CONSTRUCTION OF QUANTUM GRAPH CODES: A LIGHTS OUT APPROACH

A THESIS

submitted in partial fulfillment of the requirements

for the award of the dual degree of

Bachelor of Science - Master of Science

in

ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

by

ROHITH K

(21226)

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
INDIAN INSTITUTE OF SCIENCE EDUCATION AND RESEARCH BHOPAL
BHOPAL - 462066

April 2026

भारतीय विज्ञान शिक्षा एवं अनुसंधान संस्थान भोपाल

Indian Institute of Science Education and Research Bhopal

Department of Electrical Engineering and Computer Science

Bhopal By-pass Road, Bhauri, Bhopal 462066

CERTIFICATE

This is to certify that **Rohith K**, BS-MS (Dual Degree) student in the Department of Electrical Engineering and Computer Science, has completed bona fide work on the thesis entitled **'Construction of Quantum Graph Codes: A Lights Out Approach'** under my supervision and guidance.

April 2026	**Dr. Ankur Raina**	**Dr. Prafullkumar Tale**
IISER Bhopal	(IISER Bhopal)	(IISER Pune)

Committee Member	**Signature**	**Date**
Dr. Ankur Raina	__________________	__________
Dr. Sujit PB	__________________	__________
Dr. Santanu Talukder	__________________	__________
Dr. Priyadarshi Mukherjee	__________________	__________
Dr. Arijit Sen	__________________	__________

ACADEMIC INTEGRITY AND COPYRIGHT DISCLAIMER

I hereby declare that this project report is my own work and due acknowledgement has been made wherever the work described is based on the findings of other investigators. This report has not been accepted for the award of any other degree or diploma at IISER Bhopal or any other educational institution. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission.

I certify that all copyrighted material incorporated into this document is in compliance with the Indian Copyright (Amendment) Act (2012) and that I have received written permission from the copyright owners for my use of their work, which is beyond the scope of the law. I agree to indemnify and safegaurd IISER Bhopal from any claims that may be asserted or that may arise from any copyright violation.

April 2026 **Rohith K**

IISER Bhopal

ACKNOWLEDGEMENT

I would like to express my utmost gratitude to my thesis advisors, **Dr. Ankur Raina** and **Dr. Prafullkumar Tale**, for their guidance and encouragement, without which this work would not have been possible. My mentor at QuCIS, **Harsh** bhaiyya, for his constant support and enthusiasm, **Mainak** bhaiyya, and my **labmates** whose outstanding works have been an inspiration for my own.

My friends—**Rishi**, **Rose**, **Anjali**, **Nethania**, **Nevil**, and **Abu**—the most special group of people who were lucky enough to *have me share part of their lives*. **Henna**, for all the rants that you endured, especially during thesis work. **Mitesh**, my roommate, for the free rolls you'd leave me at night. Nothing of me is original, I am a mosaic of everyone I've ever known, and all of them have contributed in ways they might not even know, to this work and to my life.

I was privileged to have a warm and nurturing environment growing up. I have my family—**Achan**, **Amma**, and **Sree**—to thank for this. Thank you for listening with complete patience and telling me exactly what I need to hear to keep moving forward.

Rohith K

ABSTRACT

Quantum computers and algorithms have demonstrated an advantage in certain problems over their classical counterparts. There exist problems that turn out to be incredibly useful, in which these computers perform exponentially better. However, such systems are highly prone to error, which makes practical implementation extremely difficult. With the advent of quantum information science, the mitigation of errors forms the core motivation behind quantum error correcting codes. In the classical realm of computation, error correction is a well-established domain, and taking inspiration from classical error correction, in 1995, Peter W. Shor introduced the first [[9,1,3]] repetition code which could detect up to 2-qubit quantum errors.[15] In 1997, Daniel Gottesman, in his doctoral thesis, introduced stabilizer codes, which formalized a group theoretic structure to construct and analyze code.[3] Since then, quantum error correction has been well formulated and has been shown to be successful in overcoming errors to an extent.

In classical error correction, special familes of graphs have shown to produce good error correcting codes.[16,2,8,9] The concept of graph states, attributed to Raussendorf and Briegel,[11] is an important part of measurement based (or one-way) quantum computing and heavily related to stabilizer codes. While finding the minimum distance of an error correcting code from an arbitrary graph state has been shown to be difficult,[5] there have been recent advancements in establishing relations between the graph state and minimum distance.[6] The main objective of this work is to find such relations between graph states and the minimum distance of the error correcting codes derived from the graphs through measurement. With these relations, we attempt to construct families of graphs that produce codes of desired parameters.

Keywords: Quantum Error Correction, Graphs

LIST OF SYMBOLS

QLO	Quantum Lights Out
$QECC$	Quantum Error Correcting Code
CSS Codes	Calderbank-Shor-Steane Codes
$\mathcal{S}$	Stabilizer group
$\mathcal{S}_G$	Generators of stabilizer group $\mathcal{S}$
$\mathcal{N}_\mathcal{G}$	Normalizer
$\mathcal{C}$	Code
$\mathcal{C}^\perp$	Dual of $\mathcal{C}$
n	Block length of code
k	Dimension of code
d	Minimum distance of code
G	Generator matrix of classical code
P, X, Z, I	Pauli Operators
Γ	Undirected Graph
A_Γ	Adjacency matrix of graph Γ
A_o	Adjacency matrix of output subgraph
A_I	Biadjacency matrix between input and output vertices
$\mathcal{I}, \mathcal{P}, \mathcal{O}$	Set of input, pivot, and non-pivot output vertices
$\mathcal{B}$	Output vertices that are adjacent to some input vertices
$\mathcal{A}$	Output vertices that are not adjacent to any input vertices
$N(u)$	Set of vertices that are adjacent to some vertex u
$\triangle N(S)$	Set of vertices that have odd number of edges to vertices in S
$\mathbb{F}_2$	A binary field
$wt()$	Hamming weight
l	Configuration vector
f	Flip vector
$\langle . \rangle$	Euclidean dot product over $\mathbb{Z}$ applied to binary vectors
$E_\lambda(A)$	The λ-eigenspace of A
$\ker(A)$	The nullspace of A

LIST OF FIGURES

Contents

1

Introduction

1.1 Outline and Contributions

Quantum error correction is a well-established field with strong foundations. The first chapter introduces the necessary background in quantum error correction, with particular emphasis on the stabilizer formalism. A brief description of encoding using measurement-based quantum computing is also provided.

The second chapter establishes the connection between graph states and stabilizer codes through recent developments in the area. This chapter also introduces the *Quantum Lights Out (QLO)* game, which serves as a useful tool for analyzing graph states and quantum codes.[6]

The main contributions of this thesis are presented in the results chapter. First, we analyze the *QLO* framework to identify structural properties of graphs that influence the minimum distance, leading to several lower bounds. We then reformulate *QLO* in an algebraic framework to derive additional relations and bounds. Finally, using the insights obtained from these results, we construct families of graphs that yield quantum codes with desired parameters. We present three constructions, each providing a refinement over the previous one.

In the final chapter, we summarise our findings, quickly glancing through the methods we adopted, followed by a brief discussion on promising future directions.

1.2 Error Correction

Quantum systems are inherently prone to noise. It is very hard to isolate a state and prevent external interactions. Naturally, a very damaging consequence is the difficulty in storing and transferring information using qubits, without the occurrence of any unwanted measurements or interactions. It is also quite difficult to perform unitary transformations, which act as gates. Classically, we deal with errors using error detection and error correction. Error detection involves checking if any error has occurred and, if yes, resending the problematic bits. However, the no-cloning theorem implies that an arbitrary quantum state cannot be copied, which prevents resending. Thus, classically inspired error correcting codes are employed to overcome errors in quantum systems.

Quantum error correction theory was formally developed by Emanuel Knill and Raymond Laflamme in 1996.[7] For a general quantum code, the aim is to protect k-logical qubits from unwanted errors. The k logical qubits span a Hilbert space of 2^k basis states. We can encode the logical state into some state of an n qubit system, where $n > k$. The resultant codespace $\mathcal{C}$ is a subspace of the full Hilbert space with 2^n basis states. Thus, the encoding procedure is simply a mapping from the logical 2^k-dimensional Hilbert space to some subspace of the 2^n dimensional Hilbert space. Classically, the minimum distance d is defined as the minimum distance between any two distinct codewords. For a linear code, this is equal to the weight of the smallest non-zero codeword. With minimum distance d, a code can detect up to $d-1$ errors and correct up to $\lfloor \frac{d-1}{2} \rfloor$ errors. We represent such a quantum code as an $[[n, k, d]]$-Quantum code.

1.2.1 Repetition Codes: Bit-Flip or Phase-Flip Errors

Let's consider the classical case of bit-flip errors. The error operator E acts as a classical NOT gate:

$$E(0) \rightarrow 1 \text{ and } E(1) \rightarrow 0$$

A trivial solution would be sending the 3-bit message 111 when we actually want to send 1, and sending 000 instead of 0. We say:

$$1_L = 111 \text{ and } 0_L = 000$$

This scheme enables us to detect up to 2 bit flips and determine the position of up to 1 bit flip, thereby allowing us to correct 1 bit errors. For detection, we simply need to check the parity of bits 1 and 2, and bits 2 and 3. The error syndrome is given below:

$b_2 \oplus b_1$	$b_1 \oplus b_0$	Position of Error
0	0	No error
0	1	b_0
1	0	b_2
1	1	b_1

For quantum information, we may be inspired by the same idea. To protect against bit-flip errors, which are in the computational basis, X gate errors, we may encode $|0_L\rangle \rightarrow |000\rangle$ and $|1_L\rangle \rightarrow |111\rangle$. Thus, an arbitrary state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ becomes:

$$|\psi_L\rangle = \alpha|0_L\rangle + \beta|1_L\rangle = \alpha|000\rangle + \beta|111\rangle$$

It is easy to see that this does not violate the no-cloning theorem as the encoded state is different from $|\psi\rangle \otimes |\psi\rangle \otimes |\psi\rangle$. The encoding circuit is also fairly simple to construct. For parity checking operations, we shall include 2 ancillary qubits which perform $CNOT$ operations.

As for phase flip errors, such errors are created by the Pauli-Z gate. If we consider the Hadamard basis, action of this error is simply: $|+\rangle \rightarrow |-\rangle$, and $|-\rangle \rightarrow |+\rangle$. Which means, a clever basis change will help us to treat phase-flip errors just as bit-flip errors. Instead of $CNOT$ gates, for parity check, we use CZ gate.

1.2.2 Shor's [[9,1,3]] code: phase-flip and bit-flip errors

The 3-qubit repetition code is effective in overcoming single phase flip errors or bit flip errors, but not both at the same time. Peter Shor, in 1995 showed how to correct an arbitrary single qubit error by combining the phase-flip and bit-flip correcting codes.[15] Shor's code is a $[[9,1,3]]$-quantum code which encodes a single logical qubit (that is, $2^1 = 2$ basis states) into a space of 9 physical qubits and detects up to 2 independent unitary errors. It considers 3 blocks of 3 qubits each, with each block correcting phase flip errors and combines the three blocks to correct bit flip errors.[15]

1.3 Stabilizer Codes

Stabilizer formalism was first introduced by Daniel Gottesman.[3] Here, valid codewords are specified by a group of measurement operators, say $\mathcal{S}$. $\mathcal{S}$ contains Pauli operators and is commutative. It can also be specified by just the generators of the stabilizer group, $\mathcal{S}_G$. Also note that if $M \in S$ and $|\psi\rangle \in \mathcal{C}$, then $M|\psi\rangle = |\psi\rangle$. That is, all valid codewords are in the common $+1$ eigenspace of the measurement operators specified by the stabilizer group.

In short, for an abelian subgroup $\mathcal{S}$ of the n-qubit Pauli group $\mathcal{G}_n$, we define the codespace $\mathcal{C}$ as the following, if:

$$\mathcal{G}_n = \{\alpha \bigotimes_{i=1\ldots n} P_i : \alpha \in \{\pm 1, \pm\iota\}, P_i \in \{I, X, Y, Z\}\} \quad ,$$
$$\mathcal{C} = \{|\psi\rangle : M|\psi\rangle = |\psi\rangle \ \forall M \in \mathcal{S}\}$$

An $[[n,k,d]]$ quantum code can be specified by $n-k$ stabilizer generators, each acting on n qubits. We may also define logical operators $\bar{X}_i$ and $\bar{Z}_i$, with $1 \leq i \leq k$.

1.3.1 Minimum distance of stabilizer codes

Consider an error operator E, which anticommutes with some stabilizer M, that is, $EM = -ME$, and $\{E, M\} = 0$ acting on a valid codeword $|\psi\rangle$.

$$M(E|\psi\rangle) = -EM|\psi\rangle = -(E|\psi\rangle)$$

Which means $E|\psi\rangle$ is in the -1 eigenspace of M and we can detect E by measuring M.

Now consider the case when E and M commute ($EM = ME, [E, M] = 0$):

$$M(E|\psi\rangle) = EM|\psi\rangle = (E|\psi\rangle)$$

Which means $E|\psi\rangle$ is in the $+1$ eigenspace of M, and we can't detect E by measuring M alone. Now, since the Pauli group for n-qubits: $\{I, X, Y, Z\}^{\otimes n}$ form a basis for $\mathbb{C}^{2^n \times 2^n}$, we shall write any n-qubit error operator as complex linear combination of elements of this Pauli group. Thus, there are in total, three cases:

- **Case I:** $\{E, M\} = 0$: In this case, we can detect the error.
- **Case II:** $E \in S$: Since $E|\psi\rangle = |\psi\rangle$, E does not change the codeword.
- **Case III:** $[E, M] = 0, E \notin S$, $M \in S$: Error cannot be detected.

For the subgroup $\mathcal{S}$, we define the Normalizer subgroup $\mathcal{N}_{\mathcal{G}}(\mathcal{S})$ as the subgroup of $\mathcal{G}$ which contain all elements that commute with every element of $\mathcal{S}$. Case I shall correspond to the complement of the normalizer and Case II and III shall be elements of the normalizer subgroup. The minimum distance of a stabilizer code is defined as the minimum weight of a non-stabilizer element in the normalizer of $\mathcal{S}$, $\mathcal{N}_{\mathcal{G}}(\mathcal{S})$.

$$d = \min\{wt(E) : E \in \mathcal{N}_{\mathcal{G}}(S) \setminus \mathcal{S}\}$$

Here $\mathrm{wt}(E)$ is defined as the number of qubits for which E acts non-trivially (non-identity).

1.4 Cluster States

Cluster states (also known as graph states) are arrays of entangled qubits[1] which form the basis for measurement based quantum computation.[11] It has also been shown that they can be used in constructing quantum error correcting codes.[13]
Consider an undirected graph $\Gamma = (V, E)$ with $|V| = n$. From an n qubit state, we can construct a cluster state $|\psi\rangle$ from graph Γ as follows:

- For each qubit, the initial state is $|+\rangle$. Thus, the initial state of the system is $|+\rangle^{\otimes n}$.
- Assume that the qubit q_i corresponds to the i-th vertex. For each edge $e = (u, v) \in E(\Gamma)$, perform a controlled-Z operation between qubits q_u and q_v.
- We are left with the entangled state, $|\psi\rangle$.

Now, let's see what are the Pauli operators which stabilize this state. In other words, we need operators M such that $M|\psi\rangle = |\psi\rangle$. Consider the simple example of a single disconnected vertex. This would be the $|+\rangle$ state and from the properties of the Pauli matrix, we know that this is stabilized by the Pauli-X operator. Now, consider the graph made of two vertices and an edge (u, v) between them. The corresponding cluster state we construct would be:

$$|\psi\rangle = \frac{1}{\sqrt{2}} CZ(|0+\rangle + |1+\rangle) = \frac{1}{\sqrt{2}}(|0+\rangle + |1-\rangle)$$

It is easy to see that $|\psi\rangle$ is stabilized by the operators $X_u Z_v$ and $Z_u X_v$. For an n-qubit cluster state associated with an n-vertex undirected graph Γ, we get the following stabilizer generators:

$$M_i = X_i \bigotimes_{j \in N(i)} Z_j, 1 \leq i \leq n$$

For each vertex u of Γ, we obtain a stabilizer generator with Pauli-X at the u-th qubit and Pauli-Z at each of its neighbouring qubits. We have seen that stabilizer group is commutative. Since the stabilizers we get from the graph only contain X and Z, we note that two anticommute if there are odd number of vertices having anti-commuting operators (eg: X_1X_2 and Z_1X_2). But the for the generators we constructed, if two vertices u and v are adjacent, the corresponding generators will be of the form $X_uZ_vZ_i\ldots$, $X_vZ_uZ_j\ldots$ which means they anticommute at exactly 2 qubits and thus the generators should commute. If the vertices are not connected by an edge, the corresponding generators will commute at every qubit. Thus, the group generated by the operators we construct is an abelian group.

1.5 Encoding Operation-The Stabilizer Picture

Consider an $[[n,k,d]]$ code. Here, we encode information from k logical qubits to n physical qubits. For this, we start with a cluster state of $n+k$ qubits. Among these $n+k$ qubits, the set of k qubits contain the initial state we need to encode and are called *input qubits*. This state is encoded into the remaining set of n-qubits, called *output qubits*, through measurement of the input qubits.

1.5.1 Initial stabilizer tableau

We first consider the stabilizer generators obtained from the cluster state. These generators form a tableau of $n+k$ stabilizers of the form $X_iZ_aZ_b\ldots$. We may split this into two parts, one containing the X operator and another containing the Z operator and we define the initial stabilizer tableau as follows:

The initial stabilizer tableau can be specified as: $M = \left[M^X \,\middle|\, M^Z\right]$, where M^X and M^Z are $n+k \times n+k$ matrices such that:

$$M^X_{ij} = 1 \text{ If the } i\text{-th stabilizer has } X \text{ at the } j\text{-th qubit } (X_j)$$

$$M^Z_{ij} = 1 \text{ If the } i\text{-th stabilizer has } Z \text{ at the } j\text{-th qubit } (Z_j)$$

M^X and M^Z have all other entries as 0. A straightforward observation, due to the way in which we have constructed the stabilizer from the cluster state, is that $M^X = I_{n+k}$. Similarly, $M^Z_{ij} = 1$, whenever there is an edge between the i-th and j-th vertices in the original graph. In other words, M^z is just the adjacency matrix of the initial graph. Thus, the initial tableau of stabilizer generators, which is of size $(n+k) \times 2(n+k)$ contains information about the initial cluster state.

1.5.2 Evolution of stabilizer tableau upon measurement

With measurement of the set of k input qubits in the X-basis, the state has effectively changed. Which means that the stabilizer group should also change.
Assume that $\langle S_1, S_2 \ldots S_n, S_{n+1}, \ldots S_{n+k} \rangle$ is the initial set of $n+k$ stabilizer generators and we measure qubit q_m in the X basis, which means that the state has collapsed to either $|+\rangle$((eigenvalue $+1$) or $|-\rangle$ (eigenvalue -1). Hence, the changed stabilizer set should have $S_m = \pm X_m$ as the stabilizer corresponding to qubit q_m. But this might cause some other existing generators to anti-commute with S_q. Let the initial set of generators be $\langle S_1, S_2, \ldots S_l, S_{l+1}, \ldots S_{n+k} \rangle$ such that $\{S_1, \ldots, S_l\}$ anticommutes with X_m and the rest commutes with X_m. We get the new stabilizers by:[10]

- Generator set is modified into: $\langle S_1, S_2 S_1, S_3 S_1 \ldots S_l S_1, S_{l+1}, \ldots S_{n+k} \rangle$
- Replace the anti-commuting generator S_1 to $\pm X_m$. Now, the generators form an abelian group.
- If any of the remaining generators contain X_m, replace it with ± 1, which is the measurement result.

For k successive measurements, this process must be repeated k times.

In the stabilizer tableau, these series of steps will translate to some elementary row operations and replacements. An important observation about generators that anti-commute after measurement is that, these generators are precisely those which contain the operator Z_m associated with the measured qubit. This is because since the new stabilizer generator associated with measured qubit is just X_m. Thus, if M_i is a row in the stabilizer tableau which gives an anti-commuting operator, $M_{i,(n+k+m)} = 1$.

- Among the rows which have $M^Z_{i,m} = 1$, we fix a pivot row p. Without loss of generality, let $p = 1$.
- For all other rows M_i which have $M^Z_{i,m} = 1$, we perform the row operation $Mi \rightarrow M_1 \oplus M_i$
- Replace the row M_1 with $M_{1,m} = 1$ and 0 at all other entries.
- Remove the m^{th} row and the columns $m, n+m$.

1.5.3 Example with $[[4, 1, 2]]$ code

The initial cluster state of the $[[4, 1, 2]]$ code is a set of 4 output vertices forming a cycle,[14] and a single input vertex connected to all other vertices. The initial stabilizer generators of this cluster state are:

$$\langle X_1Z_3Z_4Z_5, X_2Z_3Z_4Z_5, X_3Z_1Z_2Z_5, X_4Z_1Z_2Z_5, X_5Z_1Z_2Z_3Z_4\rangle$$

The initial stabilizer tableau would then be:

$$\left[\begin{array}{ccccc|ccccc} 1&0&0&0&0&0&0&1&1&1\\ 0&1&0&0&0&0&0&1&1&1\\ 0&0&1&0&0&1&1&0&0&1\\ 0&0&0&1&0&1&1&0&0&1\\ 0&0&0&0&1&1&1&1&1&0 \end{array}\right]$$

Let R_1 be the pivot. after row operations($R_2 \rightarrow R_2 \oplus R_1, R_3 \rightarrow R_3 \oplus R_1, R_4 \rightarrow R_4 \oplus R_1$), the tableau becomes:

$$\left[\begin{array}{ccccc|ccccc} 1&0&0&0&0&0&0&1&1&1\\ 1&1&0&0&0&0&0&0&0&0\\ 1&0&1&0&0&1&1&1&1&0\\ 1&0&0&1&0&1&1&1&1&0\\ 0&0&0&0&1&1&1&1&1&0 \end{array}\right]$$

After replacing R_1 and removing R_5, C_5, and C_{10}, we end up with the final tableau:

$$\begin{array}{cccc|cccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{array}$$

Thus, the final stabilizer generators of the $[[4, 1, 2]]$ code is:

$$\langle X_1 X_2, Y_1 Z_2 Y_3 Z_4, Y_1 Z_2 Z_3 Y_4 \rangle$$

2

Methods

2.1 Formalizing the Graph

We have seen how the two seemingly separate concepts of graphs and quantum error correcting codes are related through cluster states. We shall soon establish the usefulness and convenience that such graph representation brings for analyzing a given error correcting code. As a precursor, we may now define graphs that provide valid error correcting codes.

Consider the graph $\Gamma = (V, E)$. We partition the vertex set V as $V = \mathcal{I} \cup \mathcal{O} \cup \mathcal{P}$, where $\mathcal{I}$, $\mathcal{O}$, $\mathcal{P}$ correspond to the input, non-pivot output and pivot vertices respectively. Our aim is to encode some information from the input vertices $\mathcal{I}$ into some state of the output vertices $\mathcal{O} \cup \mathcal{P}$.

Fact 1. *For a graph $\Gamma = (\mathcal{I} \cup \mathcal{O} \cup \mathcal{P}, E)$ to represent a valid quantum error correcting code, Γ should satisfy the following:*

- *$\mathcal{I}$ should be an independent set.*
- *$\mathcal{P}$ and $\mathcal{I}$ are perfectly matched.*

The first point is trivial from the fact that we need to encode an arbitrary state on k qubits into the n output qubits. Thus, any operation between the logical input qubits would be meaningless. The latter can be viewed

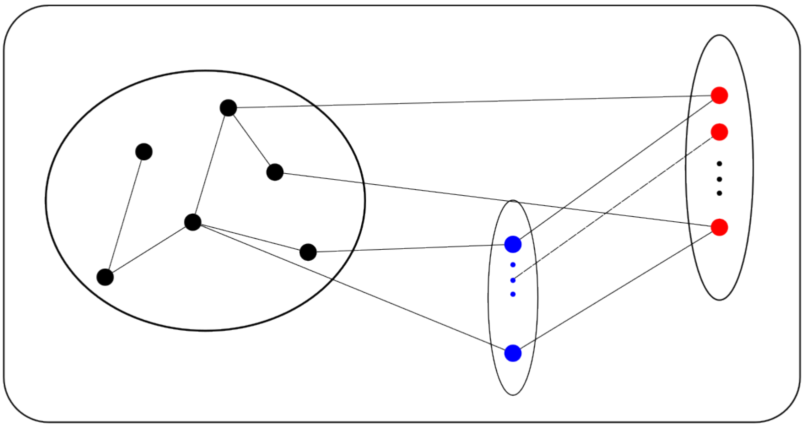

Fig. 2.1: Graph corresponding to some valid QECC

as a consequence of the way in which we have defined the evolution of stabilizers following the measurement of input qubits. For each input qubit, we perform graph manipulation with some adjacent output qubit as a pivot. Note that the row corresponding to the qubit is removed from the final stabilizer tableau. Thus, two input vertices shall not share pivots. Further, if a pivot vertex is connected to more than one input vertex, according to the matrix manipulations, the resulting tableau is dependent on the order of measurement. Thus, we consider the case where $\mathcal{P}$ and $\mathcal{I}$ are perfectly matched.

2.2 Calculating Minimum Distance from Graphs

We have seen how minimum distance may be calculated from the stabilizer tableau. However, such calculations are not immediately apparent from the initial graph state. The essential idea behind minimum distance is mapping logical operations to some physical operations and checking how many physical operations are required to shift from one valid codeword to another. For this, we use the following framework:[6]

2.2.1 Quantum Lights Out

Quantum Lights Out (QLO) is a variation of the classical game of "Lights Out".

Let $\Gamma = (V, E)$ be a graph, where $V = \mathcal{I} \cup \mathcal{O} \cup \mathcal{P},$. Each vertex is associated with a *switch* and a *light*. For each vertex v in V, the allowed operations are:

- **Destroying the light at** v: If the light is ON at v, permanently destroy the light such that it cannot be turned ON anymore.
- **Flipping the switch at** v: Permanently destroy the light at v and toggle the lights at all neighbouring vertices $N(v)$.

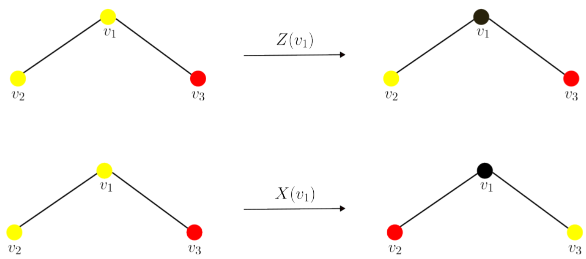

Fig. 2.2: An illustration of operations permitted in QLO

For our convenience and to emphasize the significance of the operations in *QLO*, from here on, we call the former as Z operation and the latter as X operation. The intuition behind the two types of operations in *QLO* is the behaviour of physical X and Z operations being performed on the output qubits. The presence of light at a vertex implies the existence of some implicit Z operation. Using the above operations, we may describe *QLO* now:

Given a graph $\Gamma = (\mathcal{I} \cup \mathcal{O} \cup \mathcal{P}, E)$, we define the *QLO* game on Γ as follows:

- **Round I**: Select some subset $S_{\mathcal{I}} \subseteq \mathcal{I}$ and perform flip operations on all $v \in S_{\mathcal{I}}$.
- **Round II**: In output side, for each vertex $v \in \mathcal{O} \cup \mathcal{P}$, either perform flips or destroy the light at v, or leave v as it is.

The aim of *QLO* is to reach a final configuration where:

- All lights at $v \in \mathcal{O} \cup \mathcal{P}$ are either OFF or destroyed.
- Some lights at $v \in \mathcal{I}$ are ON or we have performed some flips in round I ($S_{\mathcal{I}} \neq \emptyset$).

2.2.2 *QLO* and Minimum Distance

The *QLO* game may be used as a way to translate logical operators on the input qubits into physical operations.

Theorem 1 (Khesin, 2025)**.** *Given a graph $\Gamma = (V, E)$, The minimum distance of the quantum error correcting code is the minimum number of moves made in Round II under an optimum strategy in the QLO instance of Γ.*

$$d = \min_{\mathscr{A}} m(\mathscr{A}, \Gamma) \tag{2.1}$$

Here, $m(\mathscr{A}, \Gamma)$ represents the number of operations made in round 2 of QLO on the instance Γ, using some strategy $\mathscr{A}$.

The key motivation behind QLO is to formulate the quantum minimum distance problem in a purely graph theoretic manner. For this, we use the following:[6]

Fact 2 (ZX-Graph Rule - Khesin, 2025)**.** *For a graph $\Gamma = (\mathcal{I} \cup \mathcal{O} \cup \mathcal{P}, E)$, for any vertex v, A Pauli X on v is equivalent to applying a Pauli Z on all $u \in N(v)$.*

Any operations performed at the input vertices will correspond to some logical operators. In Round I, when we perform flips on some subset $S_{\mathcal{I}} \in \mathcal{I}$, this is equivalent to some logical $\overline{X}_{v_i}$. where $v_i \in S_{\mathcal{I}}$. As a result of these flip operations on $S_{\mathcal{I}}$, when we toggle the lights at $\Delta N(S_{\mathcal{I}})$, this can be thought of as some pending physical Z operations that we must later perform to simulate the logical $\overline{X}$ that we performed in Round I. Under a QLO strategy for which the final configuration is valid, the moves made in Round II at the output side would be some physical operators that are equivalent to some logical operations. And the number of physical operators is exactly the number of moves we make in Round II. Thus, we start with some logical $\overline{X}$ operations and perform the equivalent physical operations resulting in some input lights being turned ON. At the final configuration, the requirement that all output lights must be either turned OFF or destroyed is a consequence of the fact that, in-order for us to receive a valid codeword, the physical operations that we perform on the output qubits should exactly mimic some logical operation so that we end up with a valid codeword. The input lights which get turned ON during this represents some logical $\overline{Z}$ operations. Assume that we did not perform any operations in Round I. Since we need to perform some valid logical operations, some $\overline{Z}$ need to be performed at the input side and thus, at the final configuration, some input lights are ON. The goal of QLO is to find the minimum number of physical operations required to simulate some logical operator.

Through this QLO framework, the following simple relation can be derived:[6]

Theorem 2 (Khesin, 2025)**.** *Minimum distance is upper bounded by the minimum degree of any input or output vertices that are adjacent to some input vertices.*

$$d \leq \min_{v \in \mathcal{I} \cup N(\mathcal{I})} \deg(v) \tag{2.2}$$

Proof. Consider the following strategy:
In round 1, we flip some input vertex v. This causes some $\deg(v)$ many lights at output vertices to turn ON. In round 2, we individually destroy the lights

at each vertex. The total number of moves in this strategy would be $\deg(v)$. For the second case, assume that we perform no operations in round 1 and in round 2, we flip some output vertex u, adjacent to at least one input vertex and destroy all subsequent lights that get turned ON. Here, the number of moves would be the number of output neighbours of $u + 1$. Since u must be connected to at least 1 input vertex, $|N_o(u)| + 1 \leq \deg(u)$. □

3

Results and Discussions

3.1 Partitioned Graph Model

Consider the following graph:

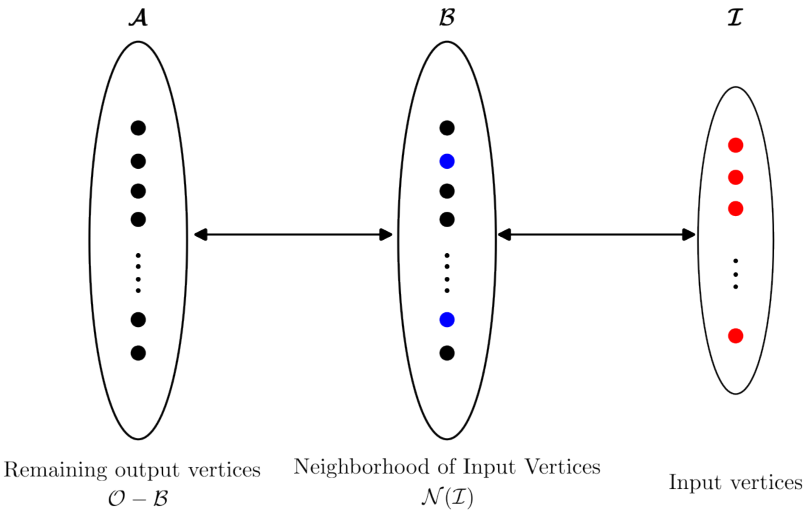

Fig. 3.1: Representation of graphs that we consider

Here, we group the vertices of the graph that we consider into three sets, namely $\mathcal{A}, \mathcal{B}$, and $\mathcal{I}$. Here, $\mathcal{I}$ represents the set of input vertices and $\mathcal{B}$ represents the set of output vertices that are adjacent to at least one input vertex. $\mathcal{B}$ includes the pivot vertices as well as some other output vertices. $\mathcal{A}$ contains output vertices that are not adjacent to any of the input vertices. As we have seen in 1, for such a graph to produce a valid quantum error correcting code, $\mathcal{I}$ has to be an independent set. In the general case, $\mathcal{A}$ and $\mathcal{B}$ are not necessarily independent sets. However, for constructing good codes, we shall see that it is convenient to consider them as independent sets as this allows us to study the QLO by simply considering the edges between $\mathcal{A}$ - $\mathcal{B}$ and $\mathcal{B}$ - $\mathcal{I}$. Another consequence is that the code we produce will be a CSS code.[6]

3.2 Upper Bounds

Now, we may analyze QLO to find out upper bounds for minimum distance on arbitrary graphs. Our motivation is not to exactly calculate the upper bounds, but to find out if there are any graph structures that yield a better minimum distance or if there are any forbidden substructures that heavily reduce the distance. It is useful to note that when we apply a collection of flip operations on a set of vertices S, the lights that get toggled are exactly at those vertices which are connected to an odd number of vertices in S. This is because an even number of connections would toggle the light back to the initial state. From hereafter, we call these vertices where lights are toggled, as *odd neighbours of S*.

Lemma 1. *Given a graph $\Gamma = (\mathcal{A} \cup \mathcal{B} \cup \mathcal{I}, E)$ as described previously, the minimum distance, d of the quantum error correcting code derived from Γ is at most the number of vertices in $\mathcal{B}$ with odd number of neighbours in any subset $S_{\mathcal{I}}$ of $\mathcal{I}$.*

$$d \leq \min_{\substack{S_{\mathcal{I}} \subseteq \mathcal{I} \\ S_{\mathcal{I}} \neq \emptyset}} \Big| \mathop{\triangle}_{v \in S_{\mathcal{I}}} N(v) \Big| \qquad (3.1)$$

Proof. The proof is a simple extension of the first part of Theorem 2. Let d^* be the size of the odd neighbour set of some non-empty subset $S_{\mathcal{I}} \subseteq \mathcal{I}$. Then consider the following QLO strategy:

Round I:	X at $S_{\mathcal{I}}$ which turns on d^*-many lights at $\triangle N(S_{\mathcal{I}})$	0 cost
Round II:	Z at all the output lights which are turned ON	d^* cost

The total number of moves made were d^*. Thus, from Theorem 1, $d \leq d^*$ over all non-empty choices of $S_{\mathcal{I}} \subseteq \mathcal{I}$. A simple example is given in Fig. 3.2. Here, the optimal choice for $S_{\mathcal{I}}$ would be the $\{i_1, i_2, i_3, i_4\} = \mathcal{I}$, with $d^* = 1$. □

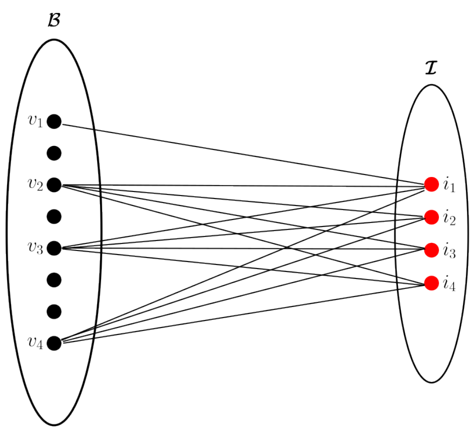

Fig. 3.2: An upper bound specifying edges between $\mathcal{B}$ and $\mathcal{I}$

Lemma 2. *Given a graph* $\Gamma = (\mathcal{A} \cup \mathcal{B} \cup \mathcal{I}, E)$ *as described previously, the minimum distance, d of the quantum error correcting code derived from* Γ *follows:*

$$d \leq \min_{\substack{S_{\mathcal{B}} \subseteq \mathcal{B} \\ \triangle N_{\mathcal{I}}(S_{\mathcal{B}}) \neq \emptyset}} (|S_{\mathcal{B}}| + |\triangle N_{\mathcal{A}}(S_{\mathcal{B}})|) \tag{3.2}$$

Proof. Assume that there exists some subset $S_{\mathcal{B}}$ with some odd neighbours in $\mathcal{I}$ and with $|S_{\mathcal{B}}| + |\triangle N_{\mathcal{A}}(S_{\mathcal{B}})| = d^* < d$. Now consider the following strategy:

Round I:	No flips ($S_{\mathcal{I}} = \emptyset$).	0 cost
Round II:	a) Flips at $S_{\mathcal{B}}$.	$\lvert S_{\mathcal{B}}\rvert$ cost
	b) Z operations at all ON output vertices.	$\lvert\triangle N_{\mathcal{A}}(S_{\mathcal{B}})\rvert$ cost

The final configuration we reach is valid and the total moves made in Round II is $|S_{\mathcal{B}}| + |\triangle N_{\mathcal{A}}(S_{\mathcal{B}})| = d^* < d$, which is in violation of Theorem 1. Thus, for all valid choices of $S_{\mathcal{B}}$, the quantity $|S_{\mathcal{B}}| + |\triangle N_{\mathcal{A}}(S_{\mathcal{B}})|$ need to be at least d.

NOTE: When we consider the case where $\mathcal{B}$ is not an independent set, some light on $v \in S_{\mathcal{B}}$ that gets turned on due to some $X(u)$ operation might also get destroyed due to some $X(v)$ operation. Thus, the number of lights turned ON as a result of flips in $S_{\mathcal{B}}$ is at most the number of odd output neighbours of $S_{\mathcal{B}}$ (as odd neighbours within $S_{\mathcal{B}}$ are destroyed). Thus, Z operations required in Round II is at most $|\triangle N_{\mathcal{A},\mathcal{B}}(S_{\mathcal{B}})|$. □

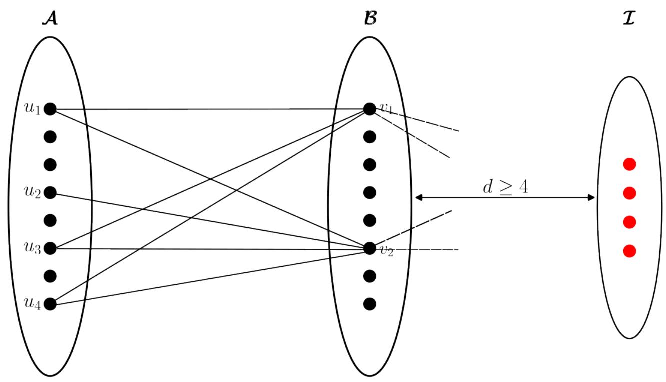

Fig. 3.3: An upper bound specifying edges between $\mathcal{A}$ and $\mathcal{B}$

Lemma 3. *Given a graph $\Gamma = (\mathcal{A} \cup \mathcal{B} \cup \mathcal{I}, E)$ as described previously, the minimum distance, d of the quantum error correcting code derived from Γ follows:*

$$d \leq \min_{\substack{S_{\mathcal{I}} \neq \Phi \\ S_{\mathcal{A}} \subseteq \mathcal{A}}} (|S_{\mathcal{A}}| + |(\triangle N(S_{\mathcal{A}}))\Delta(\triangle N(S_{\mathcal{I}}))|) \tag{3.3}$$

Proof. Assume that there is some non-empty $S_{\mathcal{I}} \subseteq \mathcal{I}$ and $S_{\mathcal{A}} \subseteq \mathcal{A}$ such that $|S_{\mathcal{A}}| + |(\triangle N(S_{\mathcal{A}}))\Delta(\triangle N(S_{\mathcal{I}}))| = d^* < d$. Now, consider the following strategy for a QLO instance on Γ.

Round I:	Flips at $S_{\mathcal{I}}$ which turns ON lights at $\triangle N(S_{\mathcal{I}})$	0 cost
Round II:	a) Flips at $S_{\mathcal{A}}$ which toggles at most $\lvert\triangle N(S_{\mathcal{I}})\rvert$ vertices	$\lvert S_{\mathcal{A}}\rvert$ cost
	b) Z operations at all remaining ON output vertices.	$\leq \lvert(\triangle N(S_{\mathcal{A}}))\Delta(\triangle N(S_{\mathcal{I}}))\rvert$

In Round II, the number of lights toggled by the flip operation is exactly $|\triangle N(S_{\mathcal{I}})|$ if $\mathcal{A}$ is an independent set. If not, just as we have seen in the previous proof, some of the lights which were supposed to be toggled are actually destroyed by the flip operation. The number of lights that are ON after we perform flips on $S_{\mathcal{I}}$ and $S_{\mathcal{A}}$ is at most the number of odd neighbours of $S_{\mathcal{I}}\Delta S_{\mathcal{A}}$. Through this strategy, the total number of moves made is $\leq d^* < d$, which is a contradiction from Theorem 1.

□

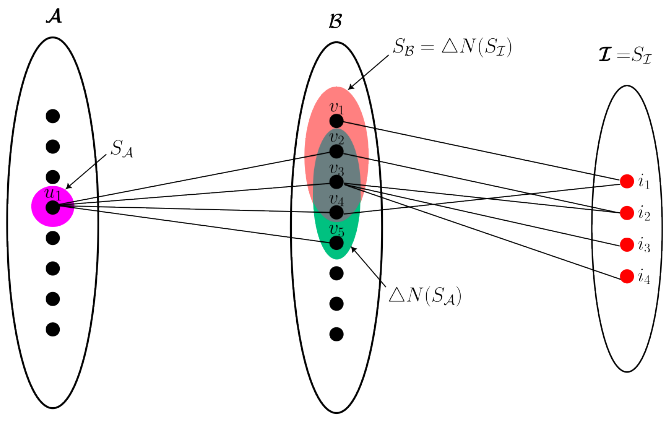

Fig. 3.4: An upper bound showing relation between edges between $\mathcal{I}$-$\mathcal{B}$ and $\mathcal{A}$-$\mathcal{B}$

3.3 Algebraic Analysis of *QLO*

The two round game of QLO establishes a more direct relation between minimum distance and the graph structure of the code. In order to better analyse this relation, we attempt to reformulate QLO. We shall soon consider QLO strategies as vectors in a binary field $\mathbb{F}_2$ and finally simulate QLO using modulo 2 operations using these vectors and the adjacency matrix A_Γ.

Definition 1. Let $f \in \mathbb{F}_2^{n+k}$ be the **flip vector** for the set S_X of the vertices where we perform the flip operation such that $S_X = \{v_i : i \in \text{supp}(f)\}$

Definition 2. Let $l \in \mathbb{F}_2^{n+k}$ represent the **configuration** of lights at vertices at an instant, such that non-zero entries correspond to lights that are either ON or destroyed. $i \in \text{supp}(l) \implies v_i$ is either destroyed or is ON.

Lemma 4. *For any optimal* QLO *strategy that minimises moves made in Round II, we may rearrange the operations so that all X operations are performed first which turn some lights ON. Then we perform Z operations which destroy all output lights that were turned ON.*

Proof. Note that the relative order of two X operations or two Z operations may always be interchanged. We only need to check the case where X operations and Z operations are performed on adjacent vertices and see if ordering matters there. If we have an odd number of X at neighbours and a Z somewhere between $(X_i \dots Z_j \dots X_k)$, this is equivalent to performing all X first which turns ON light at v_j and then performing the Z operation. In the even case, the Z operation does not make any difference since the light is turned OFF from the X operation alone and this is not an optimal QLO. Thus, for optimal strategies, Z operations may happen after all adjacent X operations. □

From Lemma 4, it is easy to see that any optimal strategy of playing QLO is equivalent to choosing some f. We may arrange any QLO strategy into first a series of flip (X) operations which turns ON some lights and destroys some vertices, and then a series of Z operations which destroys all remaining lights that are ON at output vertices.

Proposition 1. *Given a graph $\Gamma = (V, E)$ with adjacency matrix A_Γ, any* QLO *strategy $\mathscr{A}$ on Γ can be simulated using the flip vector f that represents the vertices where we apply X operation in $\mathscr{A}$, as following:*

- *Perform X operations on vertices as specified by f. Configuration after the sequence of flip operations is: $l = A_\Gamma f$*

- *Apply Z operations on non-destroyed output vertices in l to turn off all remaining output lights that are ON.*

Proof. Assume we perform the flip operation on the vertex v_i. The corresponding state after the operation would consist of all its neighbouring vertices turned ON. This is obtained by the i^{th} row of the adjacency matrix. Now consider a sequence of flips on vertices $S_X = v_1, v_2, \dots v_k$. The state that we would end up will consist of vertices that are connected to exactly odd number of vertices in S_X, since even number of toggles would turn the light back OFF. This can be represented by the modulo 2 addition of the adjacency lists of the vertices $v_1, v_2, \dots v_k$, which is simply $A_\Gamma f \mod 2$. Now, since l contains all intact output vertices where lights are ON, we turn them OFF using Z operations. □

Now, we may attempt to reformulate the minimum distance calculation using this picture of *QLO*. We partition the graph into a set of n output vertices ($\mathcal{O}$) and a set of k input vertices ($\mathcal{I}$) as seen in Fact 1. Without loss of generality, we consider the adjacency matrix of the graph A_Γ as:

$$A_\Gamma = \begin{bmatrix} A_o & A_I \\ A_I^T & 0 \end{bmatrix}$$

where A_o corresponds to the subgraph induced by the set of output vertices and A_I corresponds to the the edges between output vertices and input vertices. The $0_{k\times k}$ part is the result of $\mathcal{I}$ being an independent set as seen in Fact 1.

In a similar fashion, we may also partition the flip vector and configuration vector: $f = \begin{bmatrix} f_o & f_i \end{bmatrix}^T$ and $l = \begin{bmatrix} l_o & l_I \end{bmatrix}^T$ respectively into output and input parts. Thus, support of f_o represents the indices of the output vertices where we apply X operation and that of f_i represents the indices of input vertices where X is applied in Round I.

Theorem 3. *Given a graph $\Gamma = (V, E)$ with adjacency matrix A_Γ. The minimum distance d of the quantum error correcting code constructed from the graph is:*

$$d = \min_{validf}(wt(l_o) + wt(f_o) - \langle l_o, f_o \rangle) \tag{3.4}$$

where $l = A_\Gamma f \mod 2$ and either $f_i \neq 0^k$ or $l_I =\neq 0^k$, and $\langle l_o, f_o \rangle$ is Euclidean dot product over $\mathbb{Z}$

Proof. We have seen that any *QLO* strategy can be seen as some sequence of X operations as represented by some f and subsequent Z operations. The configuration we reach after X operations would be:

$$l_X = Af \mod 2 = \begin{bmatrix} A_o & A_I \\ A_I^T & 0 \end{bmatrix} \begin{bmatrix} f_o \\ f_i \end{bmatrix} = \begin{bmatrix} A_o f_o \oplus A_I f_i \\ A_I^T f_o \end{bmatrix} = \begin{bmatrix} l_o \\ l_I \end{bmatrix}$$

Thus, we first have performed $wt(f_o)$ many X operations. Now, we proceed to apply Z operations such that all intact output lights are destroyed. The quantity $wt(l_o)$ tells us how many output lights are ON including those which should have been destroyed while performing the X operations. From this, we subtract $\langle l_o, f_o \rangle$ which tells us how many lights that are assumed ON in l_o were actually destroyed by the X operations done as per f_o. Finally, the total number of Z operations required is $wt(l_o) - \langle l_o, f_o \rangle$. Total number of moves made in round 2 of such a *QLO* strategy would then be $wt(l_o) + wt(f_o) - \langle l_o, f_o \rangle$. From Theorem 1, minimum of such a quantity would be the minimum distance. However, we need to ensure that either we perform some X operations at the input side in round 1 which is satisfied when

$f_i \neq \emptyset$, or that some input light is ON at the end, which will be the case if $l_I = A_I^T f_o \neq 0^k$. An important note here is that while all matrix operations are performed in $\mathbb{F}_2$, the inner product is the standard dot product of $\mathbb{Z}$ acting on binary vectors. □

3.3.1 Relations from the Algebraic Framework

For an arbitrary graph Γ, without loss of generality, we have seen the adjacency matrix A_Γ of the form:

$$A_\Gamma = \begin{bmatrix} A_o & A_I \\ A_I^T & 0 \end{bmatrix}$$

Here, we also define the matrix $A' = \begin{bmatrix} A_o & A_I \end{bmatrix}$. Now, using the vector framework of QLO, we obtain the following relations:

Lemma 5. *Let adjacency matrix A_Γ as given above describe a graph Γ. Let f be the flip vector corresponding to some valid QLO strategy on Γ and d be the minimum distance of the* QECC *derived from Γ. Then:*

$$d \leq wt(f_o), \forall f_o \in E_1(A_o) \setminus \ker(A_I^T) \tag{3.5}$$

Proof. Consider the flip vector f with $f_i = 0^k$. This means that in Round I of QLO, we do not perform any flip operations. Now for f to be valid, flip operations corresponding to f_0 should turn ON some input lights. Hence, $l_I = A_I^T f_o \neq 0^k \implies f_o \notin \ker(A_I^T)$.
Now since here, $l_o = A_o f_o \oplus A_I f_i = A_o f_o$. from Theorem 3, we get:

$$d \leq wt(A_o f_o) + wt(f_o) - \langle A_o f_o, f_o \rangle$$

For $f_o \in E_1(A_o); A_o f_o, = f_o$:

$$\implies d \leq wt(f_o) + wt(f_o) - \langle f_o, f_o \rangle$$
$$\implies d \leq 2wt(f_o) - wt(f_o)$$
$$\implies d \leq wt(f_o)$$

NOTE: All vector arithmetic operations are performed in $\mathbb{F}_2$ □

Lemma 6. *For a* QECC *derived from a graph* Γ *with adjacency matrix* A_Γ *as described above, for all valid flip vectors* f *corresponding to valid* QLO *strategies;*

$$d \geq \min_{\textit{valid } f}(\max(wt(A'f), wt(f_o))) \tag{3.6}$$

Proof. We start from Theorem 3:

$$d = \min_{\text{valid} f}(wt(l_o) + wt(f_o) - \langle l_o, f_o\rangle)$$

Now, $l_o = A'f$. The inner product $\langle A'f, f_o\rangle$ tells us, among the output vertices v with destroyed lights, how many were turned ON after flip operations at vertices other than v. This calculates how many vertices are common to f_o and $A'f$, which is at most the minimum of the norms of these two vectors.

$$\langle A'f, f_o\rangle \leq \min(wt(A'f), wt(f_o))$$
$$\implies wt(A'f) + wt(f_o) - \langle A'f, f_o\rangle \geq wt(A'f) + wt(f_o) - \min(wt(A'f), wt(f_o))$$
$$\implies wt(A'f) + wt(f_o) - \langle A'f, f_o\rangle \geq \max(wt(A'f), wt(f_o))$$

Let f^* be the optimal *QLO*.

$$wt(A'f^*) + wt(f_o^*) - \langle A'f^*, f_o^*\rangle \geq \max(wt(A'f^*), wt(f_o^*) \geq \min_{\text{valid} f}(\max(wt(A'f), wt(f_o)))$$

The LHS is d:

$$d \geq \min_{\text{valid } f}(\max(wt(A'f), wt(f_o)))$$

□

3.4 Construction of "Good" Codes

From the bounds we derived in Section 3.2, we have seen how some substructures are forbidden in the graph and how others are necessary. For instance, Lemma 1 necessitates that the connections from the input vertices to output sets should be in such a way that, for any subset of input vertices, there are at least d many odd neighbours in the output vertices. This is akin to the minimum distance of a classical code. Our approach is to start with a

convenient construction and iteratively improvise by trying to decrease the total number of qubits we use.
For the proof of correctness of these codes, we consider the two cases for a valid QLO strategy given in Theorem 1. Either we have to perform some flips at $u \in \mathcal{I}$ at Round I or we need to perform some flips at $v \in \mathcal{B}$ at Round II such that some input light is turned ON. Then we show that in both cases, we arrive at a state where the lights are turned ON in such a way that the total cost becomes at least d if we try to switch them OFF. We also use Lemma 4 to confine our analysis to QLO strategies where Z operations are performed only after all X operations. Thus, it suffices to show that any sequence of valid X operations with cost x_i, leaves at least $d - x_i$ many lights ON.

3.4.1 Code I: A Naive Attempt

We first consider a very simple case which avoids any complicated structures. As a result, we end up with a $[[d^2, 1, d]]$ code which can be concatenated to obtain $[[kd^2, k, d]]$ codes. Given an independent set of k vertices, we construct the graph in the following manner:

For each input vertex u_i, add d vertices v_u in the set $\mathcal{B}$ and add edges between these output vertices and u_i. Further, connect each vertex in $\mathcal{B}$ with $d-1$ many output vertices in $\mathcal{A}$. Here, the total blocklength of the code would be $n = kd^2$. We say that a vertex v in $\mathcal{B}$ is connected to a bag $\mathcal{A}_v$ of of $d-1$ vertices in $\mathcal{A}$.

Theorem 4. Code I *provides a* QECC *with minimum distance* $d(Q) \geq d$.

Proof. We show that for any valid QLO strategy, after all X operations are performed, the total cost satisfies $a_i + b_i \geq d$, where a_i is the number of operations performed on output vertices and b_i is the number of output lights that remain ON.

For a valid QLO strategy, either some flips are performed at input vertices in Round I, or some input light must be turned ON, which requires flips at

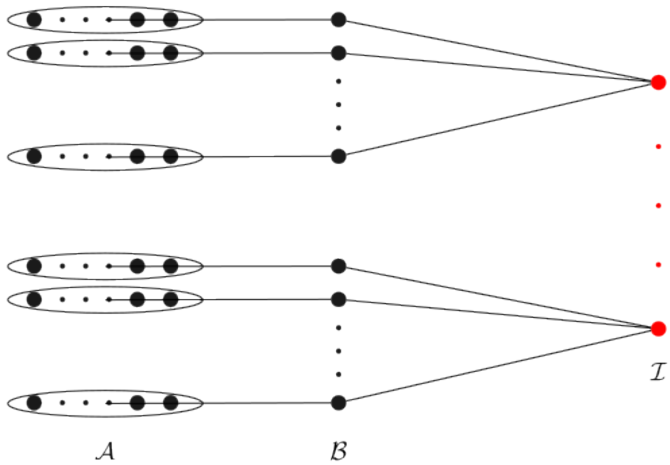

Fig. 3.5: Code I: A simple $[[kd^2, k, d]]$ code

$\mathcal{B}$. We consider these two cases.

Case 1: A flip is performed at some $v \in \mathcal{B}$.

Let $\mathcal{A}_v$ denote the bag of size $d-1$ connected to v. Suppose x_i flips are performed within $\mathcal{A}_v$. Since each flip in $\mathcal{A}_v$ destroys one light, the number of intact lights remaining in $\mathcal{A}_v$ is at least $d-1-x_i$. A flip at v turns ON all these intact lights.

Thus,

$$b_i \geq d - 1 - x_i.$$

On the other hand, the number of operations includes the flip at v and the x_i flips in $\mathcal{A}_v$, so

$$a_i \geq x_i + 1.$$

Therefore,

$$a_i + b_i \geq (x_i + 1) + (d - 1 - x_i) = d.$$

Case 2: A flip is performed at some $v \in \mathcal{I}$.

Each input vertex is connected to d distinct vertices in $\mathcal{B}$, and these neighbourhoods are disjoint. Hence, a flip at any input vertex produces at least d ON lights in $\mathcal{B}$. Since no flips are performed in $\mathcal{B}$ in this case, each such light must be removed individually using either a Z operation at that vertex or a flip in the corresponding bag in $\mathcal{A}$, each costing one move.

Thus, at least d operations are required, and hence $a_i + b_i \geq d$.

In both cases, we have $a_i + b_i \geq d$. □

An important observation here is that the graph we construct would be a disconnected graph with k-many components. This is a consequence of the fact that we have encoded each input qubit independently and the resultant code is in fact, a concatenated code from k-many $[[d^2, 1, d]]$ codes. When $d = 3$, the construction generates the simple $[[9, 1, 3]]$ Shor code.[15]

3.4.2 Code II: A Better Approach

In the previous construction, the neighbours of input vertices were disjoint. From Lemma 1, the degree of an input vertex must be at least d. If common neighbours were allowed for input vertices, this would reduce the blocklength significantly. The first improvement comes from this observation. We also note that Lemma 1 has some similarity to classical codes that we may use. We require that for any subset of input vertices, there are at least d-many vertices in $\mathcal{B}$ which are connected to an odd-number of input vertices from the subset we consider. This is similar to some modulo 2 linear combination of rows of the adjacency matrix having weight at least d. Assume given k input vertices, and a classical binary code with parameters $[[n_c, k, d]]$ and generator matrix G in its standard form, we construct the following adjacency matrix:

$$A_{\mathcal{B},\mathcal{I}} = \begin{array}{c c} & \begin{array}{cc} \mathcal{B} & \mathcal{I} \end{array} \\ \begin{array}{c} \mathcal{B} \\ \mathcal{I} \end{array} & \begin{pmatrix} 0_{k\times k} & G^T \\ G & 0_{n_c\times n_c} \end{pmatrix} \end{array}.$$

The number of odd-neighbours of any subset of $\mathcal{I}$ would be the weight of some linear combination of the first k-rows in the above matrix. Since G is the generator matrix of some $[[n_c, k, d]]$ code, this is always $\geq d$. To complete the construction, we may complete the connections between the vertex sets

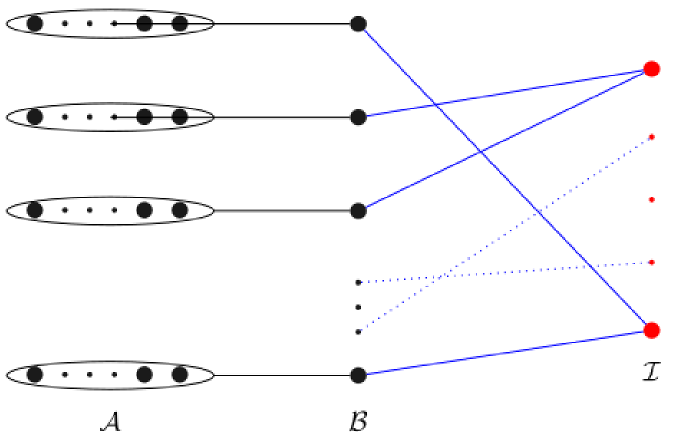

Fig. 3.6: Code II: Graph for an $[[n_c d, k, d]]$ code

$\mathcal{B}$ and $\mathcal{A}$ as seen previously. This would require $d-1$ many vertices in $\mathcal{A}$ for each vertex in $\mathcal{B}$. The total number of output vertices would then be $n_c d$. For classical codes, $n_c = O(d)$. Hence, for these quantum codes, since $n = n_c d$, for this particular construction, $n = O(d^2)$.

Now we may see what the adjacency matrix of the graph that we construct. Consider the following matrix which represents the connections between $\mathcal{A}$ and $\mathcal{B}$:

$$A_{\mathcal{B},\mathcal{A}} = \mathcal{B} \overset{\mathcal{A}}{\begin{pmatrix} 1_{\times(d-1)} & 0\cdots & \cdots & \cdots & \cdots \\ \cdots 0 & 1_{\times(d-1)} & 0\cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0\cdots & \cdots & \cdots & \cdots 0 & 1_{\times(d-1)} \end{pmatrix}}$$

Then the adjacency matrix of the graph that we construct would be:

$$A_\Gamma = \begin{matrix} \\ \mathcal{A} \\ \mathcal{B} \\ \mathcal{I} \end{matrix} \begin{matrix} \begin{matrix} \mathcal{A} & \mathcal{B} & \mathcal{I} \end{matrix} \\ \begin{pmatrix} 0 & A_{\mathcal{B}\mathcal{A}}^T & 0 \\ A_{\mathcal{B}\mathcal{A}} & 0 & G^T \\ 0 & G & 0 \end{pmatrix} \end{matrix}$$

Theorem 5. Code II *provides a* QECC *with minimum distance* $d(Q) \geq d$.

Proof. The proof follows the same idea as in the previous construction. Since the edges between $\mathcal{A}$ and $\mathcal{B}$ remain unchanged, any flip at a vertex in $\mathcal{B}$ activates a bag of size $d-1$ in $\mathcal{A}$, which increases the total cost. Hence, in any optimal QLO strategy, flips at $\mathcal{B}$ are avoided.

Since a valid QLO strategy requires either flips in Round I or input lights to be ON, and flips at $\mathcal{B}$ are avoided, we must perform flips at input vertices. Thus, $S_{\mathcal{I}} \neq \emptyset$.

By construction, the adjacency between $\mathcal{I}$ and $\mathcal{B}$ is defined using the generator matrix of a classical $[n_c, k, d]$ code. Hence, any nonempty subset of input vertices produces at least d vertices in $\mathcal{B}$ with an odd number of neighbours, i.e., at least d lights are turned ON in $\mathcal{B}$.

Since flips at $\mathcal{B}$ are not allowed, each such light must be turned OFF individually, either by applying $Z(v)$ or by performing an X operation at a vertex in the corresponding bag $\mathcal{A}_v$. As the bags are disjoint, each such operation removes exactly one light and costs one move.

Hence, at least d operations are required to turn off all the lights, and therefore the minimum distance satisfies $d(Q) \geq d$ □

3.4.3 Code III: Further Improvements

In the previous code, we tried to reduce the vertices required in $\mathcal{B}$. A similar attempt could be made for reducing the vertices in $\mathcal{A}$. Here, Lemma 3 is used for gathering insights about what sort of connections between $\mathcal{A} - \mathcal{B}$ and $\mathcal{B} - \mathcal{I}$ bring down the minimum distance.

Consider a classical $[n_c, k+r, d_0]$ code $\mathcal{C}_0$ with generator matrix G_0 in the standard form. We partition the rows of G_0 into matrices G_1 and G_2 of dimension $k \times n$ and $r \times n$ respectively such that $G_0 = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix}$. Here, G_1 and G_2 are generators for the subcodes $\mathcal{C}_1$ and $\mathcal{C}_2$ of $\mathcal{C}_0$. At $\mathcal{A}$, let there be r many bags of $d_0 - 1$ vertices each such that if a vertex v in $\mathcal{B}$ is connected to bag $\mathcal{A}_u$, then all the $d_0 - 1$ vertices in bag $\mathcal{A}_u$ is connected to v. Thus, for

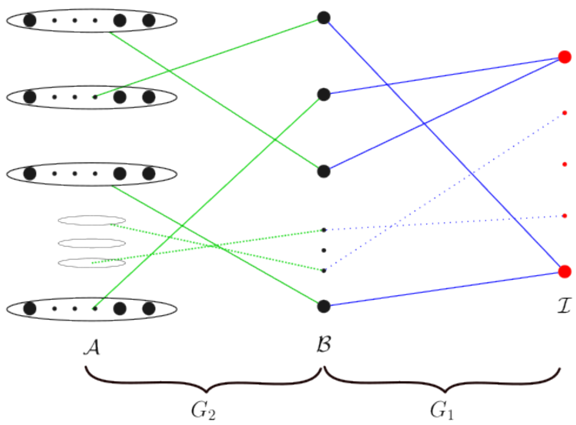

Fig. 3.7: Code III: Graph for a $[[O(kd), k, d]]$ code

ease of description, it suffices to talk about the connections between vertices in $\mathcal{B}$ and bags in $\mathcal{A}$, which we represent as some fused vertex. Now, the following adjacency matrix where instead of vertices in $\mathcal{A}$, we consider these fused vertices which represents bags:

$$A_\Gamma = \begin{array}{c} \\ \mathcal{A} \\ \mathcal{B} \\ \mathcal{I} \end{array} \begin{array}{c} \begin{array}{ccc} \mathcal{A} & \mathcal{B} & \mathcal{I} \end{array} \\ \begin{pmatrix} 0 & G_1 & 0 \\ G_2^T & 0 & G_1^T \\ 0 & G_1 & 0 \end{pmatrix} \end{array}$$

Theorem 6. *The minimum distance of* QECC *from* Code III *using a classical code* $\mathcal{C}_0$ *of minimum distance* d *and subcodes* $\mathcal{C}_1$ *and* $\mathcal{C}_2$ *is:*

$$d(Q) \geq \min(d(\mathcal{C}_0), d(\mathcal{C}_2^\perp))$$

Proof. By Lemma 4, any optimal *QLO* strategy may be reordered so that all X operations are performed first, followed by all Z operations. We prove the result by considering two cases.

Case 1: At least one flip is performed on a vertex in $\mathcal{A} \cup \mathcal{I}$.

Let $m_1 \in \mathbb{F}_2^k$ encode the choice of flipped input vertices in $\mathcal{I}$, and let $m_2 \in \mathbb{F}_2^r$ encode the choice of bags in $\mathcal{A}$ from which an odd number of flips is performed. Since only the parity of flips in each bag matters, each bag may be treated as a single vertex.

After the X operations, the set of lights turned on in $\mathcal{B}$ is represented by

$$c = c_1 \oplus c_2,$$

where

$$c_1 = \begin{bmatrix} m_1^T & 0_{1\times r} \end{bmatrix} G_0, \qquad c_2 = \begin{bmatrix} 0_{1\times k} & m_2^T \end{bmatrix} G_0.$$

Since G_0 generates $\mathcal{C}_0$, the vector c is a codeword of $\mathcal{C}_0$. Moreover, for a valid strategy in this case, $c \neq 0$. Therefore,

$$\mathrm{wt}(c) \geq d(\mathcal{C}_0).$$

Because $\mathcal{B}$ is an independent set, each ON vertex in $\mathcal{B}$ must be turned off individually by Z or X operations which costs at least one move per vertex. Hence the total cost in Round II is at least $d(\mathcal{C}_0)$.

Case 2: No flips are performed on $\mathcal{A} \cup \mathcal{I}$.

Since the strategy must be valid, some input light must be turned on in Round I. Thus we must flip a nonempty set $S_{\mathcal{B}} \subseteq \mathcal{B}$. If $S_{\mathcal{B}}$ has an odd neighbourhood in $\mathcal{A}$, then the corresponding bags in $\mathcal{A}$ become ON, and each such bag contributes at least $d-1$ additional moves to turn off all of its lights. Therefore, to obtain a strategy of minimum cost, we must choose $S_{\mathcal{B}}$ so that it has no odd neighbours in $\mathcal{A}$.

This is equivalent to finding the minimum number of linarly dependent columns of G_2, and the which is exactly $d(\mathcal{C}_2^{\perp})$. Hence any valid strategy of this type costs at least $d(\mathcal{C}_2^{\perp})$.

Combining the two cases, for every valid QLO strategy:

$$d(Q) \geq \min\left(d(\mathcal{C}_0), d(\mathcal{C}_2^{\perp})\right)$$

□

Here, if the original classical code $\mathcal{C}_0$ has parameters $[2r, k+r, d_0]$ and $\mathcal{C}_2$ is a self-dual code with parameters $[2r, r, d_2]$, where $r > k$, then $d(\mathcal{C}_2^{\perp}) \geq d(\mathcal{C}_0)$. The resultant quantum code shall have parameters $[[r(d+1), k, d_0]]$.

4

Conclusion

The primary objective of this work was to establish relations between valid graph states and minimum distance of the *QECC* they produce. We considered a simplified graph instance in Section 3.1; $\Gamma = (\mathcal{A} \cup \mathcal{B} \cup \mathcal{I}, E)$, where $\mathcal{A}, \mathcal{B}$, and $\mathcal{I}$ are independent sets. Through our analysis of *QLO* in Section 3.2, the following upper bounds were derived:

$$\text{Lemma 1:} \qquad d \leq \min_{\substack{S_\mathcal{I} \subseteq \mathcal{I} \\ S_\mathcal{I} \neq \emptyset}} |\mathop{\triangle}_{v \in S_\mathcal{I}} N(v)|$$

$$\text{Lemma 2:} \qquad d \leq \min_{\substack{S_\mathcal{B} \subseteq \mathcal{B} \\ \triangle N_\mathcal{I}(S_\mathcal{B}) \neq \emptyset}} (|S_\mathcal{B}| + |\triangle N_\mathcal{A}(S_\mathcal{B})|)$$

$$\text{Lemma 3:} \qquad d \leq \min_{\substack{S_\mathcal{I} \neq \Phi \\ S_\mathcal{A} \subseteq \mathcal{A}}} (|S_\mathcal{A}| + |(\triangle N(S_\mathcal{A}))\Delta(\triangle N(S_\mathcal{I}))|)$$

We also proposed an algebraic framework for *QLO* using the adjacency matrix A_Γ and by representing the choice of vertices where X operation is performed using a vector $f \in \mathbb{F}_2^{n+k}$. The vector $A_\Gamma f \mod 2 = l \in \mathbb{F}_2^{n+k}$ represents the configuration after flips are performed. A valid *QLO* strategy requires that either $f_i \neq 0^k$ or $l_I = A_I^T f_o \mod 2 \neq 0^k$. Here, minimum distance is calculated as follows:

$$\text{Theorem 3:} \quad d = \min_{\text{valid} f} (wt(l_o) + wt(f_o) - \langle l_o, f_o \rangle)$$

The vector based framework allowed us to arrive at additional relations, including a lowerbound:

$$\text{Lemma 5:} \quad d \leq wt(f_o), \forall f_o \in E_1(A_o) \setminus \ker(A_I^T)$$
$$\text{Lemma 6:} \quad d \geq \min_{\text{valid } f}(\max(wt(A'f), wt(f_o)))$$

The key idea here is not to calculate the bounds for an arbitrary graph state, but to identify substructures within graphs which restrict the corresponding *QECC* from attaining a high minimum distance. We then attempt to construct graphs which avoid such substructures and finally prove the minimum distance of the associated *QECC* using the *QLO* framework. The methodology here is to build upon the graph we defined in Section 3.1 and improve the parameters of *QECC* by iteratively reducing the number of output vertices. We provide three graph constructions which provide valid CSS codes:

Code I Section 3.4.1	Parameters $[[kd^2, k, d]]$	$n = O(kd^2)$
Code II Section 3.4.2	Parameters $[[n_c d, k, d]]$	$n = O(d^2)$
Code III Section 3.4.3	Parameters $[[r(d+1), k, d]]$	$n = O(rd)$

These codes are potentially comparable to widely used families of codes such as hypergraph product codes ($n = \Theta(d^2)$). On comparison with existing works on graph-centric constructions, these codes are similar to the recent $n = O(2^d)$ code[18] and $n = O(d^5)$ code.[6] However, decoding strategies for these codes need to be investigated further.

4.1 Future Directions

In addition to parameters like n, k, d, the robustness of a quantum error correcting code is also determined by the efficiency of the decoding algorithm. For classical codes, such improvements were shown when codes were constructed using expander graphs.[16] It would be of interest to see if there are decoding strategies which are better suited for these families of codes, that are able to recover a large portion of the errors.

It may be possible to construct better codes, using the lower bounds we derived in Section 3.3. Also, there may exist relations between the spectral properties of graphs and the parameters of *QECC*.[17] Since we have seen that minimum distance can be influenced by very local substructures rather than the global structure of the graph, these spectral properties may be difficult to discover.

Bibliography

[1] Hans J. Briegel and Robert Raussendorf. "Persistent Entanglement in Arrays of Interacting Particles". In: *Physical Review Letters* 86.5 (Jan. 2001), pp. 910–913.

[2] R. Gallager. "Low-density parity-check codes". In: *IRE Transactions on Information Theory* 8.1 (1962), pp. 21–28.

[3] Daniel Gottesman. *Stabilizer Codes and Quantum Error Correction.* 1997. arXiv: `quant-ph/9705052 [quant-ph]`.

[4] M. Hein, J. Eisert, and H. J. Briegel. "Multiparty entanglement in graph states". In: *Phys. Rev. A* 69 (6 2004), p. 062311.

[5] Upendra Kapshikar and Srijita Kundu. "On the Hardness of the Minimum Distance Problem of Quantum Codes". In: *IEEE Transactions on Information Theory* 69.10 (2023), pp. 6293–6302.

[6] Andrey Boris Khesin, Jonathan Z. Lu, and Peter W. Shor. "Universal Graph Representation of Stabilizer Codes". In: *PRX Quantum* 6 (4 Nov. 2025), p. 040325.

[7] Emanuel Knill, Raymond Laflamme, and Lorenza Viola. "Theory of Quantum Error Correction for General Noise". In: *Physical Review Letters* 84.11 (Mar. 2000), pp. 2525–2528.

[8] Mike Krebs and Anthony Shaheen. *Expander Families and Cayley Graphs: A Beginner's Guide.* Oxford University Press, 2011.

[9] Alexander Lubotzky. "Expander Graphs in Pure and Applied Mathematics". In: *arXiv* (2011). arXiv: `1105.2389 [math.CO]`.

[10] Ashlesha Patil and Saikat Guha. *Clifford Manipulations of Stabilizer States*. 2023. arXiv: `2312.02377 [quant-ph]`.

[11] Robert Raussendorf and Hans J. Briegel. "A One-Way Quantum Computer". In: *Phys. Rev. Lett.* 86 (22 2001), pp. 5188–5191.

[12] D. Schlingemann. "Stabilizer codes can be realized as graph codes". In: *Quantum Info. Comput.* 2.4 (June 2002), pp. 307–323.

[13] D. Schlingemann and R. F. Werner. "Quantum error-correcting codes associated with graphs". In: *Physical Review A* 65.1 (Dec. 2001).

[14] Swayangprabha Shaw et al. *Construction of non-CSS quantum codes using measurements on cluster states*. 2023. arXiv: `2207.04996 [quant-ph]`.

[15] Peter W. Shor. "Scheme for reducing decoherence in quantum computer memory". In: *Phys. Rev. A* 52 (4 Oct. 1995), R2493–R2496.

[16] M. Sipser and D.A. Spielman. "Expander codes". In: *IEEE Transactions on Information Theory* 42.6 (1996), pp. 1710–1722.

[17] Daniel A. Spielman. *Spectral and Algebraic Graph Theory*. Lecture Notes. 2025.

[18] Sowrabh Sudevan et al. "Codeword Stabilized Codes From m-Uniform Graph States". In: *IEEE Journal on Selected Areas in Information Theory* 6 (2025), pp. 296–310.