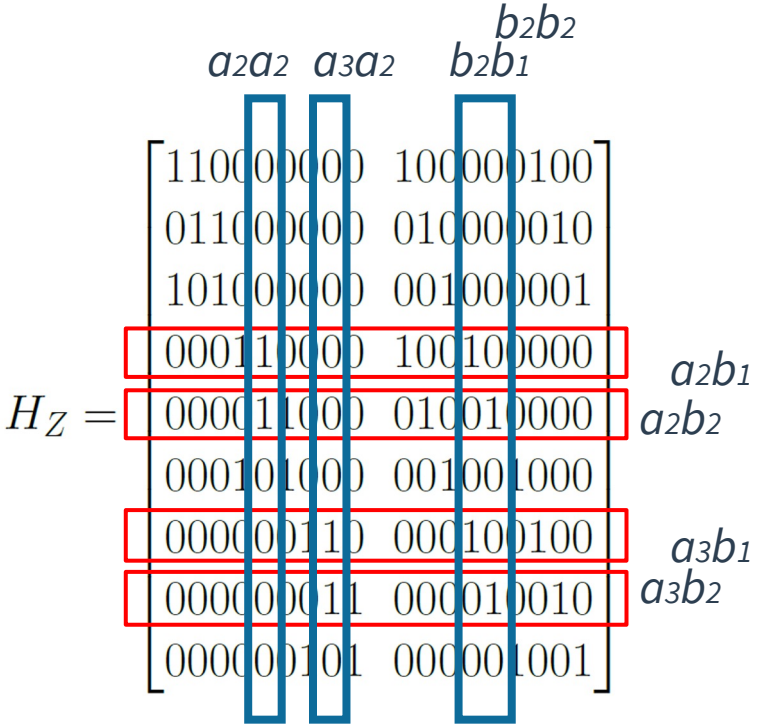
$$\left[\; H_1 \otimes I \;\middle|\; I \otimes H_2^T \;\right]$$



Visualizing Stabilizer Support in the HGP Tanner Graph



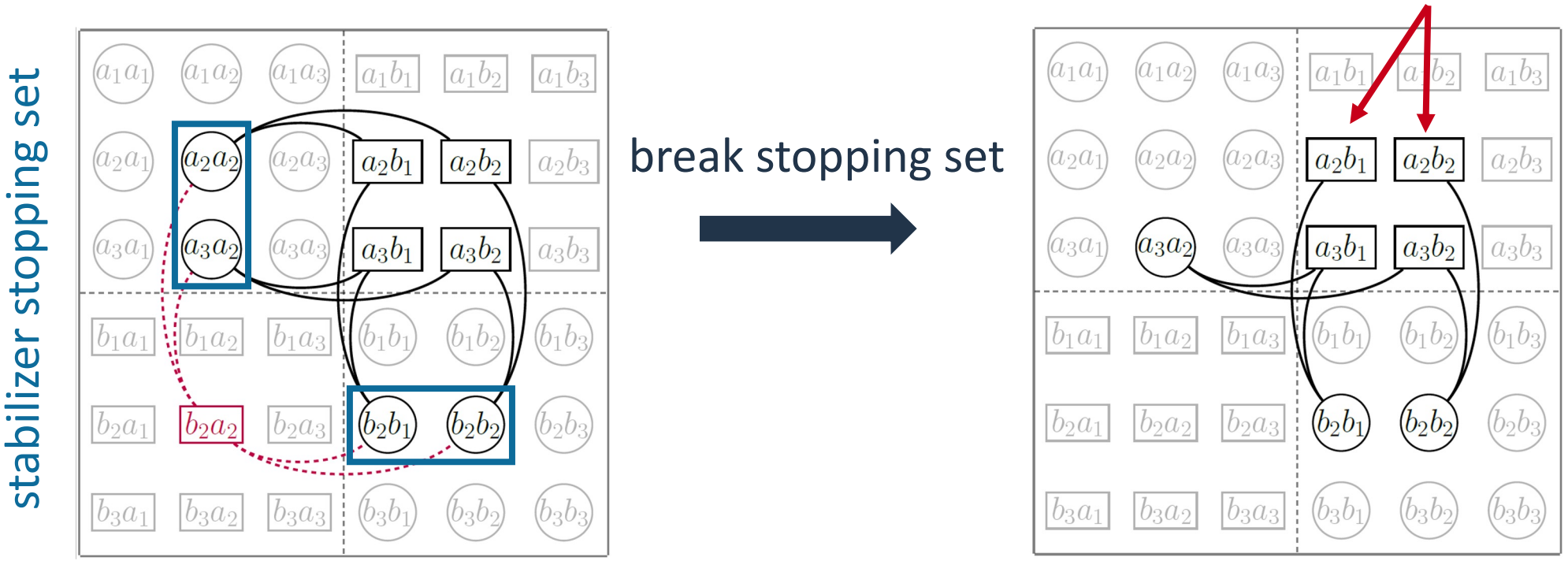




X-stabilizer generators (rows of H_X)

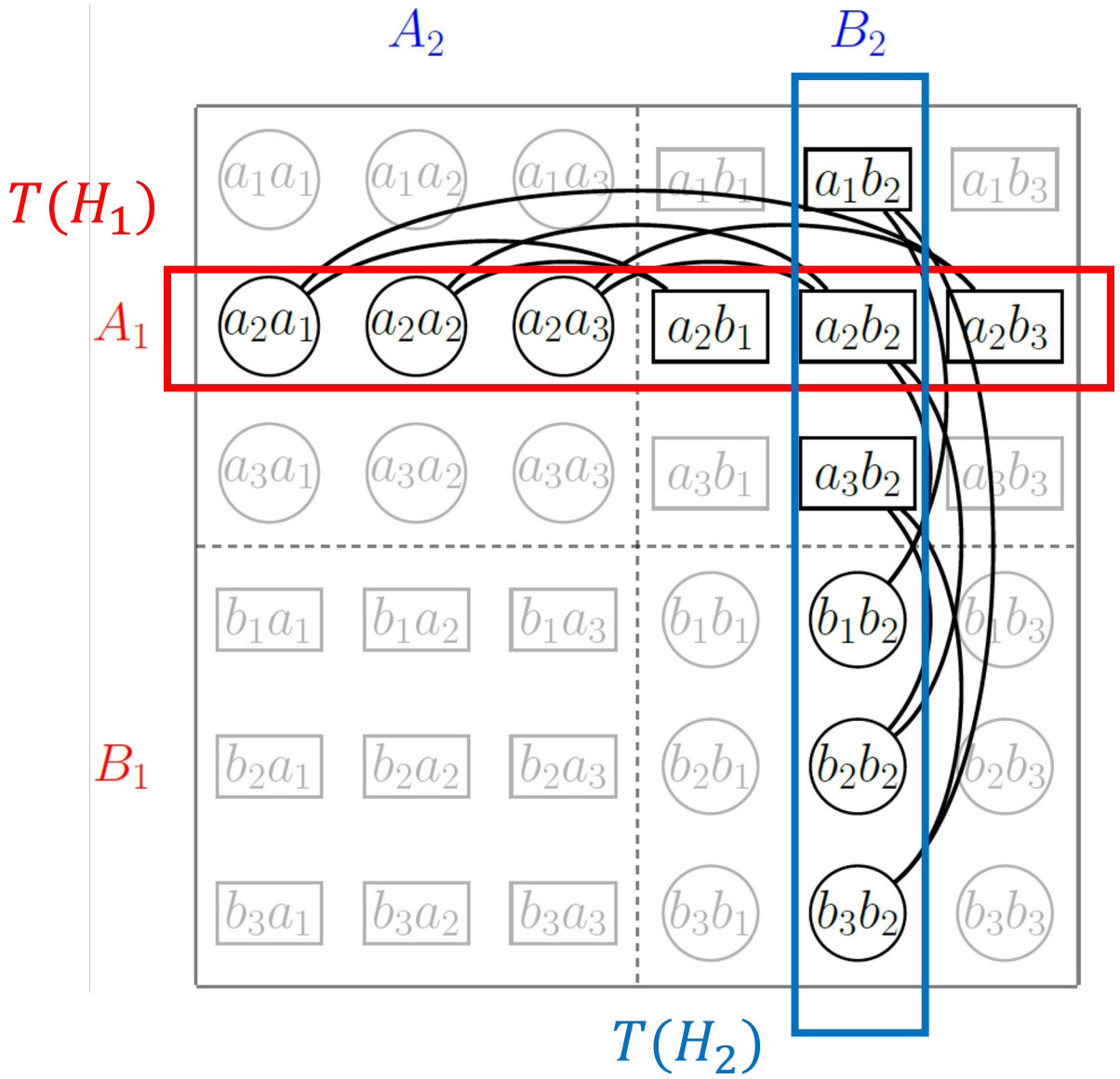
Stabilizer Stopping Sets and the Pruned Peeling Decoder

- The qubit-support of an X- or Z-stabilizer is a stopping set of $\ker(H_Z)$ or $\ker(H_X)$.
  - This is a consequence of commuting X- and Z-stabilizers for CSS codes.
- Break the stopping set by removing a single qubit from the erasure.
  - This is valid thanks to a feature of stabilizer codes.
- If there are new dangling checks, continue peeling.



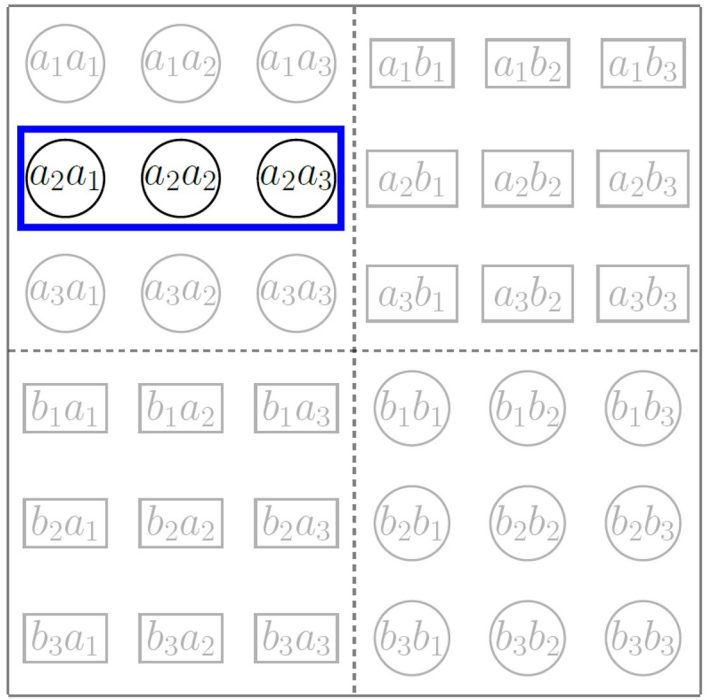
Vertical and Horizontal Stopping Sets for HGP Codes

- Peeling decoder stopping sets in classical codes C_1 and C_2 "lift" to HGP code stopping sets on a single row or column in $HGP(C_1, C_2)$.
  - Visualized in the Tanner graph.
  - Supported on row or column.
- These are classical stopping sets.
- Can be clustered by connected components in same row/column.
  - Clusters share at most 1 check.
  - Clusters define the VH graph.

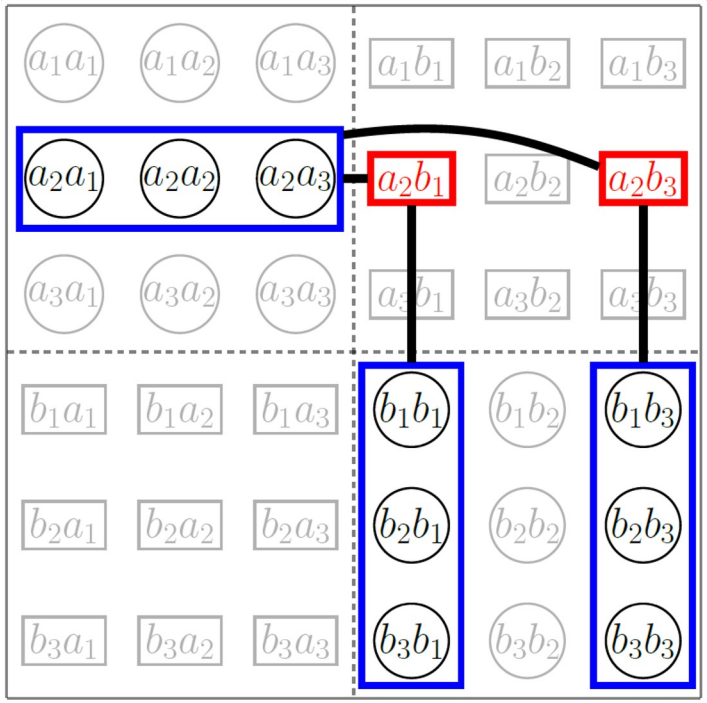


Types of Cluster Configurations in the VH Graph

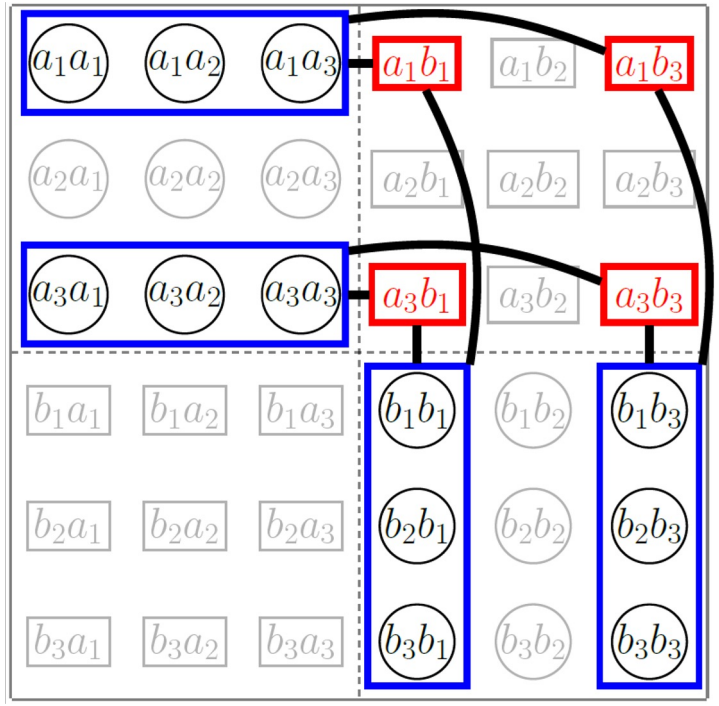
- **Vertices:** classical stopping set clusters in the same row/column
- **Edges:** checks shared between two clusters



Isolated Cluster



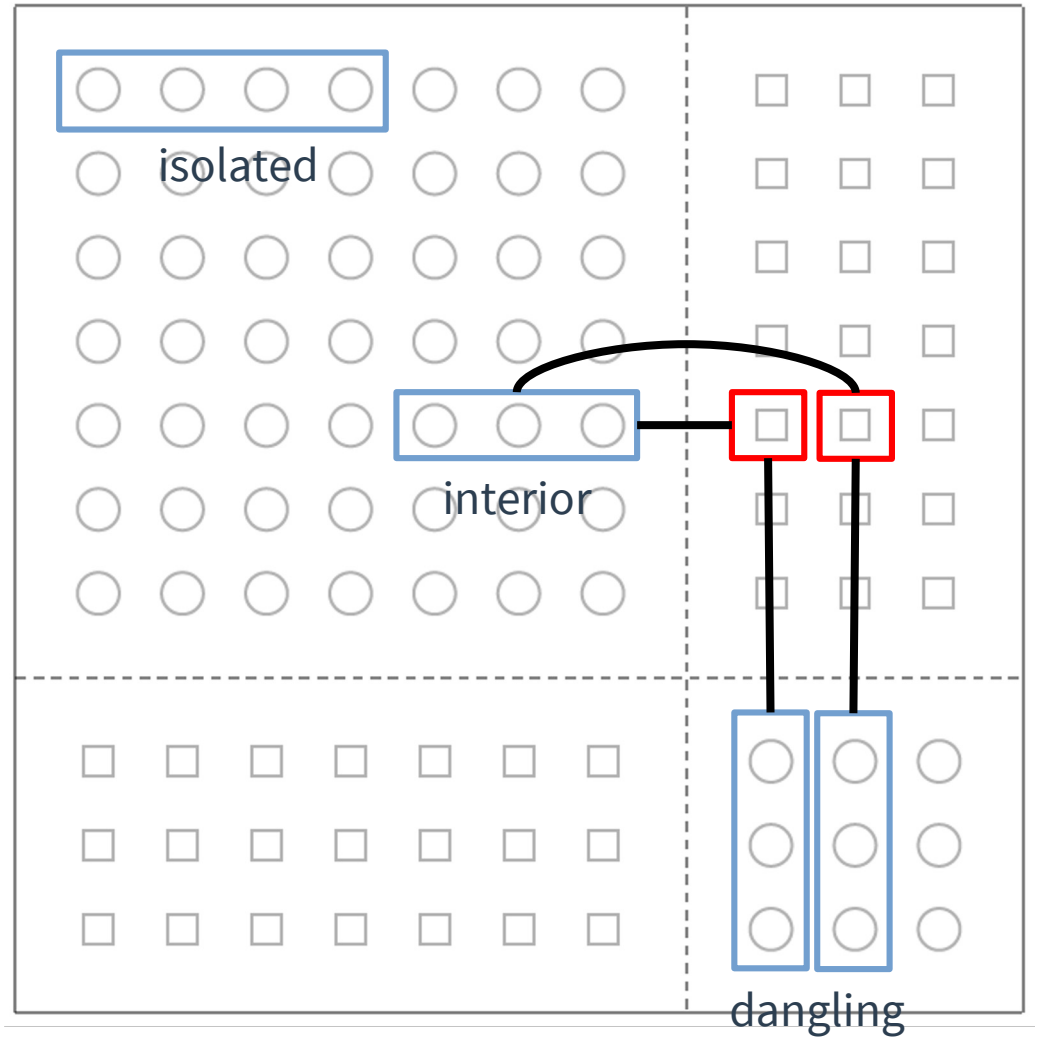
Cluster Tree



Cluster Cycle

The VH Decoder for HGP Codes

- A peeling decoder stopping set is clustered into the VH graph.
  - Isolated, dangling, or interior
  - Solvable via Gaussian elimination
- Clusters are solved in sequentially*.
  - Eliminate all isolated clusters.
  - Peel dangling clusters from trees.
  - *(Accounting for *free* vs *frozen*)
- Terminates when all clusters are solved, or a cluster cycle remains.
- $O(N^2)$ complexity on code length N.



Connolly, N., Londe, V., Leverrier, A., & Delfosse, N. (2024). Fast erasure decoder for hypergraph product codes. *Quantum*, *8*, 1450.

Peeling, Pruned Peeling, and VH-Decoder Pseudocode

Algorithm 1: Classical peeling decoder

input : An erasure vector $\varepsilon \in \mathbb{Z}_2^N$ and a syndrome $s \in \mathbb{Z}_2^r$.

output: Either **failure** or $\hat{e} \in \mathbb{Z}_2^n$ such that $H\hat{e} = s$ and $\mathrm{supp}(\hat{e}) \subseteq \mathrm{supp}(\varepsilon)$.

1 Set $\hat{e} = 0$.
2 **while** *there exists a dangling check* **do**
3 Select a dangling check c_i.
4 Let b_j be the dangling bit incident to c_i.
5 **if** $s_i = 1$ **then**
6 Flip the j-th bit of $\hat{e}$.
7 Flip s_k for all checks c_k incident with b_j.
8 Set $\varepsilon_j = 0$.
9 **if** $\varepsilon \neq 0$ **return Failure**, **else return** $\hat{e}$.

Algorithm 2: Pruned peeling decoder

input : An erasure vector $\varepsilon \in \mathbb{Z}_2^N$, a syndrome $s \in \mathbb{Z}_2^{R_Z}$, and an integer M.

output: Either **Failure** or an X-type error $\hat{E} \in \{I, X\}^N$ such that $\sigma(\hat{E}) = s$ and $\mathrm{supp}(\hat{E}) \subseteq \mathrm{supp}(\varepsilon)$.

1 Set $\hat{E} = I$.
2 **while** *there exists a dangling generator* **do**
3 Select a dangling generator $S_{Z,i}$.
4 Let j be the index of the dangling qubit incident to $S_{Z,i}$.
5 **if** $s_i = 1$ **then**
6 Replace $\hat{E}$ by $\hat{E}X_j$ and s by $s + \sigma(X_j)$.
7 Set $\varepsilon_j = 0$.
8 **if** *there exists a product S of up to M stabilizer generators among $S_{X,1}, \ldots, S_{X,R_X}$ such that* $\mathrm{supp}(S) \subseteq \mathrm{supp}(\varepsilon)$ **then**
9 Select a qubit in $\mathrm{supp}(S)$ with index j and set $\varepsilon_j = 0$.
10 Go back to step 2.
11 **if** $\varepsilon \neq 0$ **return Failure**, **else return** $\hat{E}$.

Algorithm 3: VH decoder

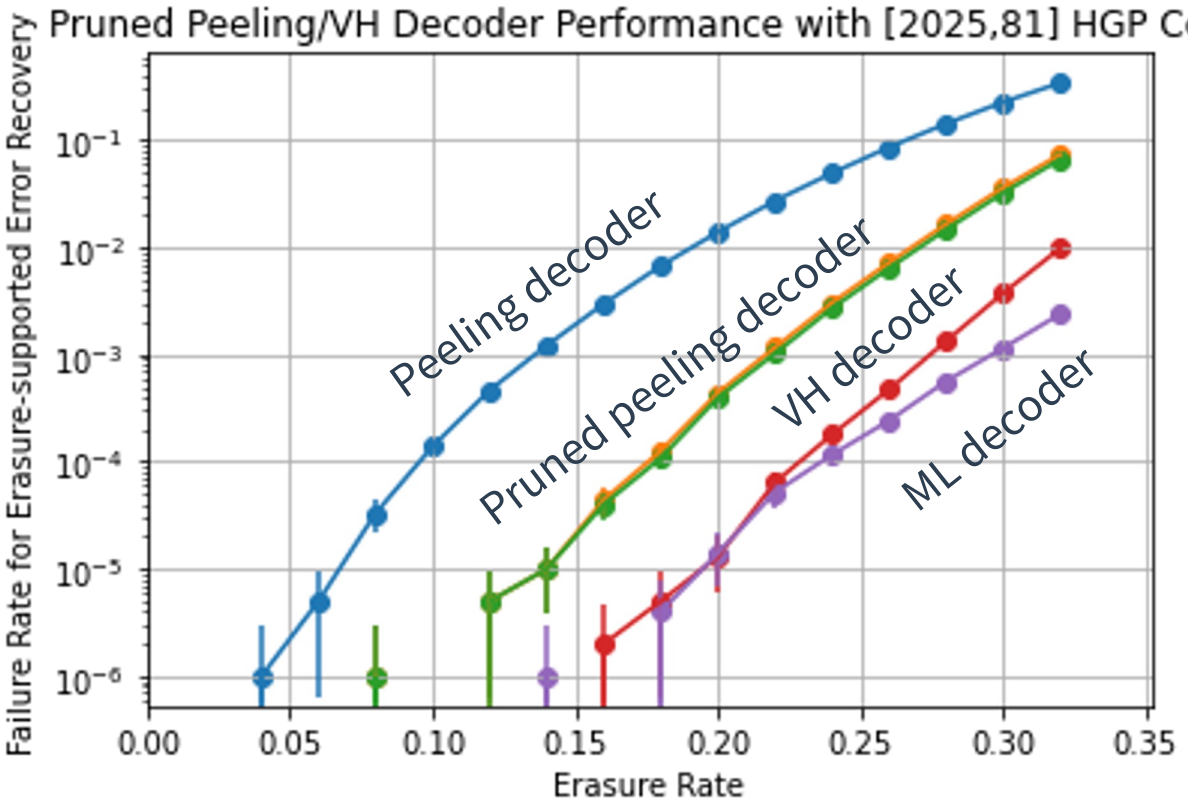
input : An erasure vector $\varepsilon \in \mathbb{Z}_2^N$, a syndrome $s \in \mathbb{Z}_2^{R_Z}$.

output: Either **Failure** or an X-type error $\hat{E} \in \{I, X\}^N$ such that $\sigma(\hat{E}) = s$ and $\mathrm{supp}(\hat{E}) \subseteq \mathrm{supp}(\varepsilon)$.

1 Set $\hat{E} = I$.
2 Construct an empty stack $L = []$.
3 **while** *there exists an isolated or a dangling cluster κ* **do**
4 **if** *κ is isolated or frozen* **then**
5 Compute an error $\hat{E}_\kappa$ supported on κ whose syndrome matches s on the internal checks of κ in $T(\mathbf{H}_Z)$ (using the Gaussian decoder).
6 Replace $\hat{E}$ by $\hat{E}\hat{E}_\kappa$ and s by $s + \sigma(\hat{E}_\kappa)$.
7 **For** all qubits j in κ, set $\varepsilon_j = 0$.
8 **else**
9 Then κ is free.
10 Remove the free connecting check c of κ from the Tanner graph $T(\mathbf{H}_Z)$.
11 Add the pair (κ, c) to the stack L.
12 **For** all qubits j in κ, set $\varepsilon_j = 0$.
13 **while** *the stack L is non-empty* **do**
14 Pop a cluster (κ, c) from the stack L.
15 Add the check node c to the Tanner graph $T(\mathbf{H}_Z)$.
16 Compute an error $\hat{E}_\kappa$ supported on κ whose syndrome matches s on all the checks of κ in $T(\mathbf{H}_Z)$, including the free check c (using the Gaussian decoder).
17 Replace $\hat{E}$ by $\hat{E}\hat{E}_\kappa$ and s by $s + \sigma(\hat{E}_\kappa)$.
18 **if** $\varepsilon \neq 0$ **return Failure**, **else return** $\hat{E}$.

Connolly, N., Londe, V., Leverrier, A., & Delfosse, N. (2024). Fast erasure decoder for hypergraph product codes. *Quantum*, *8*, 1450.

Comparison of Decoder Performances with HGP Codes

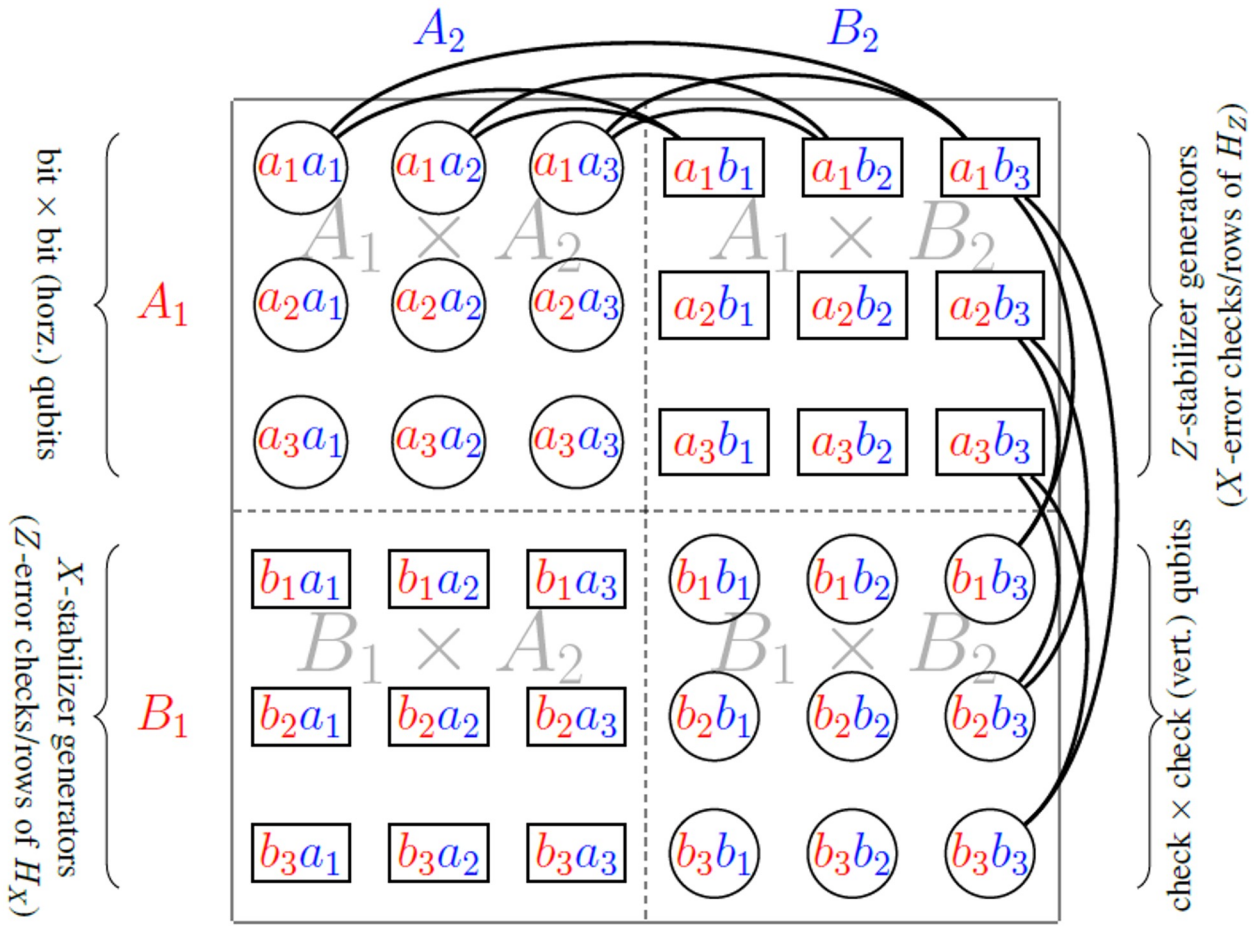


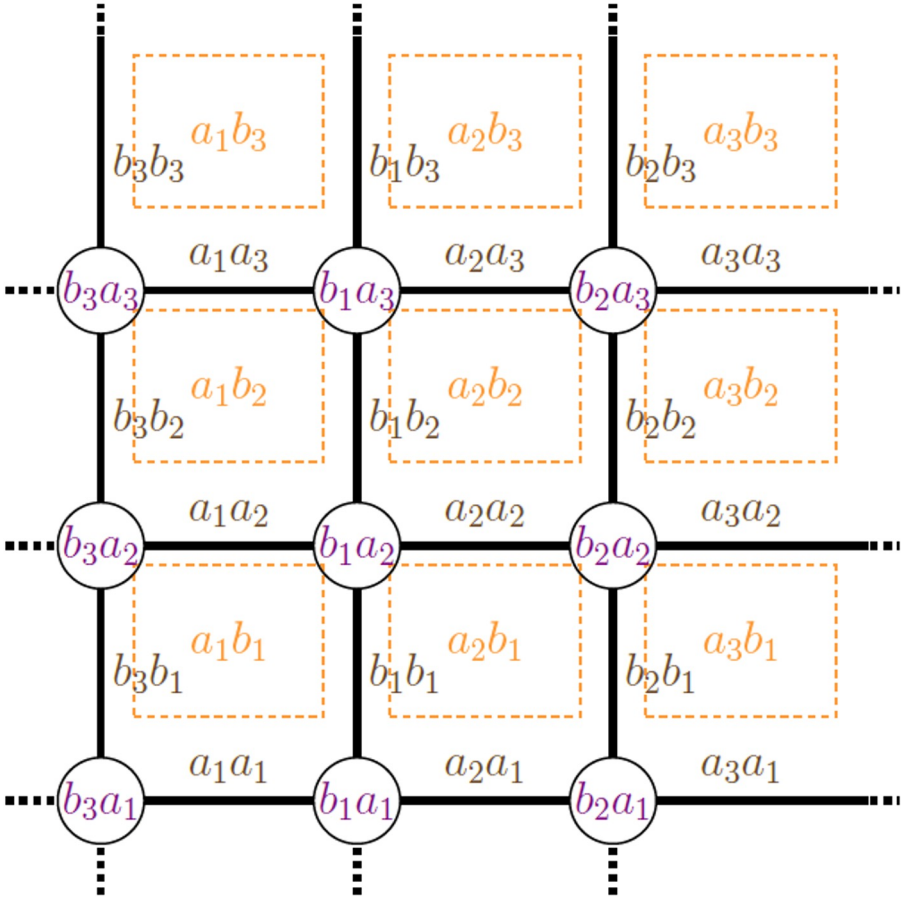
Number of qubits: 2025
Maximum erasure rate: 0.32
Number of different erasure rates: 16
Randomized trials per data point (pruned peeling and VH decoder): 10^6
Randomized trials per data point (Gaussian decoder): 10^6
y-axis scaling: logarithmic

- pruned peeling decoder (M=0)
- pruned peeling decoder (M=1)
- pruned peeling decoder (M=2)
- pruned peeling (M=2) + VH decoder
- Gaussian decoder

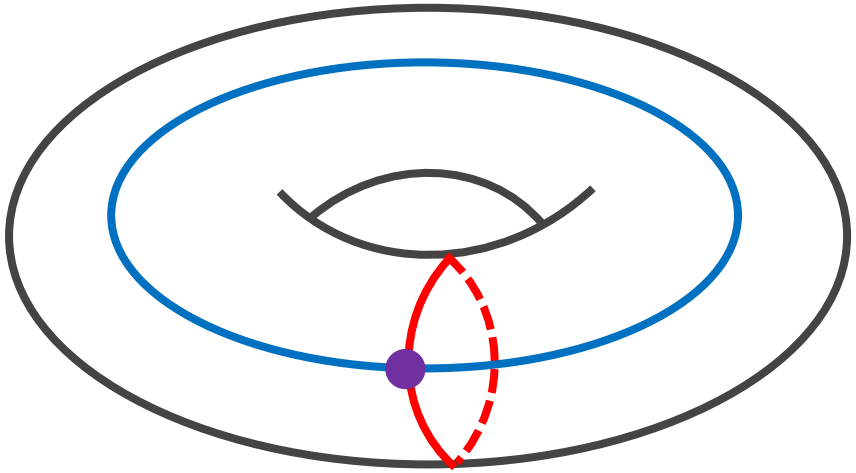
Connolly, N., Londe, V., Leverrier, A., & Delfosse, N. (2024). Fast erasure decoder for hypergraph product codes. *Quantum*, *8*, 1450.

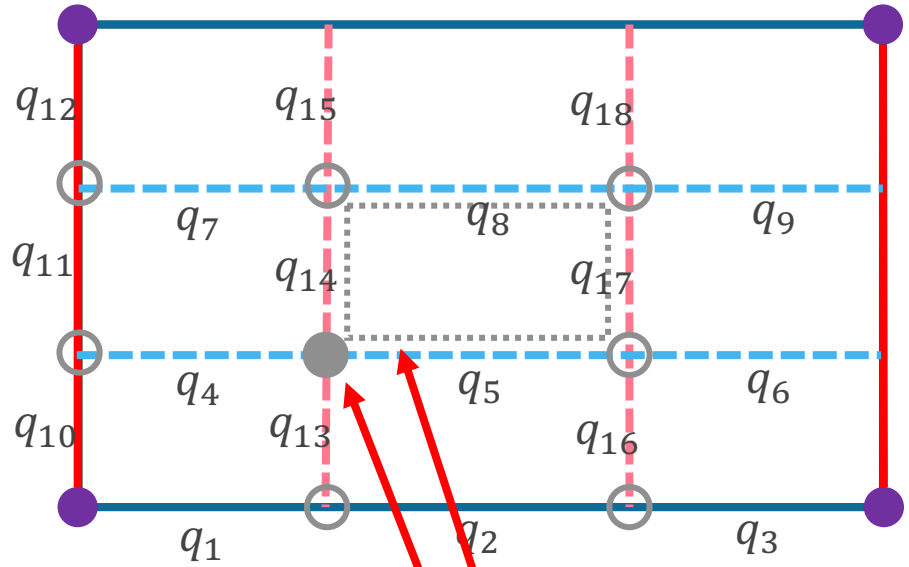
Aside: Toric Code HGP Picture versus Lattice Picture





The Geometric Structure of the Toric Code





The toric code is a stabilizer code defined via the lattice representation of the torus.

- **Edges** define qubits
- **Vertices** define X-stabilizer generators
- **Plaquettes** define Z-stabilizer generators

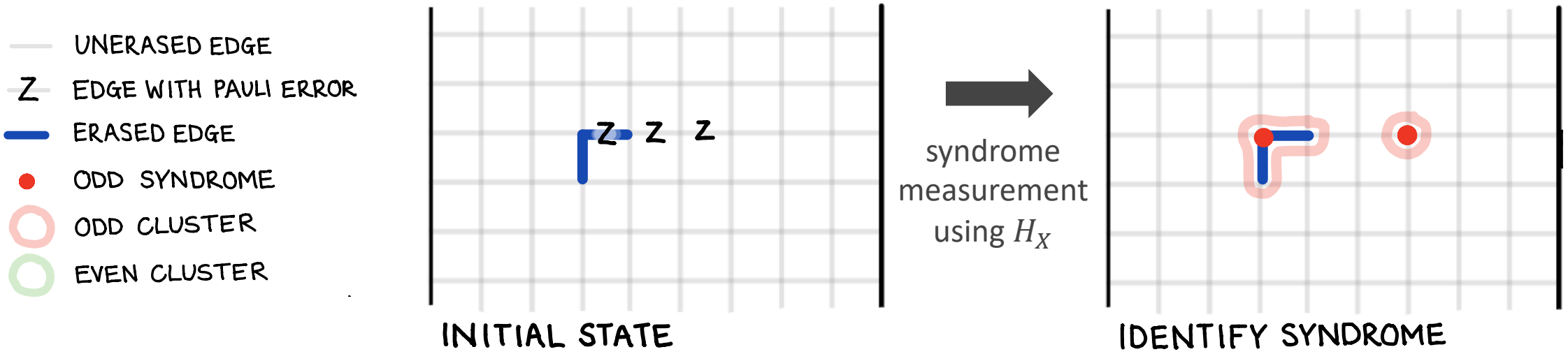
$$g_Z = Z_5 Z_8 Z_{14} Z_{17}$$

$$g_X = X_4 X_5 X_{13} X_{14}$$

Kitaev, A. Y. (2003). Fault-tolerant quantum computation by anyons. *Annals of physics*, *303*(1), 2-30.

Visualizing Errors and Syndromes in the Toric Code

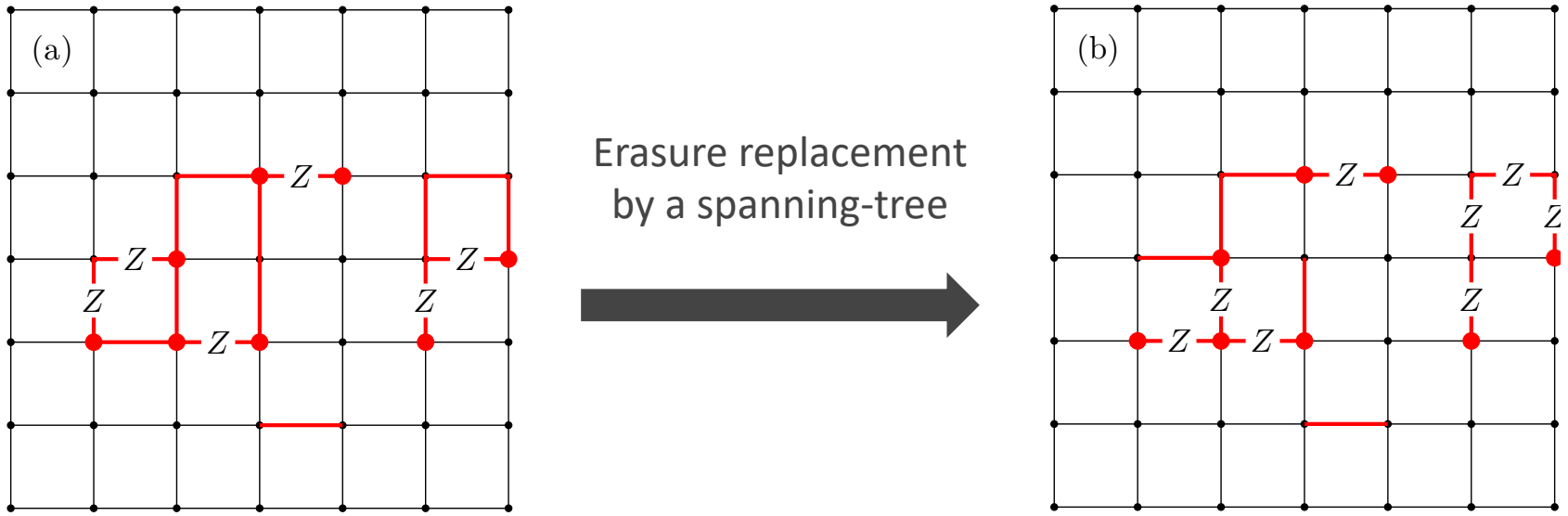
- A mixed error channel includes both erasures and Pauli errors.
- Erased qubits are still assigned random values (erasure pattern is known).
- Z-Pauli errors on qubits (edges) are identified by X-checks (vertices).
- The syndrome consists of triggered-vertices and erased edges.
- Corrections are chains of Paulis connecting pairs of triggered-vertices.



Delfosse, N., & Nickerson, N. H. (2021). Almost-linear time decoding algorithm for topological codes. *Quantum*, *5*, 595.

Surface (Toric) Code Erasure Spanning-Tree

- The erasure-pattern for a surface code can be replaced with a spanning tree.
  - All *vertices* are reached (including the triggered-checks of the syndrome).
- A solution supported on this tree is identified and used to correct.
  - Solutions are valid *up to multiplication by a stabilizer*.

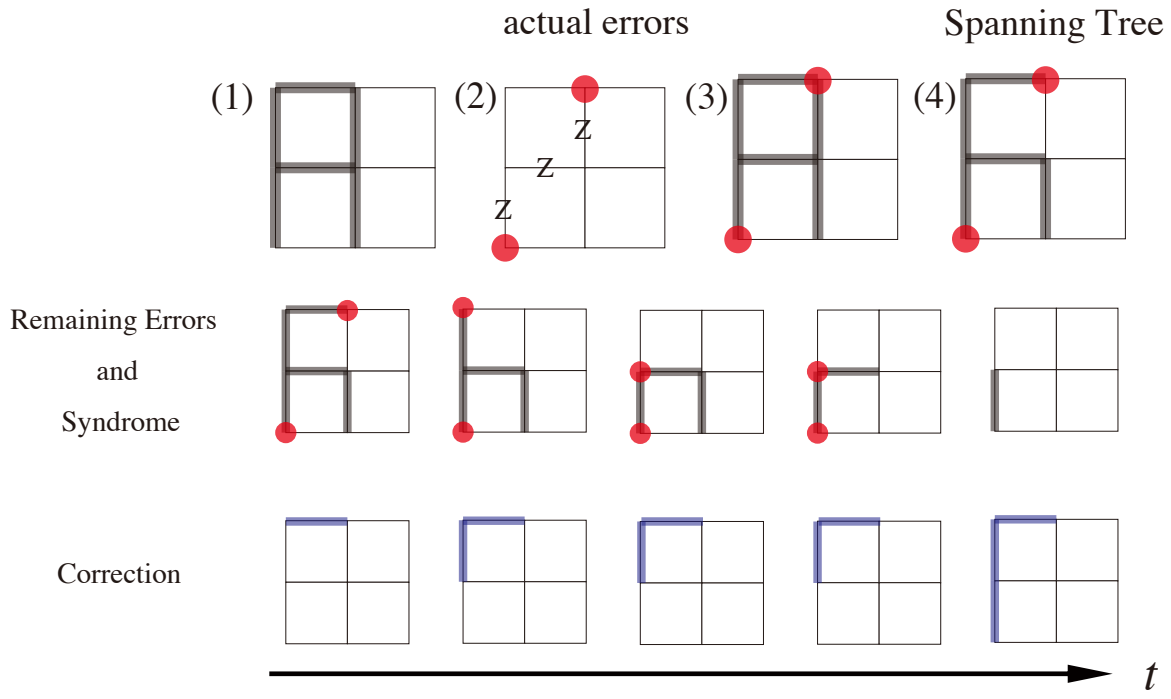


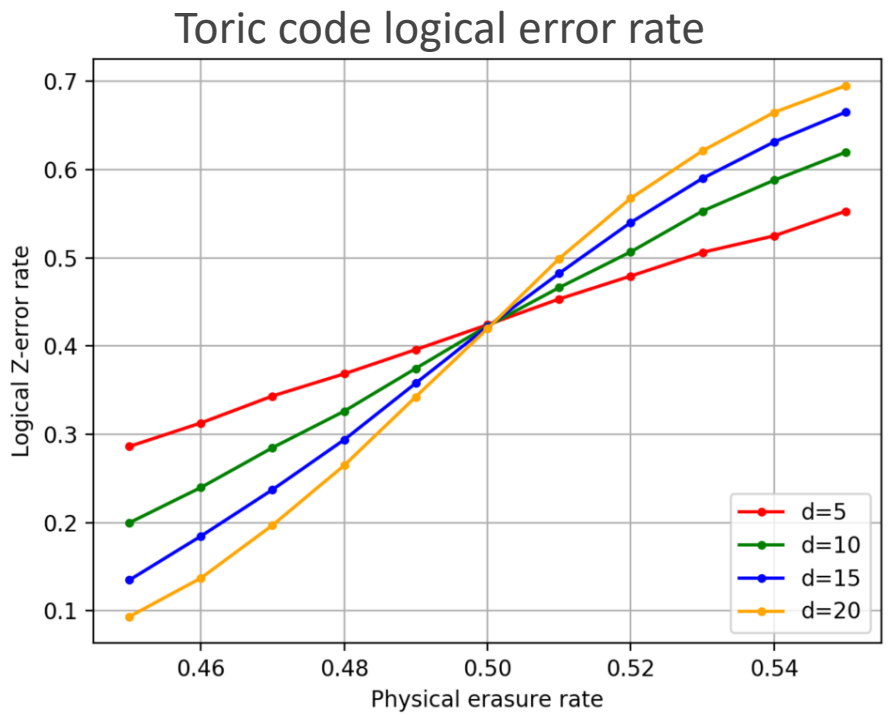
Original erasure-pattern and syndrome

Spanning-tree supported solution

Delfosse, N., & Zémor, G. (2020). Linear-time maximum likelihood decoding of surface codes over the quantum erasure channel. *Physical Review Research*, *2*(3), 033042.

Surface Code Spanning-Tree Peeling Decoder





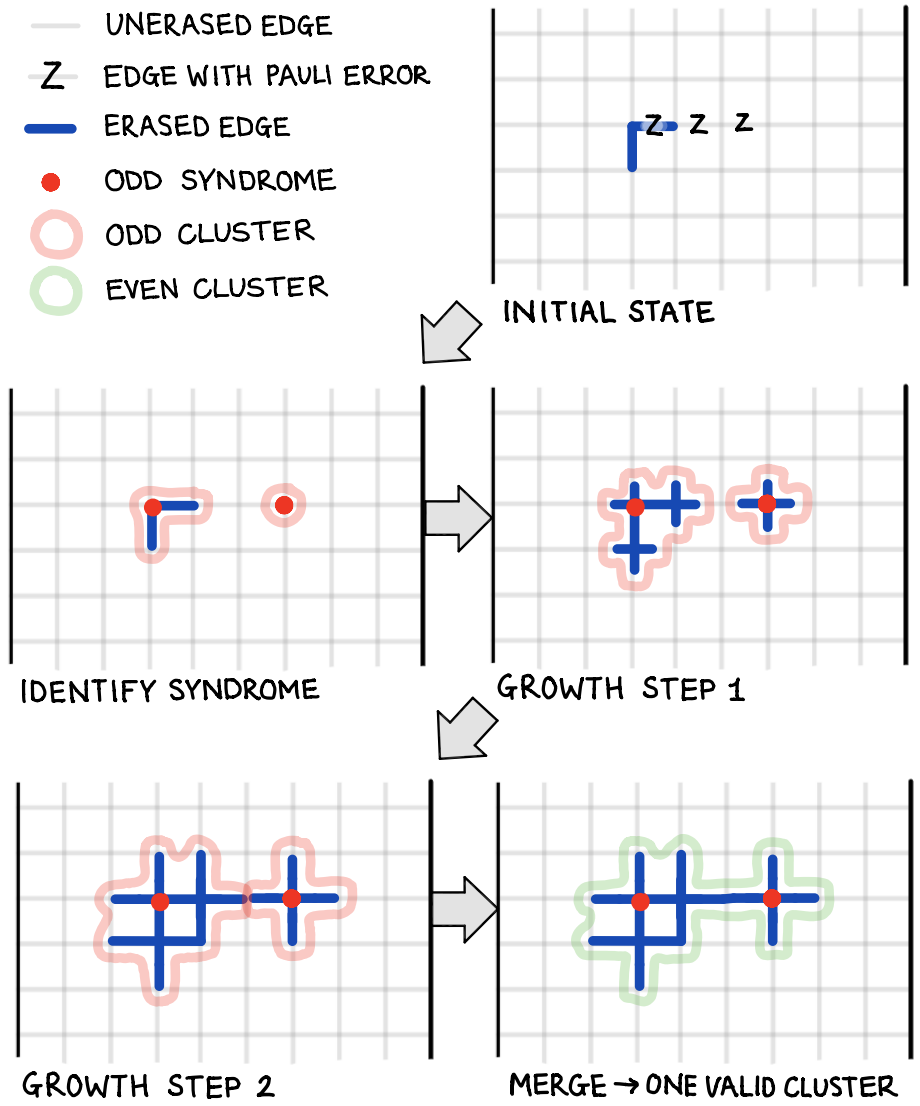
- Using an erasure spanning-tree, erased qubits can be corrected and peeled.
- This process is linear complexity in the code length (*really fast*).
- Furthermore, this is a *maximum likelihood* decoder (*as good as possible*).

(left) Nishio, S., Connolly, N., Piparo, N. L., Munro, W. J., Scruby, T. R., & Nemoto, K. (2025). Multiplexed quantum communication with surface and hypergraph product codes. *Quantum*, *9*, 1613.
(right) Delfosse, N., & Zémor, G. (2020). Linear-time maximum likelihood decoding of surface codes over the quantum erasure channel. *Physical Review Research*, *2*(3), 033042.

Union-Find Decoder for Surface Codes

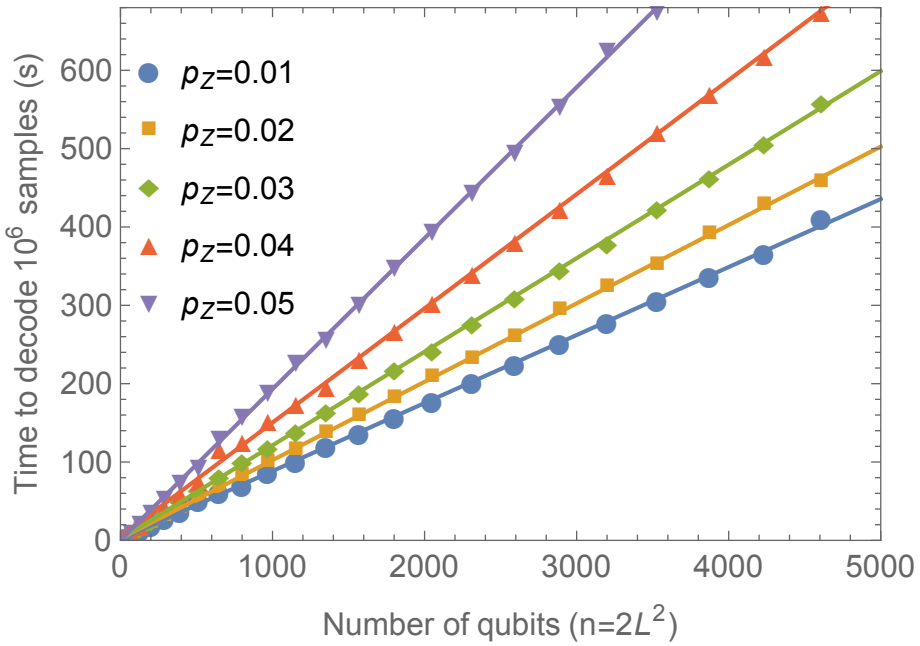
The Union-Find decoder exploits several features of the toric code.

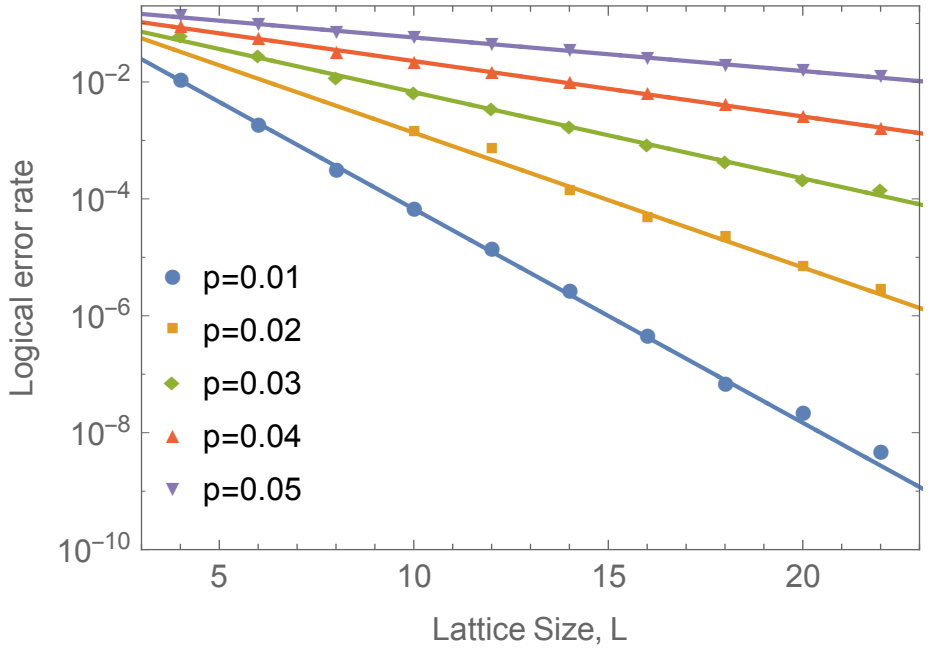
- Chains of errors are likely short (*triggered-vertices should be nearby*).
- Erasures that do not wrap around the torus do not cover logical errors (*therefore these can be corrected*).
- Errors are corrected up to a stabilizer (*so we can freely enlarge the erasure*).
- Erasure clusters containing an even number of triggered-checks support a solution entirely contained in erasure (*clusters can be solved separately*).



Delfosse, N., & Nickerson, N. H. (2021). Almost-linear time decoding algorithm for topological codes. *Quantum*, *5*, 595.

Some Numerical Results for the Union-Find Decoder

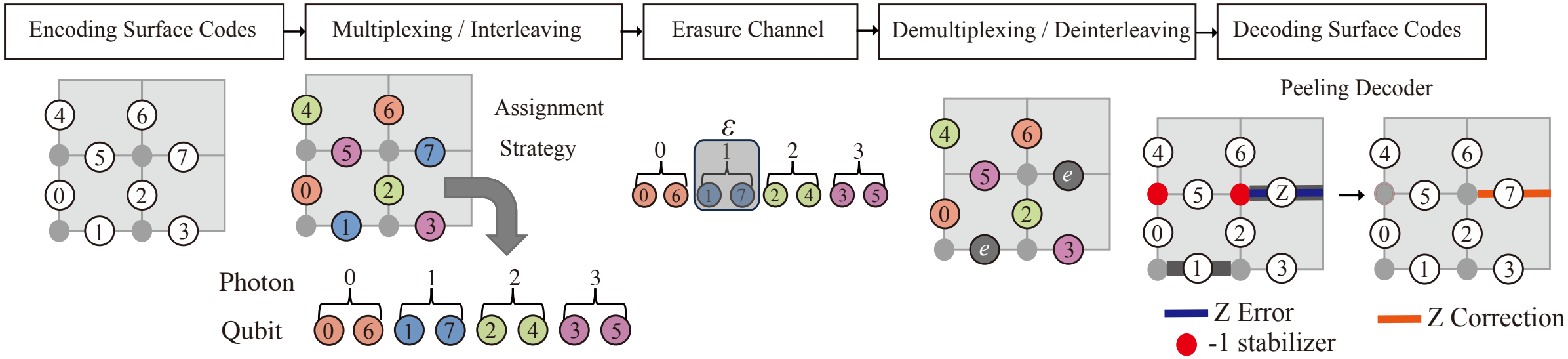




- The union-find decoder is (almost) linear time complexity in the code length.
- The union-find decoder is a close approximation of the maximum likelihood decoder.
- Increasing the torus lattice increases the performance of the decoder.
- Generalized version of the decoder exists for quantum LDPC codes.

Delfosse, N., & Nickerson, N. H. (2021). Almost-linear time decoding algorithm for topological codes. *Quantum*, *5*, 595.

Erasure Decoding Application: Multiplexed Quantum Communication

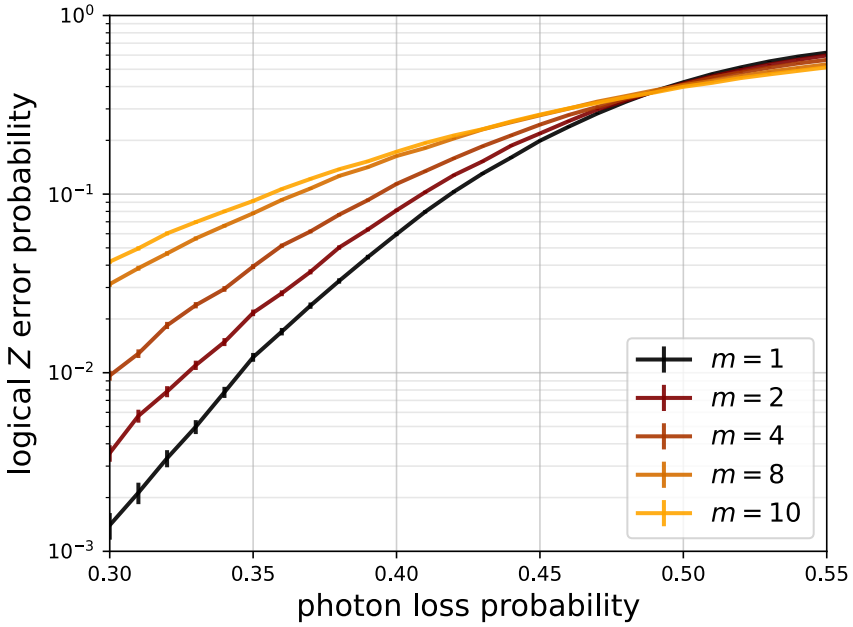


- This refers to a procedure wherein multiple qubits are encoded into a single photon.
- Loss of a single photon yields an erasure error an all encoded qubits.
- Code/decoder aware assignment strategies can mitigate the loss of information.

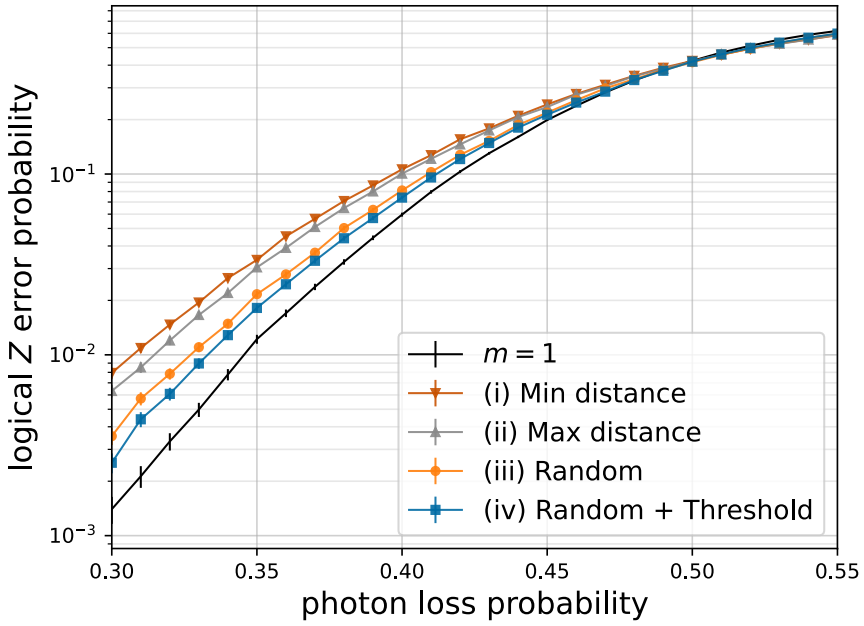
Nishio, S., Connolly, N., Piparo, N. L., Munro, W. J., Scruby, T. R., & Nemoto, K. (2025). Multiplexed quantum communication with surface and hypergraph product codes. *Quantum*, *9*, 1613.

Logical Error Rates for Multiplexed [200,10] Toric Code

Fixed assign. strategy (random), increasing m (qubits per photon)



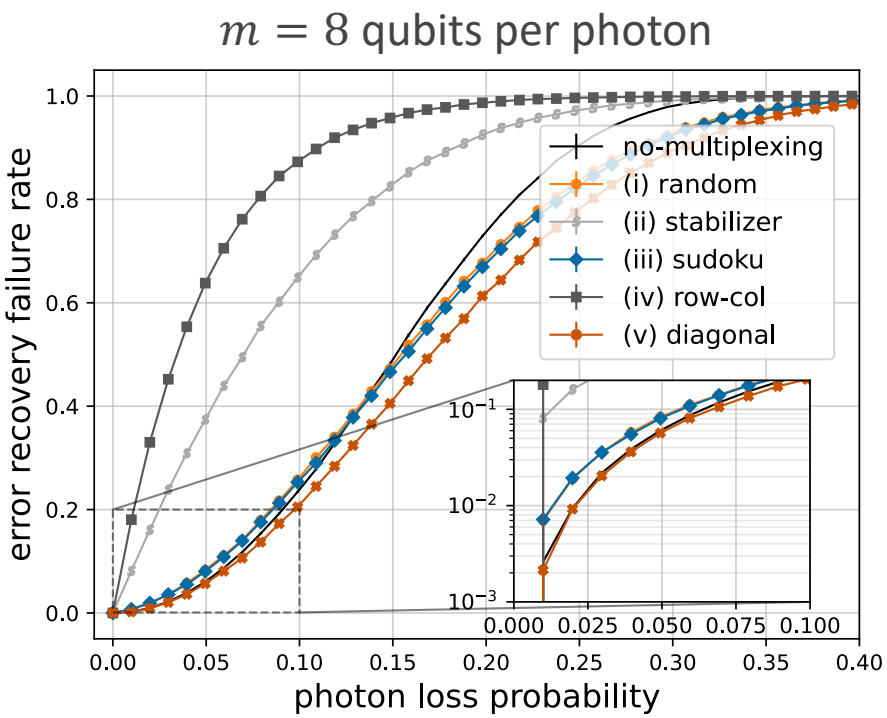
Fixed $m = 2$ qubits per photon, various assignment strategies

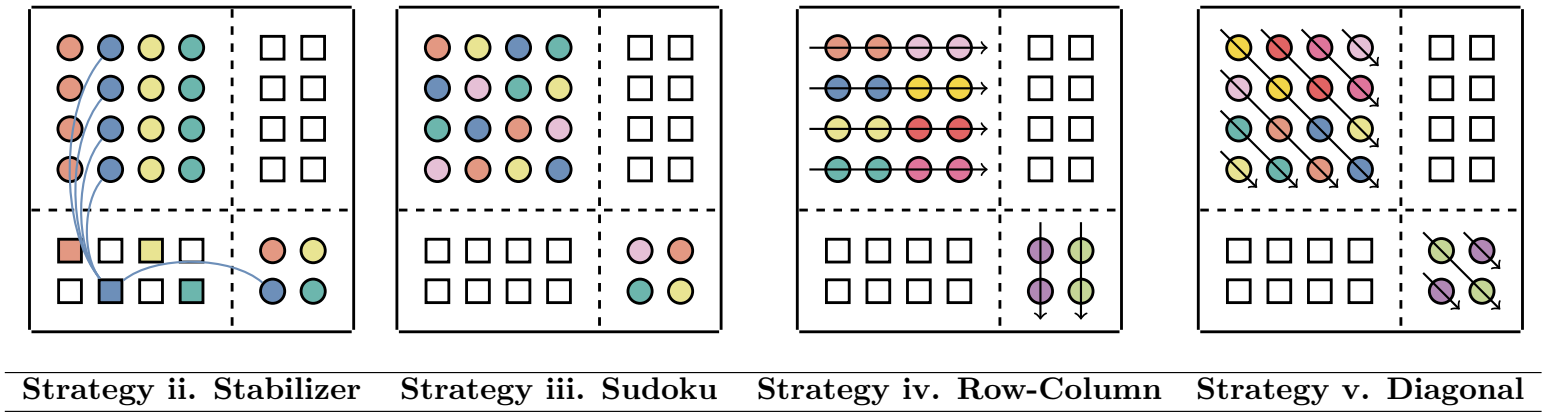


- Plots show erasure decoding (using a surface code spanning-tree) for [200,10] toric code.
- Increasing the number of encoded qubits per photon also increase the logical error rate.
- The choice of assignment strategy has a large effect on the decoding performance.

Nishio, S., Connolly, N., Piparo, N. L., Munro, W. J., Scruby, T. R., & Nemoto, K. (2025). Multiplexed quantum communication with surface and hypergraph product codes. *Quantum*, *9*, 1613.

Logical Error Rates for Multiplexed [320,80] HGP Code



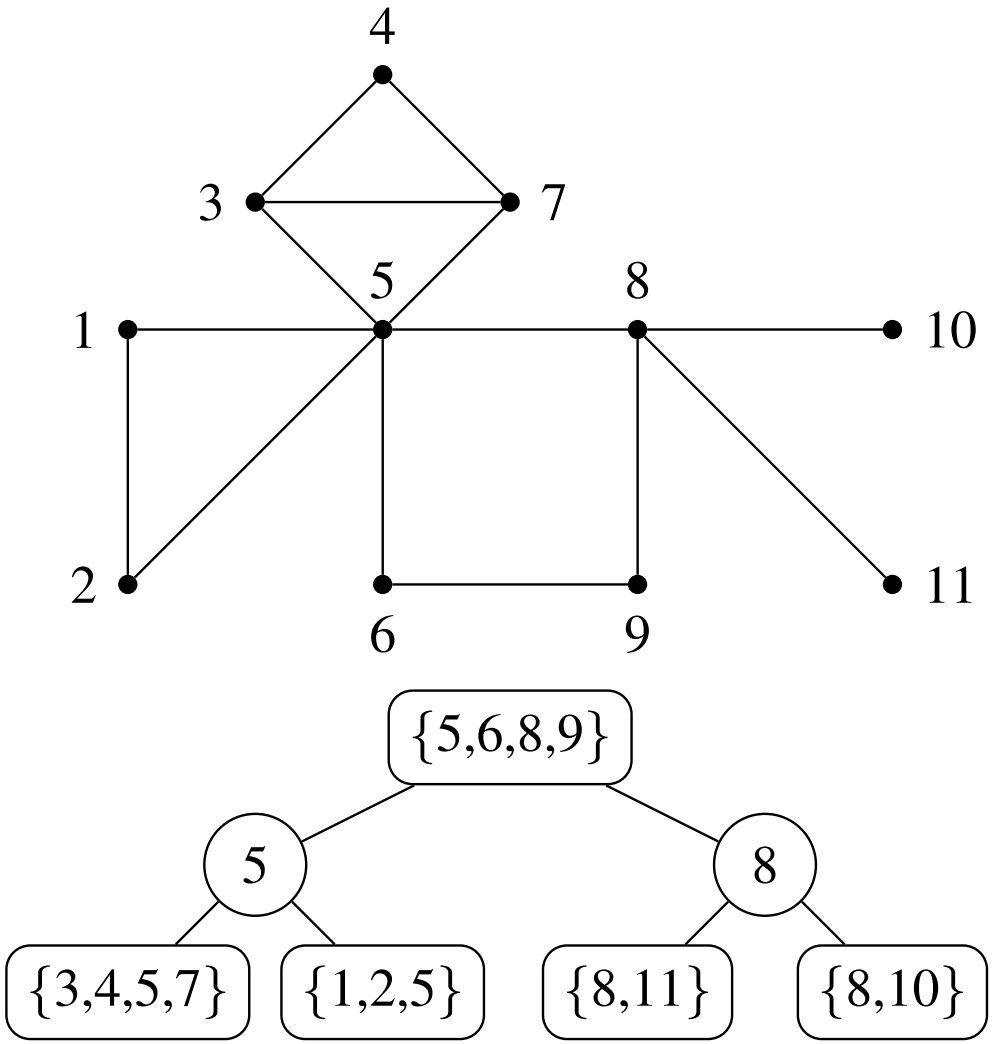


- Plots show erasure decoding using VH decoder for a [320,80] HGP code.
- Decoder-aware assignment strategies have a significant effect on performance.
- Certain strategies adapted to the decoder match the no-multiplexing case!

Nishio, S., Connolly, N., Piparo, N. L., Munro, W. J., Scruby, T. R., & Nemoto, K. (2025). Multiplexed quantum communication with surface and hypergraph product codes. *Quantum*, *9*, 1613.

Generalizing the VH-Decoder: Cluster Decompositions

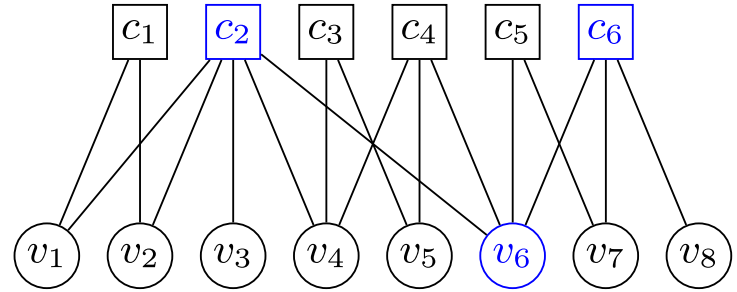
- The VH-Decoder has certain limitations.
  - Only defined for HGP codes.
  - Quadratic complexity in code length.
  - Susceptible to stopping sets.
  - Does not close ML gap at higher erasures.
- The Cluster Decoder uses a similar approach to post process peeling decoder stopping sets.
  - Defined for all quantum LDPC codes.
  - Variable complexity (linear to quadratic).
  - Resolves all* stopping sets.
  - Near ML performance (similar to VH).
- The cluster decoder relies on decomposing the Tanner graph using cluster trees.

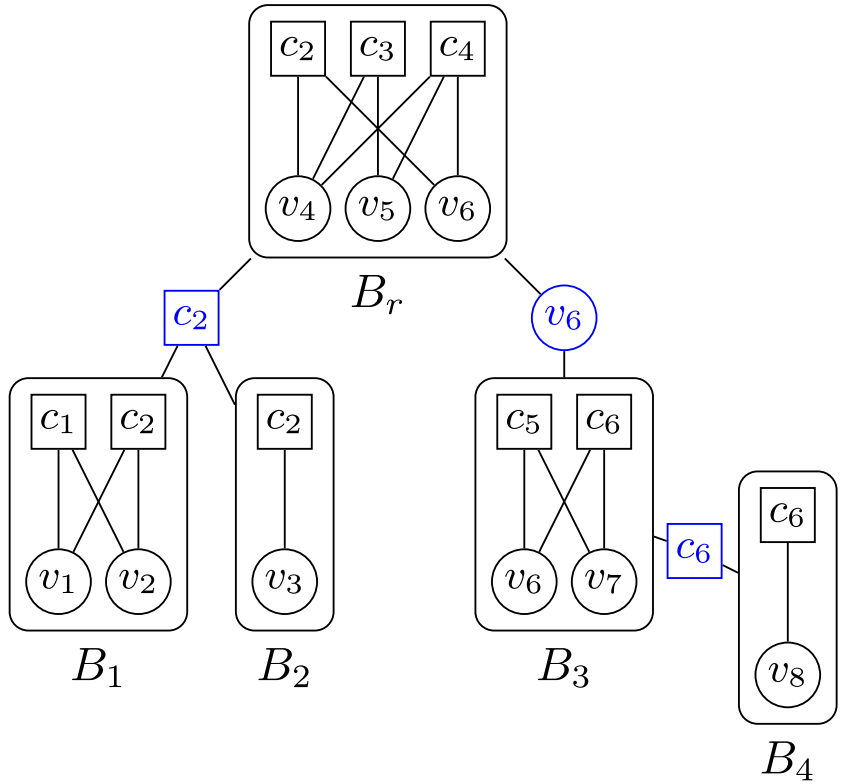


Yao, H., Gökduman, M., & Pfister, H. D. (2025). Cluster decomposition for improved erasure decoding of quantum ldpc codes. *IEEE Journal on Selected Areas in Information Theory*.

Cluster Decoder for Quantum LDPC Codes

1. The cluster decoder applies peeling until stuck.
2. A peeling decoder stopping set gives rise to an erasure-induced subgraph of the Tanner graph.
3. This subgraph is decomposed as a cluster tree.
4. Clusters can be solved via Gaussian elimination.
5. Certain constraints on clusters determine order.





Algorithm 2: RecursiveCompute

Input: Cluster B, residual syndrome s'_x
$L \leftarrow$ list of additional constraints
// Collect constraints, if any
for *v_i in the child variable cut nodes of B* **do**
  for *B' in the child clusters of v_i* **do**
    RecursiveCompute(B', s'_x)
  if *exist a frozen child cluster B'* **then**
    add $v_i(B) = v_i(B')$ to L
for *c_j in the child check cut nodes of B* **do**
  for *B' in the child clusters of c_j* **do**
    RecursiveCompute(B', s'_x)
  if *all child clusters $B_1, \ldots, B_s$ are frozen* **then**
    add $c_j(B) = (s'_x)_j - c_j(B_1) - \ldots - c_j(B_s)$ mod 2 to L
// Compute cluster solutions
if *B is not a root cluster* **then**
  $n \leftarrow$ parent cut node of B
  Compute and store two solutions of B with L, one with $n(B) = 0$, and the other with $n(B) = 1$
  if *both solutions exist* **then**
    B labeled *free* for parent n
  else
    B labeled *frozen* for parent n
else
  Compute one solution of B with L

Algorithm 3: RecursiveSelect

Input: Cluster B with parent cut node n, constraint $n(B) = b$, residual syndrome s'_x
// Select a cluster solution
if *B is frozen or it is the root cluster* **then**
  B has only one solution to select from
else
  B has two stored solutions; select the one consistent with $n(B) = b$
// Propagate constraints
for *v_i in the child variable cut nodes of B* **do**
  for *B' in the child clusters of v_i* **do**
    RecursiveSelect(B', $v_i(B') = v_i(B)$, s'_x)
for *c_j in the child check cut nodes of B* **do**
  $B^*_1, \ldots, B^*_s \leftarrow$ frozen child clusters of c_j
  $B_1, \ldots, B_t \leftarrow$ free child clusters of c_j
  Compute $\Delta s_j = (s'_x)_j - c_j(B) - c_j(B^*_1) - \ldots - c_j(B^*_s)$ mod 2
  // Frozen child clusters
  for *B' in $\{B^*_1, \ldots, B^*_s\}$* **do**
    RecursiveSelect(B', $c_j(B')$ is frozen, s'_x)
  // Free child clusters
  RecursiveSelect(B_1, $c_j(B_1) = \Delta s_j$, s'_x)
  for *B' in $\{B_2, \ldots, B_t\}$* **do**
    RecursiveSelect(B', $c_j(B') = 0$, s'_x)

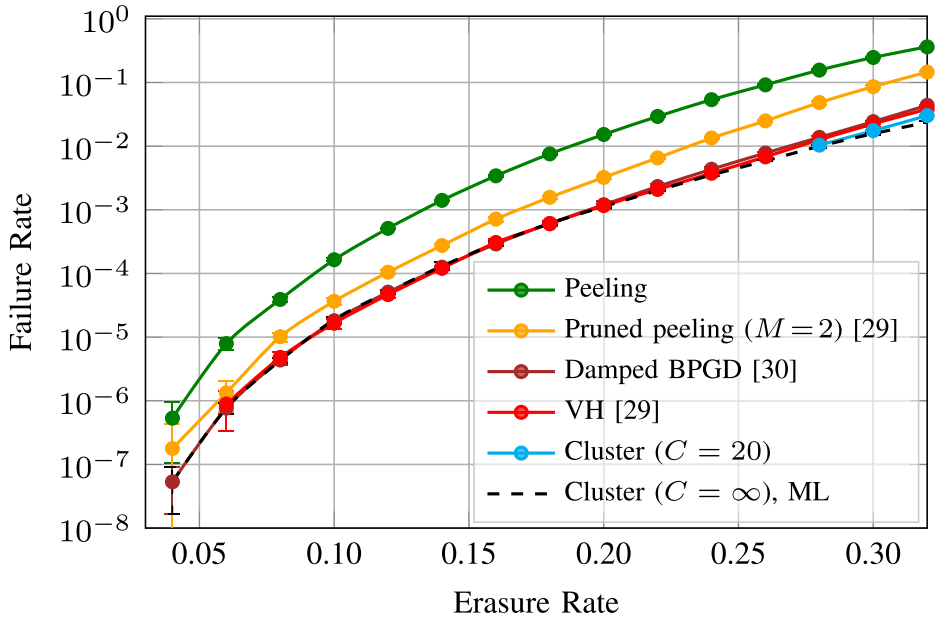
Algorithm 4: Cluster Decomposition Post-Processing

Input: block length n, residual Tanner graph $\mathcal{G}'$, residual syndrome s'_x
Output: $\widehat{x}'$
// Cluster decomposition
Decompose $\mathcal{G}'$ into a cluster forest with Hopcroft-Tarjan's algorithm
// Construct cluster solutions
for *each cluster tree in the forest* **do**
  $B_r \leftarrow$ root cluster
  RecursiveCompute(B_r, s'_x)
  RecursiveSelect(B_r, s'_x)
Merge cluster solutions into a global solution $\widehat{x}'$
return $\widehat{x}'$

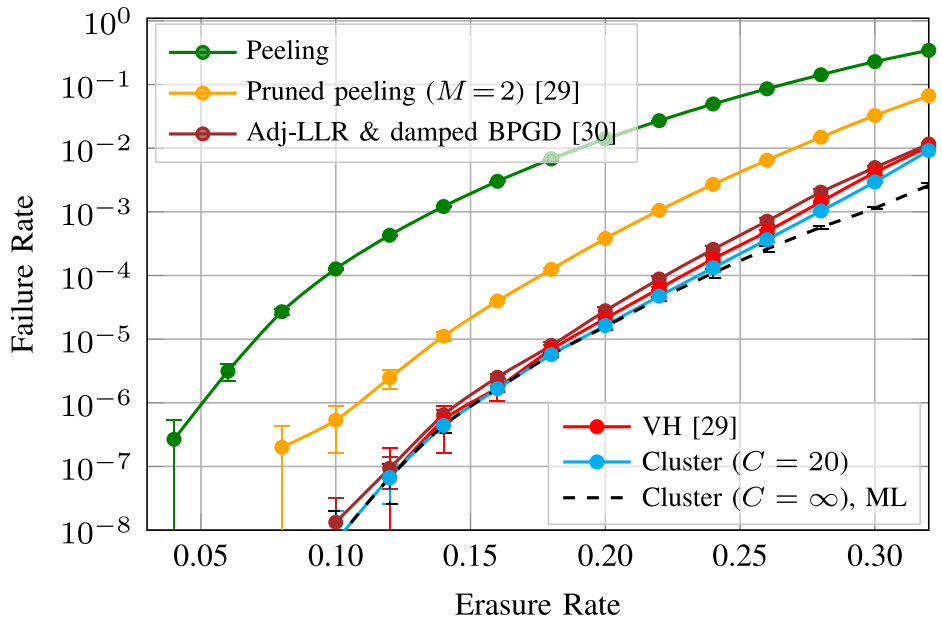
Yao, H., Gökduman, M., & Pfister, H. D. (2025). Cluster decomposition for improved erasure decoding of quantum ldpc codes. *IEEE Journal on Selected Areas in Information Theory*.

Some Numerical Results for the Cluster Decoder

[1600,64] HGP Code

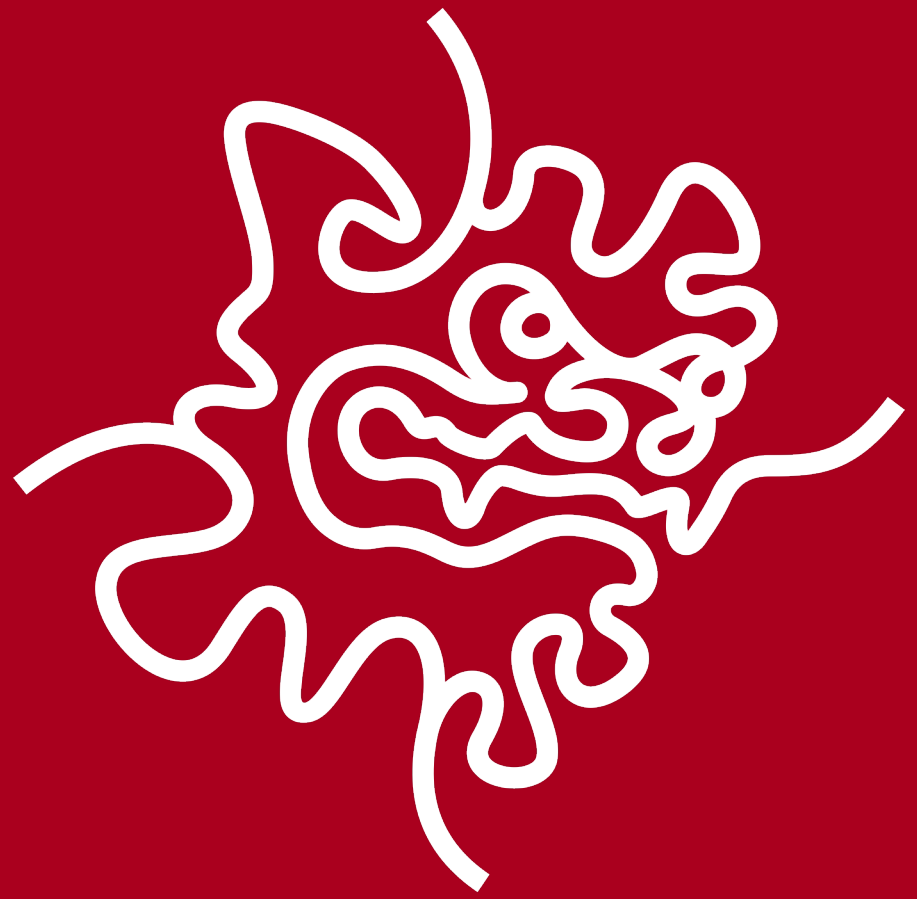


[2025,81] HGP Code



- Plots show a comparison of erasure decoders for two HGP codes.
- Cluster decoder with *constrained* size outperforms VH (linear complexity).
- Cluster decoder with *unconstrained* size achieves ML (quadratic complexity).

Yao, H., Gökduman, M., & Pfister, H. D. (2025). Cluster decomposition for improved erasure decoding of quantum ldpc codes. *IEEE Journal on Selected Areas in Information Theory*.



Thank you!