An Efficient Erasure Decoder and Quantum Multiplexing Using Hypergraph Product Codes

Nicholas Connolly[1,2], **Shin Nishio**[3,1,4], **Vivien Londe**[5], **Nicolò Lo Piparo**[1], **William John Munro**[1,4], **Thomas Rowan Scruby**[1], **Anthony Leverrier**[2], **Nicolas Delfosse**[6,7], **and Kae Nemoto**[1,4]

[1]OIST, [2] Inria Paris, [3]SOKENDAI, [4]NII, [5]Microsoft France, [6]IonQ, [7]Microsoft Quantum

arXiv:2208.01002, arXiv:2406.08832

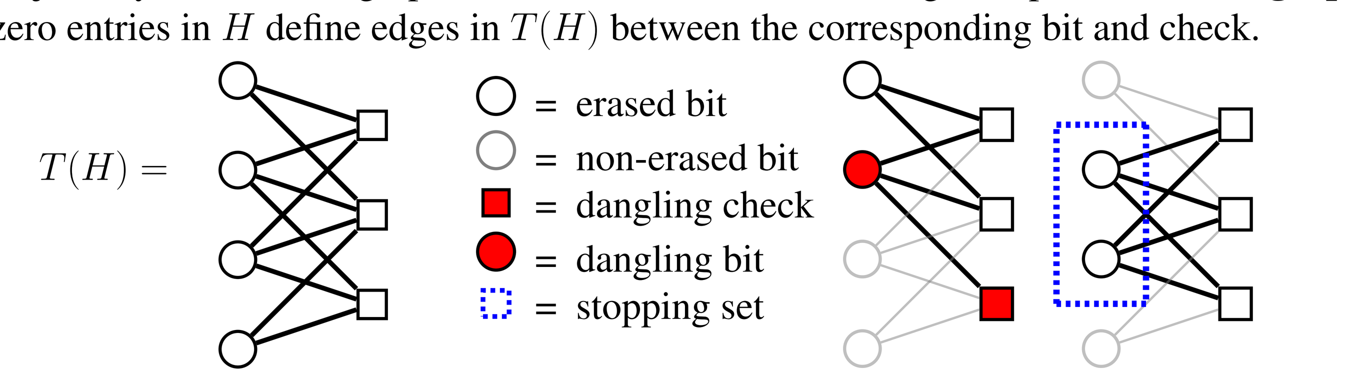
Abstract

We consider the erasure decoding problem for HGP codes, a popular family of quantum LDPC codes. We introduce an erasure decoder for HGP codes, based on the classical peeling decoder, that closely approximates the ML decoder, but with reduced computational complexity. We also illustrate a practical application of this decoder with quantum multiplexing, showing how physical resources can be saved without sacrificing performance.

Classical Codes and the Erasure Channel

A classical binary linear **code** C is a vector space over $\mathbb{Z}_2$, where **codewords** are vectors $x \in C \subseteq \mathbb{Z}_2^n$. A code can also be defined as the kernel of a **parity check matrix** H. Error correction is modeled as a linear algebra problem using H, where the columns of H correspond to **bits** in C and the rows in H correspond to **checks** in C. Using H as the adjacency matrix for a graph, a code can be visualized using its bipartite **Tanner graph** $T(H)$, where the nonzero entries in H define edges in $T(H)$ between the corresponding bit and check.

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



(a) Parity check matrix H and Tanner graph $T(H)$ **(b)** Examples of erasure induced subgraphs of $T(H)$

Figure 1: Visualization of a classical code via its Tanner graph and examples of erasure induced subgraphs.

An **erasure error** refers to the loss of a *known* subset of bits. The set of erased bits $\mathcal{E}$ induces an **erasure subgraph** in the Tanner graph $T(H)$. Erasure correction can be converted into error correction by assigning the erased bits in $\mathcal{E}$ values at random, and assuming no errors in $\mathcal{E}^C$. The **Peeling Decoder** is a linear-complexity algorithm for correcting erasure errors. The algorithm can fail, but it works well for classical LDPC codes.

Classical Peeling Decoder Algorithm

1. Given erasure pattern $\mathcal{E}$, a **dangling** (degree 1) check in the erasure induced subgraph of $T(H)$ is selected.
2. The adjacent erased bit is corrected and then removed from $\mathcal{E}$, shrinking the erasure and the subgraph.
3. The algorithm terminates when $\mathcal{E} = \emptyset$, or fails when $\mathcal{E}$ is a **stopping set** (contains no dangling checks).

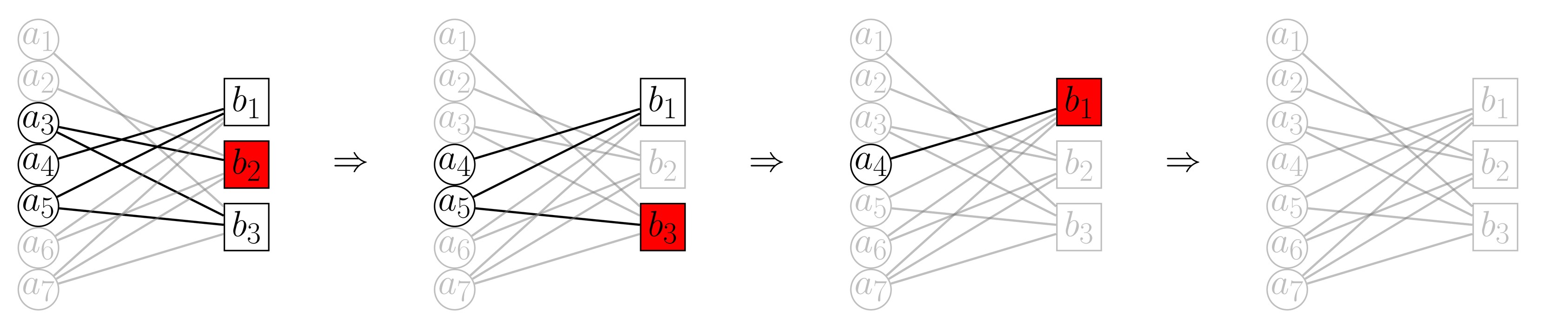


Figure 2: Example of a sequence of stages in the peeling decoder; each "peel" reduces the size of the erasure $\mathcal{E}$ until empty.

Families of Quantum Codes

A **quantum code** of length N and dimension K is a subspace of a Hilbert space; vectors are N-qubit states $|\psi\rangle \in \mathbb{C}^N$. Errors on a state $|\psi\rangle$ are described discretely by N-qubit Pauli operators in $P_N = \{I, X, Z, Y\}^{\otimes N}$.

- **Stabilizer codes** are defined as the space of states left fixed by some subgroup of the Pauli-group P_N.
- **CSS codes** are a subclass of stabilizer codes defined by *commuting* N-qubit X- and Z-Pauli operators.
  - X- and Z-Pauli **stabilizer generators** define the rows of matrices H_X and H_Z (where $H_X H_Z^T = 0$).
  - CSS Z and X error correction is modeled using the classical codes $C_X = \ker(H_X)$ and $C_Z = \ker(H_Z)$.
- **Surface codes** are a subclass of CSS codes defined from a cellulation of a surface.

Hypergraph Product Codes (Tillich-Zémor)

The **hypergraph product** (HGP) code of two classical codes $C_1 = \ker(H_1)$ and $C_2 = \ker(H_2)$ is the quantum code $C = \mathrm{CSS}(C_X, C_Z)$, 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 by the following formulas.

$$H_X = (H_1 \otimes I | I \otimes H_2^T) \qquad \text{and} \qquad H_Z = (I \otimes H_2 | H_1^T \otimes I)$$

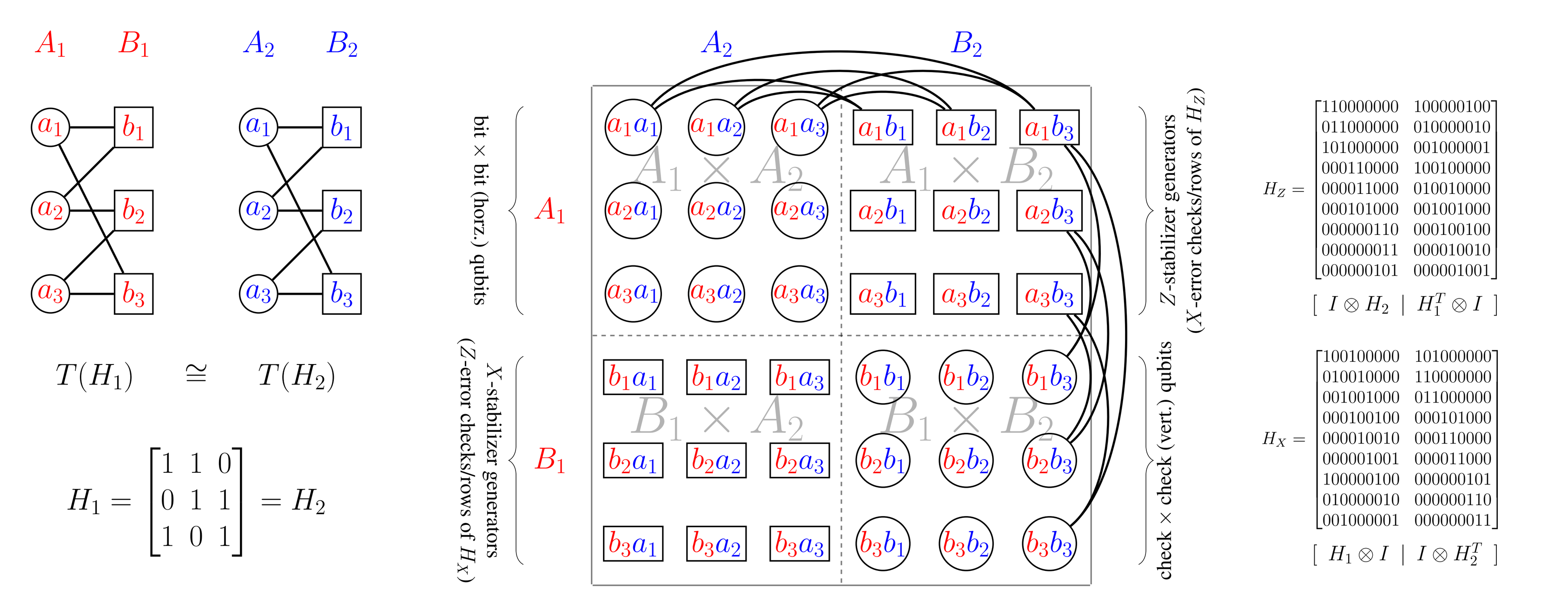


Figure 3: Geometric structure of the Tanner graph of a HGP of two 3-bit repetition codes (equivalent to the 3×3 toric code).

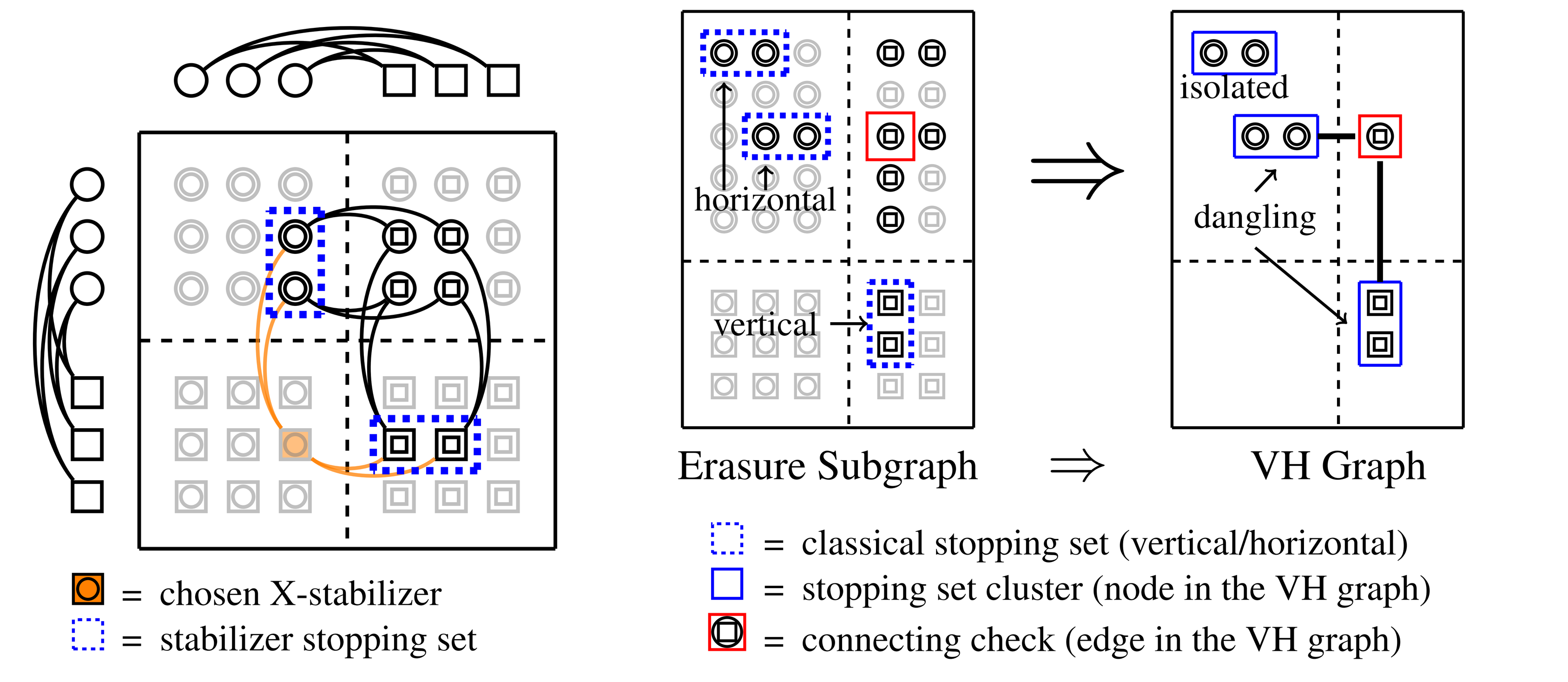
By mapping Pauli errors $X_i, Z_i \in P_N$ onto binary strings $e_i \in \mathbb{Z}_2^N$, the erasure decoding problem for a CSS code can be modeled as a classical erasure problem using H_Z or H_X, and the peeling decoder directly applied.

Generalized Peeling Decoder for Quantum Codes

Peeling works poorly for quantum codes due to the existence of *quantum-specific stopping sets*. By addressing these, we propose two modified versions of the classical peeling algorithm generalized to the quantum case.

Stabilizer Stopping Sets and the Pruned Peeling Decoder for CSS Codes

1. Given erasure pattern $\mathcal{E}$, apply the peeling decoder algorithm until $\mathcal{E}$ contains no remaining dangling checks.
2. If $\mathcal{E}$ contains the qubit-support S of an X-type stabilizer, then $S \subseteq \mathcal{E}$ is a **stabilizer stopping set** of $T(H_Z)$.
3. Remove a qubit in S from $\mathcal{E}$ and continue peeling (errors are corrected *up to multiplication by a stabilizer*).



(a) A stabilizer stopping set in a HGP code **(b)** Classical stopping sets in a HGP code and corresponding VH graph

Figure 4: Examples of HGP stabilizer and classical stopping sets corrected by the Pruned Peeling and VH decoders.

Classical Stopping Sets and the Vertical-Horizontal (VH) Decoder for HGP Codes

1. Apply the pruned peeling decoder algorithm until $\mathcal{E}$ contains no dangling checks and no erased stabilizers.
2. $\mathcal{E}$ contains a **classical stopping set** if it contains a stopping set for $T(H_1)$ (vertical) or $T(H_2)$ (horizontal).
3. Apply the Gaussian (ML) decoder on classical stopping set clusters *in sequence*, combining local solutions.
4. The VH decoder will terminate provided the **VH-graph** contains no cycles (the graph can be "peeled").

Numerical Performance and Computational Complexity of the Proposed Decoders

Numerical results show that the Pruned Peeling + VH decoder performs almost as well as the maximum likelihood decoder at low erasure rate, but in $O(N^2)$ complexity, where N is the length of the HGP code.

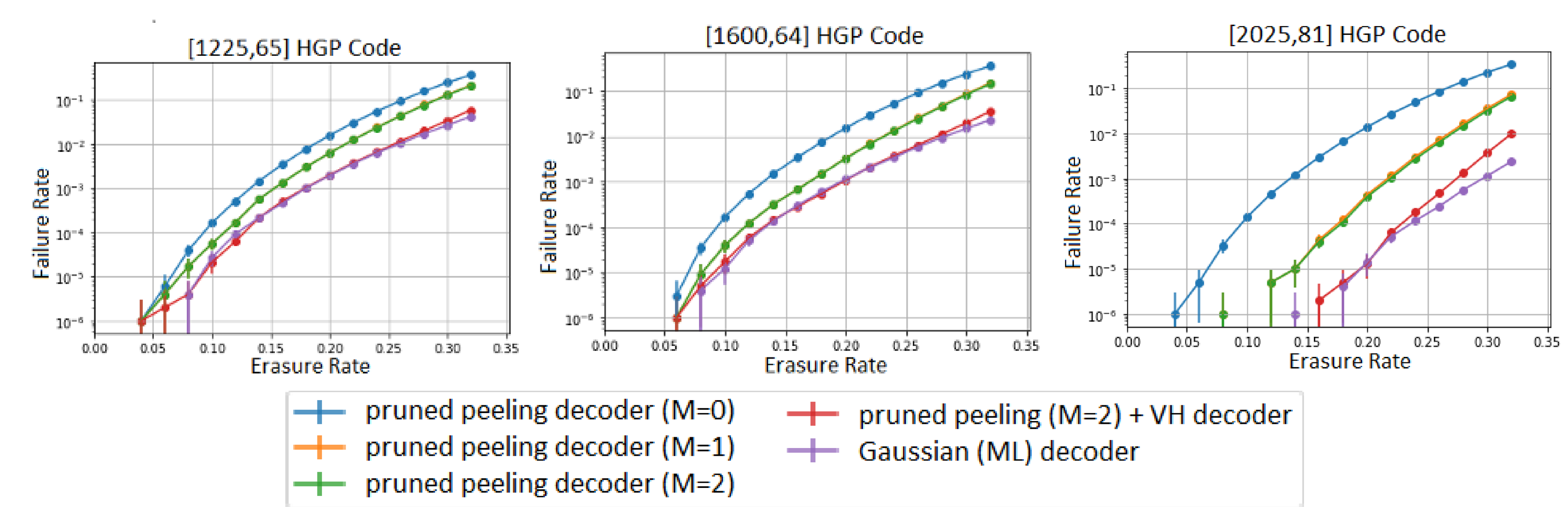


Figure 5: Comparison of performance for Pruned Peeling, VH, and ML decoders for three randomly generated HGP codes.

Pruned Peeling + VH Decoder Application: Quantum Multiplexing

Multiplexed quantum communication refers to a photonic system wherein each photon encodes m-qubits of information. The loss of a single photon corresponds to an erasure error on all encoded qubits, and so errors are correlated based on how qubits are assigned to photons. We show how the the performance cost of multiplexing with HGP codes can be offset by adapting the assignment strategy to the decoder.

Examples of HGP Assignment Strategies

i. **Random:** qubits assigned to photons at random.
ii. **Stabilizer:** photons correspond to the qubit-support of X and Z-type stabilizer generators.
iii. **Sudoku:** qubits of a given photon come from different rows or columns in the Tanner graph.
iv. **Row-Column:** qubits of a given photon come from the same row or column in the Tanner graph.
v. **Diagonal:** qubits of a given photon come from the same diagonal slice in the Tanner graph.

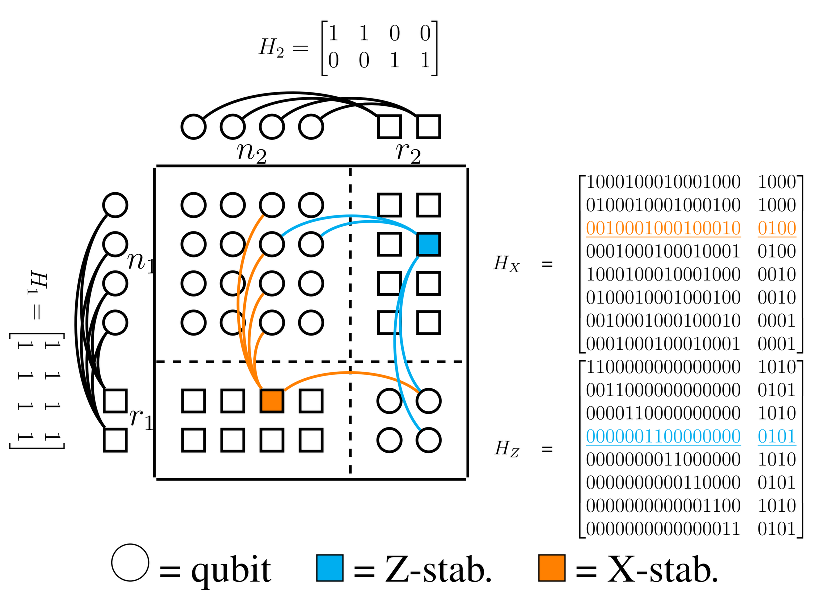


Figure 6: Example of a HGP code Tanner graph

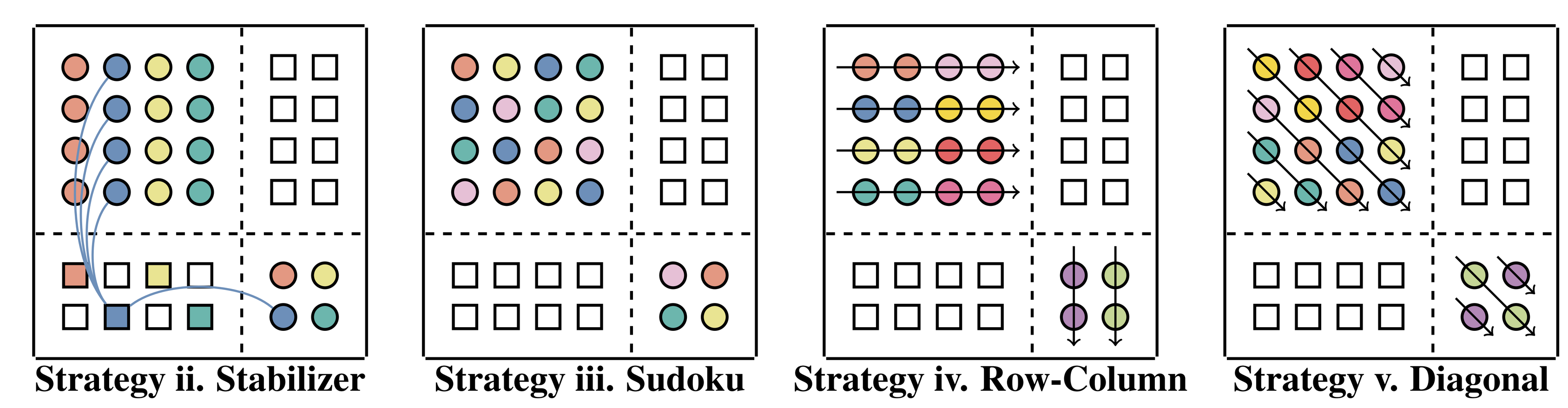
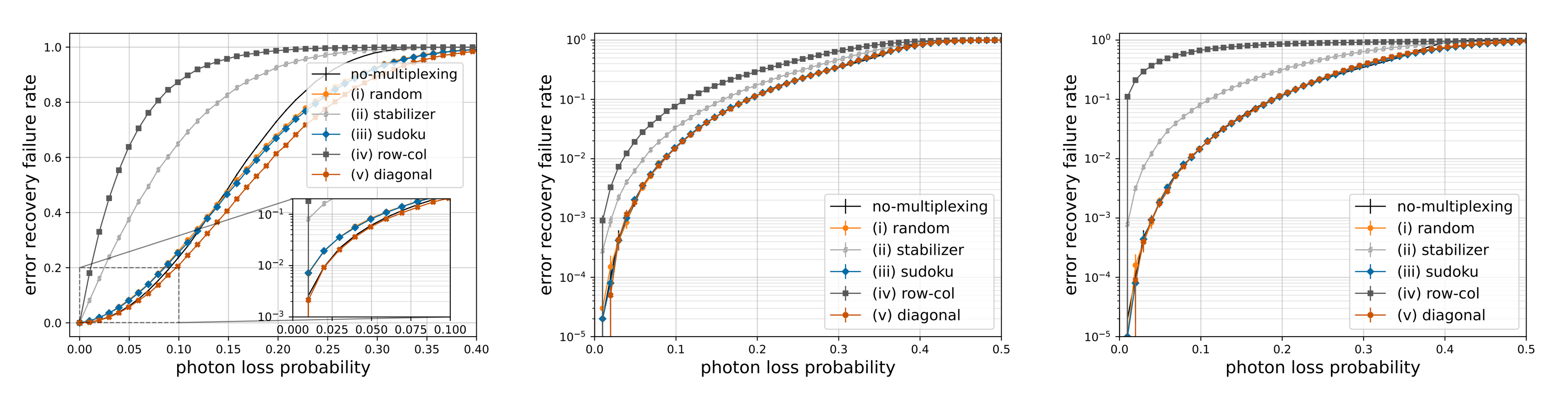


Figure 7: Visualization of some multiplexing HGP assignment strategies. Qubit color denotes the assigned photon.

Numerical Performance of Multiplexed HGP Codes

Although multiplexing usually increases the logical error rate, our simulations show that decoder-aware assignment strategies can mitigate this or even improve performance while requiring fewer physical resources.



(a) [[320, 82]] HGP code with $m = 8$ **(b)** [[512, 8]] HGP code with $m = 8$ **(c)** [[512, 8]] HGP code with $m = 16$

Figure 8: Multiplexing decoder performance of each strategy for some HGP codes with different multiplexing values m.