Novel Methods in Quantum Error Correction

by

Tomas Jochym-O'Connor

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Physics

Waterloo, Ontario, Canada, 2016

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Quantum error correction is the backbone of fault-tolerant quantum computation, a necessary requirement for any large scale quantum computer. The fault-tolerance threshold theorem has long been a target for experimental precision, allowing for the possibility of reducing logical error rates to arbitrarily low levels without excessive overhead. While there are many promising fault-tolerant architectures, the path towards the most practical fault-tolerant scheme is far from decided and may vary for differing physical models. This thesis proposes new schemes for universal fault-tolerant quantum computation in both the concatenated and topological code settings.

Through the concatenation of two different error correcting codes, a set of universal fault-tolerant gates can be obtained without requiring the need for magic state distillation. A lower bound of 1.28×10^{-3} for the fault-tolerance threshold under circuit level depolarizing noise is obtained. Additionally, stacked codes are proposed as a means to simulate the action of a 3D topological code in 2D, allowing for the application of a universal set of transversal operations. While fault-tolerant, unfortunately the scheme does not exhibit a threshold due to the decreasing pseudo-threshold with growing code distance, yet points to potential interesting avenues for fault-tolerant computation in 2D without distillation.

One of the primary avenues to constructing fault-tolerant logical operations is through transversal operations. In this thesis, the set of single qubit logical gates that can be implemented transversally are characterized and determined to all belong to the Clifford hierarchy. Moreover, any diagonal two-qubit operation that can be applied transversally must belong to the same level of the Clifford hierarchy as the set of gates that can be implemented in the single-qubit case.

The opposite to quantum error correction is privacy, where the output of a channel is disguised from its input. The two are fundamentally related through the complementary channel construction. This thesis presents a new class of private quantum channels, expanding the existing class beyond a seemingly fundamental restriction. This yields interesting insights into the structure of quantum information and the leaking of information to external environments. Additionally, the duality is only recovered when extending the complementary channel to sufficiently high environmental dimension.

Finally, the error properties of bucket brigade quantum Random Access Memory (qRAM) are assessed. It is determined that using the bucket brigade qRAM architecture for the running of Grover's algorithm necessitates reducing the error rate of the individual components to exponentially small levels for an exponential sized memory. As such, fault-tolerant architectures will likely play an essential role in the construction of such computing primitives.

Acknowledgements

Firstly, I would like to thank my PhD supervisor Raymond Laflamme for his constant support, he has provided continuous guidance in my six years at Waterloo. Not only is he a great research mentor who encourages me to pursue my research interests, he continually gives valuable academic advice and provided ample opportunities for collaboration and is constantly inspiring.

I would like to thank Microsoft Research, and particularly Krysta Svore, for giving me the opportunity to work with the QuArC group. I would like to thank David Poulin, and John Preskill for inviting me to visit their respective groups and have stimulating discussions. I would like to thank David Kribs for introducing me to the notion of privacy in quantum information and for always being very welcoming to visit his group and engage in collaborations. I would like to thank Michele Mosca for his invitation to collaborate with his group and support of my academic endeavours. I would like to thank Stephen Bartlett for hosting my visit down under and for continued efforts as we collaborated at a distance. Thank you all for your gracious hospitality and stimulating positive research environments.

I strongly believe that scientific research is not only completed by individuals, but is rather a continuous communal effort and I am grateful to have had the opportunity to work and learn at the Institute for Quantum Computing (IQC) and the University of Waterloo. I would like to thank in particular the following individuals from IQC and abroad for many enlightening research discussions: Jonas Anderson, Srinivasan Arunachalam, Michael Beverland, Kamil Bradler, Sergey Bravyi, Aharon Brodutch, Christopher Chamberland, Josh Colmes, Ben Criger, Nicolas Delfosse, Guillaume Duclos-Cianci, Kent Fisher, Alan Geller, Vlad Gheorghiu, David Gosset, Daniel Gottesman, Chris Granade, Patrick Hayden, Mark Howard, Vadym Kliuchnikov, Hemant Katiyar, Aleksander Kubica, Jeremy Levick, Dawei Lu, Xian Ma, Osama Moussa, Adam Paetznick, Annie Park, Daniel Park, Gina Passante, Fernando Pastawski, Sarah Plosker, Daniel Puzzuoli, Ben Reichardt, Nayeli Rodriguez-Briones, Yuval Sanders, Priyaa Varshinee Srinivasan, Joel Wallman, Zak Webb, Dave Wecker, Nathan Wiebe, Mark Wilde, Beni Yoshida; you have all taught me very much.

I would also like to acknowledge the support of various funding agencies. I am grateful for the Vanier Scholarship from the National Science and Research Council (NSERC) and Government of Canada, as well as from the Fonds de recherche Nature et Technologies from the Government of Québec, Ontario Graduate Scholarship and CryptoWorks21. I would also like to thank Mike Lazaridis, Industry Canada, and the Canadian and Ontario Government for their continuous support of IQC, such an amazing institute would not be possible without them.

I am grateful to Ken Brown, Joseph Emerson, Robert König, Matteo Mariantoni, and Ashwin Nayak, for their insights and feedback as part of my PhD advisory and examining committees.

I would like to thank all of my teachers at F.A.C.E., Dawson, McGill, for providing a well-rounded education and helping me determine my path in life.

Last but definitely not least, I would like to thank all of my family and friends in Montreal and Kitchener/Waterloo for their continuous support throughout my studies. I would like to say a special thanks to my aunt Betty for her love, support, and words of encouragement. I would like to especially thank Marta for always encouraging me to pursue my interests and dreams. *Je vous aime.*

Dedication

To my parents, Anna and Roger. You always encouraged me to pursue my interests and be who I wanted to be. I am forever grateful.

Table of Contents

List of Tables xi

List of Figures xii

1 Quantum mechanics: a new realm for computation 1

1.1 Research motivation . . . 1

1.2 The fundamentals of quantum information . . . 3

1.2.1 From bits to qubits . . . 3

1.2.2 Superposition in quantum algorithms . . . 12

1.3 Quantum error correction . . . 15

1.3.1 Overcoming quantum noise . . . 15

1.3.2 Conditions for quantum error correction . . . 22

1.3.3 The stabilizer formalism . . . 23

1.3.4 Fault-tolerant quantum computation . . . 25

2 Transversality and fault-tolerant quantum error correction 29

2.1 Fault-tolerant universal gate sets . . . 29

2.1.1 Transversal gate sets . . . 29

2.1.2 Magic state distillation . . . 38

2.2 Universal fault-tolerance via code concatenation . . . 44

2.2.1 Overview . . . 44

2.2.2 Concatenated QEC . . . 44
2.2.3 Example: A 105-qubit quantum error correcting code . . . 47
2.2.4 Summary & outlook . . . 49
2.3 Thresholds for universal concatenated codes . . . 50
2.3.1 Fault-tolerance threshold under depolarizing noise . . . 51
2.3.2 Concatenated 105-qubit thresholds . . . 56
2.3.3 Summary & outlook . . . 63
2.4 Gate restrictions for transversal operations in stabilizer codes . . . 64
2.4.1 Overview . . . 64
2.4.2 Single qubit transversal Z rotations . . . 65
2.4.3 Multi-block gates . . . 88
2.4.4 Summary & outlook . . . 97

3 Topological methods for universality **99**
3.1 Topological codes . . . 99
3.1.1 The Toric and surface codes . . . 99
3.1.2 2D color code . . . 101
3.1.3 3D color code . . . 102
3.2 Code conversion for topological stabilizer codes . . . 104
3.2.1 Overview . . . 104
3.2.2 Transforming to the stacked code . . . 105
3.2.3 Proof of transversal logical $\pi/8$ gate for the stacked code . . . 108
3.2.4 Unfolding the stacked code: A 2D implementation . . . 119
3.2.5 Mapping from a larger distance 2D color code to the stacked code . . . 124
3.2.6 The existence of a threshold? . . . 126
3.2.7 Summary & outlook . . . 127
3.3 Other methods to circumvent magic state distillation . . . 127
3.3.1 Stabilizer state preparation methods . . . 128

3.3.2 Gauge fixing . . . 128
3.3.3 Dimensional jumps . . . 129
3.3.4 Pieceable fault-tolerance . . . 130

4 Private quantum channels **131**
4.1 Privacy vs. quantum error correction . . . 131
4.2 Private quantum subsystems . . . 132
4.2.1 Introduction & background . . . 132
4.2.2 Private subsystems in the absence of private subspaces . . . 134
4.2.3 Testable conditions for private quantum codes . . . 140
4.2.4 Quantum error correction revisited . . . 154
4.2.5 Conclusion & outlook . . . 159

5 Practical quantum architectures: qRAM **161**
5.1 qRAM: converting state to path information . . . 162
5.1.1 Introduction . . . 162
5.1.2 Quantum RAM architectures . . . 165
5.1.3 Error analysis . . . 168
5.1.4 Circuit model . . . 175
5.1.5 Error correction . . . 177
5.1.6 Summary & outlook . . . 180

6 Conclusion **182**

References **185**

APPENDICES **197**

A Conditions on a generalized private quantum channel **198**

B Operator private quantum channels **202**

B.1 Introduction 202

B.2 Private quantum subsystems & subspaces 203

B.3 Quasiorthogonal subalgebras & conditional expectations 205

B.4 Commutation relations of the Pauli group 209

B.5 Privatizing qubits with Abelian subgroups 212

B.6 Summary 214

C A simple decoherence model for bucket brigade qRAM **215**

D Error correction schemes for qRAM **217**

D.1 Correcting simple bit-flip errors 217

D.2 The failure of repetition codes for Regev and Schiff error model 218

D.3 The failure of repetition codes for our error model 219

List of Tables

2.1 Lower bounds for the pseudo and asymptotic threshold results for the Hadamard, T gate and CNOT gates. The Hadamard asymptotic-threshold is larger than its pseudo-threshold resulting from the double protection of the CNOT gates as seen by the high CNOT pseudo-threshold. In bold, the overall thresholds for the 105-qubit and 23-qubit codes are compared. 60

List of Figures

1.1 Protocol for superdense coding [19]. Alice and Bob initially share an entangled pair. Alice performs a Pauli X and Z gate to her half of the entangled pair, dependent on the message she would like to send. She then sends her half of the entangled pair to Bob, who then measures the two qubits in the Bell basis to infer the intended 2-bit classical message. 10

1.2 Protocol for quantum teleportation [18]. Alice and Bob each share one half of a Bell pair. Alice begins by performing a Bell measurement on the qubit she wants to send along with her half of the entangled pair, this is represented by the $CNOT$, H, and Z basis measurements in the first two qubits. She then sends the two measurement results to Bob (2 bits of classical information) upon which he chooses to apply the Pauli X and Z operators to his half of the Bell pair. 11

1.3 Quantum circuit to solve Deutsch's problem with a single query to the black box O_f [49]. 12

1.4 Encoding circuit for Shor's 3-qubit code. An arbitrary quantum state $|\psi\rangle$ is coupled to two ancillary qubits through $CNOT$ gates, yielding a state robust against X errors. 16

2.1 Protocol for quantum gate teleportation [61]. Alice and Bob each share one half of a Bell pair and Alice begins with the arbitrary state used in the computation. Alice performs a Bell measurement on her two qubits, while Bob applies the unitary U to his half of the Bell pair. Alice then sends the result of the measurement to Bob who in turn performs a controlled unitary $UX^aZ^bU^\dagger$ that depends on the measurement result (a, b). The resulting final state on Bob's side is $U|\psi\rangle$. 39

2.2 General construction of a logical gate for a concatenated error correction scheme. The qubit of information is encoded in a quantum error correcting code $\mathcal{C}_1$, whose qubits are in turn encoded into a code $\mathcal{C}_2$. As such, the logical gate g_1 (given by the boxed region) on the encoded space $\mathcal{C}_1$ will be composed of multiple logical gates g_2 on the $\mathcal{C}_2$ codeblocks. 45

2.3 (a) Logical T gate for the Steane code $\mathcal{C}_1$, composed of logical $CNOT_{15}$ and T_{15} gates on the $\mathcal{C}_2$ codeblocks. These gates are transversal in $\mathcal{C}_2$, and therefore only propagate errors to different codeblocks, without propagating within a given $\mathcal{C}_2$ codeblock. (b) Logical H gate for the Steane code $\mathcal{C}_1$, implemented using logical H_{15} gates on each of the $\mathcal{C}_2$ codeblocks. Note that the individual H_{15} are non-transversal on each codeblock. 49

2.4 Steane error correction scheme for detecting (a) X and (b) Z errors. A logical stabilizer state is prepared fault-tolerantly (not shown in Figure) and is coupled to the logical state using a transversal $CNOT$. The transversal $CNOT$ has no effect on the logical state, however errors of the desired type are propagated onto the ancillary state for detection. 52

2.5 Extended rectangle consisting of leading and trailing error correcting circuits implementing the desired logical gate G. 53

2.6 An example of shared EC's between two consecutive level-one exRecs . . . 56

2.7 (a) Polynomials upper bounding the event mal_{IZ} for either the full level-one CNOT exRec or the level-one CNOT exRec with one or both TEC's removed. The polynomial upper bounding the event mal_{IZ} will upper bound all the curves in the above figure. (b) Polynomials upper bounding the level-one CNOT exRec for the different logical error types. 57

2.8 Logical Hadamard H circuit for $[[15, 1, 3]]$ Reed-Muller code. The bold dark lines represent resting qubits subject to storage errors. The dotted vertical lines are used to separate the time steps for which gates are applied in parallel. Logical H for the 105-qubit code is implemented fault-tolerantly by applying each non-fault-tolerant logical H gates in parallel. 58

2.9 (a) Polynomials upper bounding the events mal_X, mal_Z and mal_Y for the level-one Hadamard exRec. Input Z errors are most likely to result in a logical X error on a 15-qubit codeblock which explains why the event mal_X is most likely to occur. (b) Polynomials upper bounding the event mal_Z for the level-one T gate exRec. Note that the logical error probabilities for the event mal_X and mal_Y are too small to be displayed. 59

2.10 Probability of logical error as function of physical error rate for the level-1 and level-2 logical (a) Hadamard and (b) CNOT. The crossing point of the fitted curve allows for the determination of a lower bound for the asymptotic threshold for each of the logical gates. The CNOT gate exhibits a much lower logical error rate than the Hadamard at the first level. 61

2.11 The plots on the left column illustrate the probability of logical error as function of physical error rate for logical (a) CNOT, (c) Hadamard and (e) T gate. The crossing point of the fitted curve allows for the determination of the level-1 pseudo-threshold for each of the logical gates. The CNOT pseudo-threshold is the largest among all three gates due to the double protection of the 7-qubit and 15-qubit code. The plots on the right column illustrate the polynomials upper bounding the probability of obtaining a logical error E for the first, second and third level of concatenation. The crossing point between the level-one and level-two polynomials determine the asymptotic threshold for the gate under consideration. For the logical CNOT gate (b), it is the event mal_{ZI} which limits the threshold value. For the logical gate H (d), mal_X limits the threshold value. Lastly, for the logical T gate (f), mal_Z limits the threshold value. 62

3.1 The surface code with distance $d = 6$. (a) Layout of the surface code with qubits on the edges. Z stabilizers are defined by square plaquette operators, shown in Blue. X stabilizers are defined by star operators, shown in Red. (b) Logical operators for the surface code, Z_L in blue and X_L in red. The Z_L logical operator connects rough boundaries on left and right while the X_L logical operators connect the smooth boundaries on top and bottom. . 100

3.2 2D hexagonal color code of distance $d = 5$. Two instances of equivalent logical operators are presented. Logical operators are formed by connecting string-like excitations to all three colored boundaries, fusing within the bulk at the white qubit. In the case of subfigure (a), the logical operator lives on the purple boundary and is trivially connected to that boundary (fusion could be defined at any point in the chain). (b) Logical operators are formed by connecting the three boundary points in the bulk. 101

3.3 Instance of the $[[15, 1, 3]]$ Reed-Muller code as a 3D color code. 103

3.4 Two instances of the 2D hexagonal color code of distance (a) $d = 3$ and (b) $d = 5$. In each case, a set of independent edges $\{H_{e_i}\}$, shown in red, can be chosen as the set that will form the Z gauge operators when paired with the identical edge from another code copy, thus forming weight-4 gauge operators $\{H_{e_i}^{(2k-1)} H_{e_i}^{(2k)}\}$. 106

3.5 (a) Graphical representation of the primal lattice of the $(d-1)+1$ stacked code formed by stacking different copies of 2D color codes, shown here for $d = 5$. The copies of the 2D code are coupled either by measuring gauge operators or logical operator pairs (shown in blue) between the different layers. (b) Dual lattice for the 3D stacked code ($d = 5$). Vertices represent cell stabilizers in the primal lattice and edges represent faces shared by connected stabilizers. 113

3.6 Examples of the different representations of equivalent logical error strings that exist in the 3D stacked code. The color of the logical strings are chosen according to the color of the edges they follow. The curved lines represent joining of edges through a stabilizer of the same color. In (a), because the string lies on the green–yellow boundary, it can be chosen to be either of the complementary colors, blue or purple. In (b), the error string connects the bottom blue boundary to the joint boundary of the other three colors at the ancilla qubit, following blue edges. Example (c) shows how multiple colored boundaries can fuse in the bulk, thus negating the excitation that would otherwise be present. 115

3.7 A 2D layout for the implementation of the stacked code ($d = 5$ shown). Pairs of copies of the 2D hexagonal color code are layered on top of one another in a single 2D layer, in such a way as to keep the gauge operators geometrically local. (a) Initial layout of the stacked code transformation in 2D. The 2D layers $(2k)$ and $(2k+1)$ are coupled by measuring joint logical X and Z operators (Bell stabilizers), with supporting qubits shown in blue. Although Bell stabilizers for the stacked code are high-weight, involving all blue qubits, the only required measurements are those associated with local 2D stabilizer/gauge operators together with one-dimensional operators of weight $\mathcal{O}(d)$ (shaded blue). The only 2D plane that is not initially coupled to another layer (or ancilla qubit) is the bottom $k = 1$ layer, which stores the encoded qubit. (b) Measurement of the weight-4 Z-type gauge operators, shown in Red. X-type stabilizers from individual layers are combined to form cell-like stabilizers by stabilizer evolution. Original joint logical X measurements, given by Blue shaded region, are mapped to all Blue qubits. 119

3.8 A two-dimensional layout of the construction presented in Fig. 3.7. The two originally superimposed lattices have respective grey and white lattice qubits. Only one of the color code stabilizers (per pair) have been colored, for clarity. Gauge measurement operators are given by red faces. Here, we have identified three individual gauge measurements per pair of codes for clarity, there are actually $3(d^2-1)/8$ such gauge measurements for each pair of distance d codes. 122

3.9 Initial coupling of split regions of a 2D color code. The original code is split into multiple color code copies by turning off and modifying certain stabilizer measurements. Different patches are coupled to form a Bell pair by measuring joint logical X and Z stabilizers between them, shown in blue. The patch that is not coupled in this way retains the quantum information that was originally stored in the code. The different patches are then further joined together by measuring gauge operators by matching up weight-2 edges from the different patches (forming weight-4 gauge operators), shown by red and cyan edges. 124

4.1 The gates in the box region implement the encoding of an arbitrary two-qubit state belonging to the $I \otimes \mathbb{C}^{2\times 2}$ algebra into encoded states of the form of Eq. (4.17). The encoded two-qubit state subjected to the two-qubit phase damping channel $\Lambda = \Lambda_2 \circ \Lambda_1$ results in an output of the completely mixed state on two qubits, $\rho_0 = \frac{1}{4} II$. 139

4.2 Isometric extension of the two-qubit dephasing channel $\Lambda = \Lambda_2 \circ \Lambda_1$. The extension of the channel to be a unitary transformation is given in the solid box (—) by introducing ancilla qubits 4 and 5. The dashed box (– –) contains the encoding of ρ of the form in equation (4.17) into the algebra $I_2 \otimes \mathbb{M}_2$. The dotted box $(\cdots)$ gives a particular preparation of the mixed state for the subsystem encoding, using a "mixing ancilla" that is traced out for both the channel Λ and its complementary channel $\Lambda^\sharp$. 148

4.3 Generalized form of extending a private quantum subsystem channel to a unitary transformation via Stinespring's dilation theorem. The subsystem $\sigma_A \otimes \sigma_B$ that encodes the arbitrary state of quantum information σ_B is prepared by entangling an ancillary pure state mixing ancilla $|\Theta\rangle\langle\Theta|_M$ with a chosen pure state $|\varphi\rangle\langle\varphi|_A$ via the unitary U_{MA} and tracing out over the mixing ancilla space M. This operation corresponds to the dotted box in Figure 4.2. The action of the private quantum channel Φ can also be extended to a unitary transformation over a larger Hilbert space, as described by the action of U_Φ on systems ABK by introducing the ancillary state $|\zeta\rangle\langle\zeta|_K$, as described in Eq. 4.20. The unitary transformation U_Φ corresponds to the dashed and solid boxes in Figure 4.2. The complementary channel is defined on the output space of the extension of the channel Φ, and therefore corresponds to the final state on system K, yet in general will not be quantum error correctable for an arbitrary subsystem channel. However, a generalized conjugate channel $\tilde{\Phi}$ can be constructed on the Hilbert space MK, as described in Eq. 4.21, which will necessarily be a quantum error correctable channel since the overall extension is a subspace channel. 151

5.1 Representation of the energy levels of a qutrit used at the nodes of the routing binary tree. The states $|0\rangle$ and $|1\rangle$ form a metastable subspace since the energy difference between the states is required to be much smaller than the difference between the ground state $|\bullet\rangle$ and $|0\rangle$. 166

5.2 Bucket brigade scheme for a qRAM with 8 memory locations. The address register is $|010\rangle$, corresponding to the memory location m_{010}. The path $0 \to 1 \to 0$ is established by sequentially introducing the address qubits $|010\rangle$ into the root of the binary tree. 167

5.3 Example of a *wrong path* produced by an error at the third time step, given the address $|010\rangle$. 170

5.4 Example of a *wrong path* produced by an error at the second time step, given the address $|010\rangle$. 170

5.5 Example of a *no-path* given the address $|010\rangle$ 171

5.6 Comparison of errors for fixed ε as a function of n. 173

5.7 Comparison of errors for fixed n as a function of ε. 173

5.8 Required ε (in dimensionless units of 10^{-2}) as a function of n, for a fixed circuit fidelity. GLM denotes the model proposed in [56]. 174

5.9 Circuit for bucket brigade qRAM. Nodes to the left of the memory cell are *routing nodes*. The dashed squares represents the memory locations. The first layer of nodes immediately to the right of the memory are the *coupling nodes*. Finally, the nodes on the right are the *read out nodes*. A possible input is e.g. $|a_0a_1a_2\rangle = |010\rangle$, for which the circuit reads the memory location m_{010}. The path leading to the location m_{010} is represented in blue colour, and the active routing and readout nodes are highlighted. One could more closely mimic the physical flow of information in the bucket brigade qRAM by adding an additional qubit at each node in the binary tree we see in the diagram. Then, for $k \in \{0, \ldots, n-1\}$, we add an initial controlled-NOT gate to copy a_k to the root node, followed by a series of $\mathcal{O}(2^k)$ controlled-SWAPs that will bring the value of a_k to the unique node in level k defined by the bits $a_0, a_1, \ldots, a_{k-1}$. While this adds exponentially many gates, it does not change the overall gate complexity, and these additional gates only add $\mathcal{O}(k)$ to the depth of the circuit. This also illustrates that the exponential depth implicit in the circuit we describe in the diagram can easily be reduced to polynomial depth by further mimicking the ideas presented in the qRAM proposal. We leave the circuit diagram in this simpler form, since it does not affect our arguments in Subsections 5.1.3 and 5.1.5. 181

Chapter 1

Quantum mechanics: a new realm for computation

1.1 Research motivation

The discovery of quantum mechanical phenomena and the axiomatization of its principles has been at the core of physics for the last century. In parallel, the development of computational devices has proved to be instrumental in modern society, revolutionizing everyday life and enhancing the scientific questions that modern research aims to answer. As researchers continue to push the frontiers of all areas of science, a new player that implements quantum states as the basis for storing and manipulating information has emerged: quantum computers.

Historically, quantum computers were proposed as a means to simulate quantum systems that were thought to be intractable for traditional computing devices. However, quantum algorithms have since been proposed that could push frontiers in many different directions. Paul Benioff proposed the quantum Hamiltonian model as a means to simulate the Turing machine model [15]. Richard Feynman proposed the idea of simulating complex quantum systems [54, 99], Bennett and Brassard developed a method for quantum key distribution [17] laying the foundations for quantum cryptography, and Peter Shor discovered an algorithm to efficiently factor a number into its prime roots [125]. All of these examples point to an exciting future for quantum computing and information research and open a realm of unexplored techniques and algorithms that may further the advantages of using quantum computers instead of their classical counterparts.

Classical algorithms manipulate information stored in binary systems, such as transistors, where a state is in one of two possible states at a given time. Alternatively, most quantum algorithms use systems that can be in a superposition of states, allowing for a form of parallel computing that is not accessible to classical devices. Moreover, quantum systems allow for correlations that are stronger than those present in classical systems, defined as entangled systems. Many experimental efforts have been aimed to show the presence of these two phenomena in different physical systems [42, 105, 44, 67, 108, 135, 103, 50]. The progress in these areas has been monumental and has arrived at the stage where the question of scalability has come to the forefront of experimental research.

Due to the high sensitivity of quantum systems to external parameters, quantum states will be susceptible to errors. Quantum error correction is the theory behind the suppression of noise for quantum systems. Although experimental systems have continuously improved, demonstrating tremendous progress in the control and shielding of quantum systems to external noise, there will reach a point at which relying only on technological improvements will be unrealistic. Namely, in order to implement complex quantum algorithms that could have on the order of 10^{15} operations, by using the bare physical systems themselves as the means for storing and manipulating the quantum information would necessitate noise levels to be suppressed on the order of magnitude of 10^{-15}. In order to combat this continuous need to experimentally improve precision[1], the quantum information is encoded in multiple quantum systems by using entanglement, and in doing so the noise can be suppressed when considering its action on the logical qubit.

The holy grail for noise suppression is a fault-tolerant quantum computer. That is, a quantum computing device in which the logical qubits can be stored and manipulated, with the error rate being suppressed to arbitrarily small levels, as well as reasonable overhead of quantum computing resources. Such a device is plausible due to the quantum fault-tolerance threshold theorem. The theorem states that if the physical error rate of the fundamental operations needed for a quantum computing device is below a certain threshold value, then suppression of the noise is achievable with realistic overhead[2]. While the threshold theorem was established in the early stages of theoretical quantum computing research, the experimental systems that have the potential for scalability have recently began to exhibit the level of control to approach and even surpass the fault-tolerance threshold for certain fault-tolerant architectures [50, 12, 40, 83, 134]. As such, it has become of prime importance to understand which quantum computing architectures exhibit the smallest

[1] We shall define the specific meaning of an error rate of 10^{-15} later in this work when we expand on error correction as a whole, yet for now you can take it for granted that this is very challenging.

[2] In reality, the fault-tolerance threshold theorem allows for a poly-logarithmic overhead in the inverse of the target error rate in order to suppress logical noise.

possible resource overhead, as well as the architectural requirements that a particular error correcting code will impose on the underlying physical device.

This thesis aims to further develop the theory of quantum error correcting codes and fault-tolerant architectures as well as to provide potential practical alternatives to traditional fault-tolerant architectures. The research presented in this thesis has contributed to a new class of fault-tolerant models which avoid bottlenecks such as magic state distillation that yield a major road block to current proposed architectures. Additionally this thesis addresses the limits of error correction and fault-tolerance by classifying a particularly prominent class of fault-tolerant operations. Moreover, the connection of quantum error correction to privacy and protection of quantum information is explored, and a characterization of the set of operations that can privatize information is proposed. Finally, a study of the practicality of quantum Random Access Memories is discussed in the context of one of the fundamental quantum algorithms, Grover's algorithm.

The remainder of this chapter will be devoted to reviewing the foundations of quantum mechanics and information, highlighting notions that will reappear throughout the remainder of the thesis. We shall begin by reviewing the structure of the qubit and the roles of superposition and entanglement before presenting the core ideas behind quantum error correction and fault-tolerance. There are many good textbooks and academic notes that provide excellent reading on these subjects [82, 107, 53, 120, 136].

1.2 The fundamentals of quantum information

1.2.1 From bits to qubits

1.2.1.1 The qubit

In traditional computers, information is encoded in bits, that is, in sequences of 0s and 1s. Therefore, for an n-bit string s, a total of 2^n possible states (or messages) can be encoded, that is $s \in \{0,1\}^n$.

The fundamental unit of information in a quantum computer is the *qubit*, which like its classical counterpart is defined in a two-state system. A quantum state is defined according to its expectation values of Pauli observables. We will come to axiomatizing the notion of measurement and observables later in this section, but for now it will be sufficient to

define the Pauli observables as follows:

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{1.1}$$

A general quantum state ρ is defined as:

$$\rho = \frac{1}{2}(I + \sigma_X X + \sigma_Y Y + \sigma_Z Z), \tag{1.2}$$

where $\alpha_P \in \mathbb{R} \cap [-1, 1]$ such that $\sigma_X^2 + \sigma_Y^2 + \sigma_Z^2 \leq 1$. Qubits are elements of a 2-dimensional complex Hilbert space $\mathcal{H} = \mathbb{C}_2$. Note that since the Pauli matrices P are traceless, the trace of a quantum state is always 1, $\mathrm{Tr}\,(\rho) = 1$.

A single-qubit *pure state* is a quantum state such that $\sigma_X^2 + \sigma_Y^2 + \sigma_Z^2 = 1$. A pure state can always be expressed as the outer product of a complex vector $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, that is $\rho = |\psi\rangle\langle\psi|$, where $\alpha,\ \beta \in \mathbb{C}$. This statement can be affirmed by choosing α and β such that:

$$\sigma_Z = |\alpha|^2 - |\beta|^2, \tag{1.3}$$
$$\sigma_X = \alpha\beta^* + \alpha^*\beta, \tag{1.4}$$
$$\sigma_Y = i\alpha\beta^* - i\alpha^*\beta. \tag{1.5}$$

We define composite quantum systems composed of multiple qubits by taking the tensor product of the respective Hilbert spaces of each of the qubit systems. The set of quantum states $\rho \in \mathcal{H}^{\otimes n} = \mathbb{C}_2^{\otimes n}$ is the set of density matrices which is the set of completely positive matrices with trace 1. The Pauli group on n qubits is the set of all possible tensor products of Pauli operators, denoted $\mathcal{P}_n$. In a similar manner to that of the single qubit system, a pure state on a multi-qubit system is a state that can be expressed as $\rho = |\psi\rangle\langle\psi|$, where $|\psi\rangle$ is a normalized complex vector $|\psi\rangle \in \mathbb{C}_2^n$. A necessary and sufficient condition for a state ρ to be pure is $\mathrm{Tr}\,(\rho^2) = 1$. Any state that is not pure is defined to be *mixed*, and any such mixed state can be expressed as a probabilistic combination of pure states, that is $\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$, where $\sum_i p_i = 1$.

It should be noted that the tensor product of a set of n quantum states ρ_i is a valid quantum state since $\mathrm{Tr}\,(\rho_1 \otimes \cdots \otimes \rho_n) = \mathrm{Tr}\,(\rho_1) \cdot \mathrm{Tr}\,(\rho_2) \cdots \mathrm{Tr}\,(\rho_n) = 1$, by the multiplicative property of the trace function. A quantum system B is a *subsystem* of $\mathcal{H}$ if we can write $\mathcal{H} = (A \otimes B) \oplus (A \otimes B)^{\perp}$. This definition is symmetric in that A is also considered a subsystem of S. The sub*spaces* of $\mathcal{H}$ can be viewed as subsystems B for which A is one-dimensional.

Definition 1. *A quantum state $\rho \in A \otimes B$, where A and B are subsystems, is a* ***separable state*** *if and only if it can be expressed as follows:*

$$\rho = \sum_i p_i \eta_i, \tag{1.6}$$

where $\eta_i = \rho_{i,A} \otimes \rho_{i,B}$ and $\sum_i p_i = 1$. All states ρ that are not separable are defined as being ***entangled****.*

The simplest form of entangled states are the Bell states, $\rho = |B_i\rangle\langle B_i|$, where:

$$|B_0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}(|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle), \tag{1.7}$$

$$|B_1\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle), \tag{1.8}$$

$$|B_2\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), \tag{1.9}$$

$$|B_3\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle). \tag{1.10}$$

$$\tag{1.11}$$

1.2.1.2 Measurement and evolution

We shall define observables O as Hermitian operators in finite-dimensional Hilbert spaces. Typically for infinite-dimensional systems, quantum observables are defined as self-adjoint operators. In this thesis, we will be focusing on finite-dimensional systems where self-adjoint operators are equivalent to Hermitian operators.

Definition 2. *Given a finite-dimensional Hilbert space $\mathcal{H}$, an* ***observable*** *$O \in \mathcal{H}$ is a Hermitian operator with spectral decomposition $O = \sum \lambda_i \Pi_i$, where Π_i are orthogonal projectors that have the property $\Pi_i \Pi_j = \delta_{ij} \Pi_i$ and $\Pi_i = \Pi_i^\dagger$. The* ***measurement*** *of O given a state of the system $\rho \in \mathcal{H}$ will produce outcome λ_i with probability $\mathrm{Tr}\,(\Pi_i \rho)$, the resulting state of the system after measurement is $\dfrac{\Pi_i \rho}{\mathrm{Tr}\,(\Pi_i \rho)}$.*

The measurement axiom has two fundamental components: each eigenvalue of O corresponds to a measurable outcome and the probability of such an outcome is given

by $\mathrm{Tr}\left(\Pi_i \rho\right)$, which is often referred to as **Born's rule**. We define the expectation value of an observable as the average value of the measurement outcome, that is:

$$\langle O \rangle = \sum_i p_i \lambda_i = \sum_i \mathrm{Tr}\left(\Pi_i \rho\right) \lambda_i = \mathrm{Tr}\left(\left(\sum_i \lambda_i \Pi_i\right) \rho\right) = \mathrm{Tr}\left(O\rho\right). \tag{1.12}$$

Suppose closed state evolution is given by an operator U. Let $|\psi_1\rangle$ and $|\psi_2\rangle$ be the initial and final normalized states, respectively. Consider then the following:

$$\langle \psi_2 | \psi_2 \rangle = \langle \psi_1 | U^\dagger U | \psi_1 \rangle = 1 = \langle \psi_1 | \psi_1 \rangle, \tag{1.13}$$

since this must hold for all states $|\psi_1\rangle$ and $|\psi_2\rangle$ the evolution operator U must satisfy $U^\dagger U = I$, that is U must be unitary. We therefore restrict all closed-system state evolution processes as unitary processes. A particular example of such an evolution is given by the time-independent Schrödinger equation:

$$i\hbar \frac{\mathrm{d}}{\mathrm{dt}} |\psi\rangle = H|\psi\rangle, \tag{1.14}$$

where H is a Hermitian operator. The solution to this equation for time t is given by: $|\psi(t)\rangle = e^{-iHt}|\psi\rangle = U|\psi\rangle$.

The restriction of closed evolution being a unitary process has an interesting consequence for the ability to copy arbitrary quantum states.

Theorem 3 ([139])**.** *Let $\mathcal{H}$ be a Hilbert space of dimension greater than 1. Then there exists no unitary operation U and ancilla state $|a\rangle$ such that:*

$$U|\psi\rangle|a\rangle = |\psi\rangle|\psi\rangle, \ \forall \ |\psi\rangle \in \mathcal{H}. \tag{1.15}$$

Proof. Suppose not. Let $|\psi\rangle$ and $|\phi\rangle$ be two non-orthogonal states such that $\langle\psi|\phi\rangle \in \mathbb{R}$ and $0 < \langle\psi|\phi\rangle < 1$. Then,

$$\begin{aligned} \langle\psi|\phi\rangle &= (\langle\psi| \otimes \langle a|)(|\phi\rangle \otimes |a\rangle) \\ &= (\langle\psi| \otimes \langle a|)U^\dagger U(|\phi\rangle \otimes |a\rangle) \\ &= (\langle\psi| \otimes \langle\psi|)(|\phi\rangle \otimes |\phi\rangle) = \langle\psi|\phi\rangle\langle\psi|\phi\rangle \\ \Rightarrow \langle\psi|\phi\rangle &= \langle\psi|\phi\rangle^2. \end{aligned}$$

The last statement is a contradiction to the fact that $0 < \langle\psi|\phi\rangle < 1$, implying no such unitary exists. □

1.2.1.3 Stinespring dilation and the unknown

Quantum systems can be very complex, spanning many degrees of freedom in the universe, not all of which we may have access to in the form of observables. Suppose we have a quantum state on the joint Hilbert space AB, that is $\rho_{AB} \in \mathcal{H}_{AB}$. It is important to establish a consistent definition for the restricted state ρ_A on a single subsystem, that is we want the observable O_A to produce the same measurement result independent of which description we were to choose. We require that for any state ρ_{AB}, ρ_A is a state such that:

$$\mathrm{Tr}\left(O_A \rho_A\right) = \langle O_A \rangle = \langle (O_A \otimes I_B) \rangle = \mathrm{Tr}\left((O_A \otimes I_B)\rho_{AB}\right). \tag{1.16}$$

The only consistent definition for the state ρ_A is taking the partial trace of the state ρ_{AB}.

Definition 4. *Given a state $\rho_{AB} \in \mathcal{H}_{AB}$, the state ρ_A of the restricted subsystem $\mathcal{H}_A$ is given by the **partial trace** of the subsystem $\mathcal{H}_B$:*

$$\rho_A \equiv Tr_B\left(\rho_{AB}\right) \equiv \sum_b \langle u_b | \rho_{AB} | u_b \rangle, \tag{1.17}$$

where $\{|u_b\rangle\}$ is an orthonormal basis for the subsystem $\mathcal{H}_B$.

In Subsection 1.2.1.2 we discussed two different transformations of quantum states. The first was the measurement of a quantum observable, whose final state was in the projected subspace of the appropriate eigenvalue. Secondly, we saw that time-evolution from one pure state to another is performed via a unitary transformation. These are both examples of quantum channels.

Definition 5. *A quantum channel $\mathcal{N} : \mathcal{H}_A \to \mathcal{H}_B$ is a completely positive, trace-preserving map on linear operators.*

The fact that it is trace-preserving means that it maps valid density matrices to density matrices, and thus quantum states to quantum states (quantum states are always positive and as such the channel is required to be completely positive). A quantum channel can always be expressed in its **Kraus representation** and moreover such a representation is unique up to unitary equivalences.

Lemma 6. *Let $\mathcal{N} : \mathcal{H}_A \to \mathcal{H}_B$ be a quantum channel. Then there exists a finite collection of operators $\{A_k\}$ such that:*

$$\mathcal{N}(\rho_A) = \sum_k A_k \rho_A A_k^\dagger, \ \forall \ \rho_A \in \mathcal{H}_{\mathcal{A}}. \tag{1.18}$$

Moreover, if there is another set of operators $\{B_l\}$ such that (1.18) holds, then $B_l = \sum_k U_{lk} A_k$, where u_{lk} are entries from a unitary matrix U.

Example 7. *Let $\mathcal{H} = \mathcal{H}_2^{\otimes n}$ be the Hilbert space associated with n qubits. Let $\mathcal{P}_n$ be the Pauli group on $\mathcal{H}$, then the* ***depolarizing channel*** *$\mathcal{D}_p$ with strength p is defined as:*

$$\mathcal{D}_p(\rho) = (1 - p + \frac{p}{4^n})\rho + \frac{p}{4^n} \sum_{P_i \in \mathcal{P}_n \setminus I} P_i \rho P_i. \tag{1.19}$$

Example 8. *Let $\mathcal{H}_2$ be a qubit space. Then the* ***phase damping*** *channel with strength p is defined as:*

$$\Phi(\rho) = (1 - \frac{p}{2})\rho + \frac{p}{2} Z \rho Z. \tag{1.20}$$

Example 9. *Let $\mathcal{H}_2$ be a qubit space. Then the* ***amplitude damping*** *channel with strength p is defined as:*

$$\mathcal{A}(\rho) = A_0 \rho A_0^\dagger + A_1 \rho A_1^\dagger, \tag{1.21}$$

where $A_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-p} \end{pmatrix}$, $A_1 = \begin{pmatrix} 0 & \sqrt{p} \\ 0 & 0 \end{pmatrix}$.

One of the most important insights in the mathematical construction of quantum mechanics is that any quantum channel can be expressed as a unitary transformation on a higher-dimensional system followed by the partial trace of part of the system.

Theorem 10 ([132])**.** *Given a quantum channel $\mathcal{N} : \mathcal{H}_A \to \mathcal{H}_B$, there exist ancillary subsystems $\mathcal{H}_{A'}$, $\mathcal{H}_E$, and a unitary $U_\mathcal{N} : \mathcal{H}_A \otimes \mathcal{H}_{A'} \to \mathcal{H}_B \otimes \mathcal{H}_E$, such that*

$$\mathcal{N}(\rho) = \mathrm{Tr}_E \left(U_\mathcal{N}(\rho \otimes |0\rangle\langle 0|_{A'}) U_\mathcal{N}^\dagger \right), \ \forall \rho \in \mathcal{H}_A. \tag{1.22}$$

The notion that any quantum channel can be expressed as a unitary transformation on a higher-dimensional Hilbert space, and moreover that such a unitary is unique (up to choice of ancillary input state), leads to the notion of a complementary channel.

Definition 11. *Given a quantum channel $\mathcal{N} : \mathcal{H}_A \to \mathcal{H}_B$ and a unitary dilation $U_\mathcal{N} : \mathcal{H}_A \otimes \mathcal{H}_{A'} \to \mathcal{H}_B \otimes \mathcal{H}_E$, then the* ***complementary channel*** *$\mathcal{N}^\sharp : \mathcal{H}_A \to \mathcal{H}_E$ is given by:*

$$\mathcal{N}^\sharp(\rho) = \mathrm{Tr}_B \left(U_\mathcal{N}(\rho \otimes |0\rangle\langle 0|_{A'}) U_\mathcal{N}^\dagger \right), \ \forall \rho \in \mathcal{H}_A. \tag{1.23}$$

The complementary channel can be thought of as the quantum information that is leaking to the environment given the action of the quantum channel $\mathcal{N}$. We will return to this in much more detail in Chapter 4.

1.2.1.4 Entanglement and locality

In the next few subsections we review important properties of entangled states and how they differ from their classical counterparts. The key difference is that an entangled state can only be prepared *jointly*, that is through some joint quantum operation on the set of qubits to be entangled. Therefore, suppose Alice and Bob are two distance parties, each with a qubit (initially non-entangled), then there exists no protocol by which they can entangle their qubits using only local quantum operations along with classical communication (LOCC). However, that is not to say that correlated states cannot be created using LOCC. Suppose Alice and Bob each start with their respective states $|0\rangle$, and Alice flipped an unbiased coin and changes the state of her system accordingly by applying Pauli X. Then, through classical communication, she relays the result of her flip to Bob, at which point he does the same. Without knowledge the the result of the coin flip, their joint state can thus be expressed as:

$$\frac{1}{2}(|00\rangle\langle 00| + |11\rangle\langle 11|), \tag{1.24}$$

which is a correlated state yet remains separable.

1.2.1.5 Superdense coding

Entanglement can be used in various ways and plays a central role in quantum algorithms. In this subsection we discuss a method to increase the rate of classical communication using an entangled state between two parties, Alice and Bob [19]. Suppose Alice and Bob share an entangled state and are spatially separated. They have access to a quantum channel and thus the ability to share qubits, but would like to minimize their (perhaps costly) communication. Alice's goal then becomes to send classical information to Bob with minimal uses of the quantum channel.

Of course, without using the entangled state, Alice could send a classical bit of information to Bob encoded in the form of a quantum state. However, Alice would be limited to only sending one classical bit of information this way. By using her side of the entangled state, Alice can encode two classical bits of information by manipulating the entangled Bell state. Suppose Alice and Bob begin with the Bell state $|B_0\rangle = (|00\rangle + |11\rangle)_{AB}/\sqrt{2}$, then Alice can encode the classical message (ab) by first applying Pauli X if $a = 1$ and then

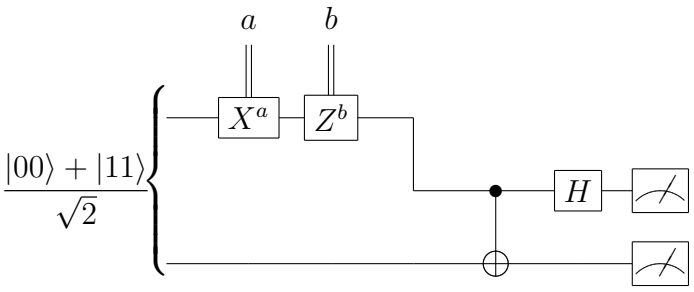


Figure 1.1: Protocol for superdense coding [19]. Alice and Bob initially share an entangled pair. Alice performs a Pauli X and Z gate to her half of the entangled pair, dependent on the message she would like to send. She then sends her half of the entangled pair to Bob, who then measures the two qubits in the Bell basis to infer the intended 2-bit classical message.

applying Pauli Z if $b = 1$, resulting in the following transformation:

Classical Message	Final state	
(00)	$\|B_0\rangle = (\|00\rangle + \|11\rangle)/\sqrt{2}$	(1.25)
(01)	$\|B_1\rangle = (\|00\rangle - \|11\rangle)/\sqrt{2}$	(1.26)
(10)	$\|B_2\rangle = (\|01\rangle + \|10\rangle)/\sqrt{2}$	(1.27)
(11)	$\|B_3\rangle = (\|01\rangle - \|10\rangle)/\sqrt{2}.$	(1.28)

Since these are all orthogonal states, there exists a two-qubit observable that can distinguish between these four states. As such, if Alice sends her state to Bob, he can perform a joint measurement in the entangled basis to infer the 2-bit message that Alice intended to communicate. A graphical representation of the scheme is presented in Figure 1.1.

Entanglement plays an essential role in this protocol as Bob will need to distinguish between four possible outcomes upon receiving the qubit sent by Alice (equivalent to 2 bits). However, if the state are not entangled, then no matter the transformation Alice performs on her state, Bob will only be able to gather 1 bit of information as his state will have no bearing on any information that can be extracted from Alice's state via measurement.

1.2.1.6 Quantum state teleportation

Entanglement can also be a useful resource for the communication of quantum states between two distant parties that do not have access to a quantum channel but are restricted to classical communication. Alice can send Bob an arbitrary quantum state by coupling

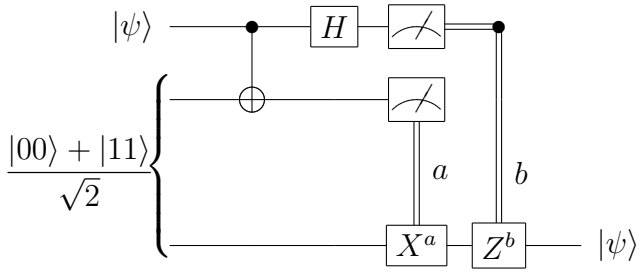


Figure 1.2: Protocol for quantum teleportation [18]. Alice and Bob each share one half of a Bell pair. Alice begins by performing a Bell measurement on the qubit she wants to send along with her half of the entangled pair, this is represented by the $CNOT$, H, and Z basis measurements in the first two qubits. She then sends the two measurement results to Bob (2 bits of classical information) upon which he chooses to apply the Pauli X and Z operators to his half of the Bell pair.

her qubit to her half of the entangled state through a quantum measurement and communicating to Bob the classical result of the quantum measurement [18].

Suppose Alice and Bob share an entangled Bell pair $|B_0\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$ and Alice would like to send Bob the arbitrary state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. Alice begins the protocol by measuring the qubits at her disposal in a Bell measurement, that is using the observable: $O = \sum_{ij} c_{ij}|B_{ij}\rangle\langle B_{ij}|$. The resulting state of Alice and Bob's systems, according to the appropriate measurement result that can be obtained by normalizing the projected state $(|B_{ij}\rangle\langle B_{ij}|_A \otimes I_B)(\alpha|0\rangle + \beta|1\rangle)_A(|00\rangle + |11\rangle)_{AB}/\sqrt{2}$, is:

Measurement outcome	Resulting state	
c_{00}	$\|B_0\rangle_A \otimes (\alpha\|0\rangle + \beta\|1\rangle)_B$	(1.29)
c_{01}	$\|B_1\rangle_A \otimes (\alpha\|0\rangle - \beta\|1\rangle)_B$	(1.30)
c_{10}	$\|B_2\rangle_A \otimes (\alpha\|1\rangle + \beta\|0\rangle)_B$	(1.31)
c_{11}	$\|B_3\rangle_A \otimes (\alpha\|1\rangle - \beta\|0\rangle)_B.$	(1.32)

Therefore, by performing a Bell measurement, Alice has mapped Bob's half of the entangled Bell pair to one of 4 states that differ from Alice's original state by the application of Pauli operators. Moreover, Alice will know which state Bob will have, and as such can communicate the recovery operation by sending two classical bits of information. Alice sends the message (ab) and Bob applies Pauli X if $a = 1$ followed by Pauli Z if $b = 1$. Bob's final state is then the original state Alice wanted to send and no quantum channel was needed! Just as in the case of superdense coding, this would be impossible without the initial shared entanglement between the two parties.

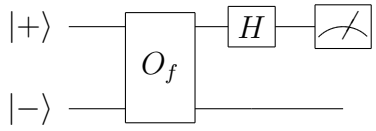


Figure 1.3: Quantum circuit to solve Deutsch's problem with a single query to the black box O_f [49].

1.2.2 Superposition in quantum algorithms

Quantum superposition and entanglement play an essential role in the development of quantum algorithms. Deutsch's problem is one of the first algorithms that uses superposition as a key ingredient [49]. The premise of the problem is to query a black box and determine certain properties of the box given some promise. The quantum black box model is stated in a reversible manner as per the description of quantum mechanics as unitary evolution. Given a binary input space $\mathcal{X}$ and output space $\mathcal{Y}$, the black box oracle is defined as:

$$O_f|x\rangle|a\rangle = |x\rangle|a \oplus f(x)\rangle, \ \forall\ x \in \mathcal{X}. \tag{1.33}$$

1.2.2.1 Deutsch's problem

Deutsch's problem asks, given a function $f : \{0,1\} \to \{0,1\}$, to compute $f(0) \oplus f(1)$. Given a classical black box, one would have no choice but to query the function for both inputs, and then compute the result, therefore requiring two queries to the oracle. However, in the quantum setting we can use superposition to compute the desired result only using a *single* query to the quantum black box [49]. Consider the circuit in Fig. 1.3, it will result in the following evolution of the input states:

$$\begin{aligned}
|+\rangle|-\rangle =& \frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle) && (1.34)\\
\longrightarrow& \frac{1}{2}\left(|0\rangle(|f(0)\rangle - |1 \oplus f(0)\rangle) + |1\rangle(|f(1)\rangle - |1 \oplus f(1)\rangle)\right) && (1.35)\\
\longrightarrow& \frac{1}{2}((-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle)(|0\rangle - |1\rangle) && (1.36)\\
=& \frac{(-1)^{f(0)}}{2}(|0\rangle + (-1)^{f(0)+f(1)}|1\rangle)(|0\rangle - |1\rangle) && (1.37)\\
\longrightarrow& (-1)^{f(0)}|f(0) \oplus f(1)\rangle|-\rangle, && (1.38)
\end{aligned}$$

therefore the measurement of the first qubit at the output of the circuit will yield the desired solution to Deutsch's problem (the global phase $(-1)^{f(0)}$ has no bearing on the state or solution). Notice the interesting feature that the oracle is set up to yield phase information on the first qubit which can be extracted using a Hadamard gate. This is a common feature in quantum algorithms and will play an integral role in the upcoming examples.

1.2.2.2 Grover's algorithm

Deutsch's problem provided an example of quantum information processing yielding a strict speedup over its classical counterpart. How far can these speedups be extended? One example is in the problem of searching for a marked item in a list. Suppose there is a list of $N = 2^n$ bits, one of which is 1 while all others are 0. The problem is to minimize the number of calls to a black box in order to query the values of all the items. It has been shown that in order to succeed with probability at least 2/3, it will take $\mathcal{O}(2^n)$ queries in the case of a classical oracle. However, one can identify the marked state using only $\mathcal{O}(\sqrt{N}) = \mathcal{O}(2^{n/2})$ calls to the quantum black box [62], and this was subsequently shown to be optimal scaling for the number of oracle calls in terms of N [16].

First, we can use an equivalent description of the black box following the procedure used for Deutsch's problem:

$$O_f|x\rangle|-\rangle = \frac{1}{\sqrt{2}}|x\rangle(|f(x)\rangle - |f(1)\rangle) \tag{1.39}$$

$$= \frac{(-1)^{f(x)}}{\sqrt{2}}|x\rangle(|0\rangle - |1\rangle) \tag{1.40}$$

$$= (-1)^{f(x)}|x\rangle|-\rangle, \tag{1.41}$$

therefore we can think of the oracle as the transformation $O_f|x\rangle = (-1)^{f(x)}|x\rangle$. For now suppose such a black box can be prepared error-free, Chapter 5 will focus on the implementation of such an oracle in a fault-tolerant setting. There is one more element needed in the implementation of Grover's algorithm, and that is the reflection operator $U_r = 2|0^n\rangle\langle 0^n| - \sum_{x\in\mathcal{X}} |x\rangle\langle x|$, which implements a phase of (-1) on all inputs except for the all-0 string (this could be implemented using a high-controlled phase operator). Note that the reflection operator acts independently of the oracle, and can be thought of as a unitary implemented on the quantum computing device. The Grover iterate is then defined as the sequence $G = H^{\otimes n}U_r H^{\otimes n}U_f$.

Consider the action of the Grover iterate G on the superposition over all input states:

$$G\left(\frac{1}{2^{n/2}}\sum_{x\in\mathcal{X}}|x\rangle\right) = H^{\otimes n}U_r H^{\otimes n}U_f\left(\frac{1}{2^{n/2}}\sum_{x\in\mathcal{X}}|x\rangle\right) \tag{1.42}$$

$$= H^{\otimes n}U_r H^{\otimes n}\left(\frac{1}{2^{n/2}}\sum_{x\in\mathcal{X}}(-1)^{f(x)}|x\rangle\right) \tag{1.43}$$

$$= H^{\otimes n}U_r\left(\frac{1}{2^{n}}\sum_{x\in\mathcal{X}}\sum_{z\in\mathcal{X}}(-1)^{f(x)}(-1)^{x\cdot z}|z\rangle\right) \tag{1.44}$$

$$= H^{\otimes n}\left(\frac{1}{2^{n}}\sum_{x\in\mathcal{X}}(-1)^{f(x)}\left(|0^n\rangle - \sum_{z\in\mathcal{X},z\neq 0^n}(-1)^{x\cdot z}|z\rangle\right)\right) \tag{1.45}$$

$$= H^{\otimes n}\left(\frac{1}{2^{n}}\sum_{x\in\mathcal{X}}(-1)^{f(x)}\left(2|0^n\rangle - \sum_{z\in\mathcal{X}}(-1)^{x\cdot z}|z\rangle\right)\right) \tag{1.46}$$

$$= H^{\otimes n}\left(\frac{1}{2^{n}}\left(2(2^n-2)|0^n\rangle - \sum_{x\in\mathcal{X}}\sum_{z\in\mathcal{X}}(-1)^{f(x)}(-1)^{x\cdot z}|z\rangle\right)\right) \tag{1.47}$$

$$= \frac{2(2^n-2)}{2^n}\frac{1}{2^{n/2}}\sum_{x\in\mathcal{X}}|x\rangle - \frac{1}{2^{n/2}}\sum_{x\in\mathcal{X}}(-1)^{f(x)}|x\rangle \tag{1.48}$$

$$= (1-\frac{4}{2^n})\frac{1}{2^{n/2}}\sum_{x\in\mathcal{X},x\neq m}|x\rangle + (3-\frac{4}{2^n})\frac{1}{2^{n/2}}|m\rangle, \tag{1.49}$$

where $|m\rangle$ is the marked state. Note that from line 1.47 to 1.48, the fact that $H^{\otimes n}$ is its own inverse was used to simplify calculations. Therefore, we note that the final state is a superposition of the marked state $|m\rangle$ and another state which is an even superposition over all other computational basis states, which we shall denote $|m_\perp\rangle$. It turns out that

the Grover iteration preserves this basis as per the following:

$$G\left(\frac{1}{2^{n/2}}\sum_{x\in\mathcal{X}}|x\rangle\right) = G\left(\frac{1}{2^{n/2}}\sum_{x\in\mathcal{X},x\neq m}|x\rangle + \frac{1}{2^{n/2}}|m\rangle\right) \tag{1.50}$$

$$= G\left(\sqrt{\frac{2^n-1}{2^n}}|m_\perp\rangle + \sqrt{\frac{1}{2^n}}|m\rangle\right) \tag{1.51}$$

$$= G\left(\cos\theta|m_\perp\rangle + \sin\theta|m\rangle\right) \tag{1.52}$$

$$= (1-\frac{4}{2^n})\frac{1}{2^{n/2}}\sum_{x\in\mathcal{X},x\neq m}|x\rangle + (3-\frac{4}{2^n})\frac{1}{2^{n/2}}|m\rangle \tag{1.53}$$

$$= (1-\frac{4}{2^n})\sqrt{\frac{2^n-1}{2^n}}|m_\perp\rangle + (3-\frac{4}{2^n})\sqrt{\frac{1}{2^n}}|m\rangle \tag{1.54}$$

$$= (1-4\sin^2\theta)\cos\theta|m_\perp\rangle + (3-4\sin^2\theta)\sin\theta|m\rangle \tag{1.55}$$

$$= \cos 3\theta|m_\perp\rangle + \sin 3\theta|m\rangle, \tag{1.56}$$

where we have used the trigonometric identities: $3\sin\theta - 4\sin^2\theta = \sin 3\theta$ and $(1 - 4\sin^2\theta)\cos\theta = (-3\cos\theta + 4\cos^3\theta) = \cos 3\theta$. Therefore, the action of the Grover iterate G is to rotate the state in the $\{|m\rangle, |m_\perp\rangle\}$ plane by the angle 2θ. Repeating this procedure k times will yield the following state:

$$G^k\left(\frac{1}{2^{n/2}}\sum_{x\in\mathcal{X}}|x\rangle\right) = G^k\left(\cos\theta|m_\perp\rangle + \sin\theta|m\rangle\right) \tag{1.57}$$

$$= \cos(2k+1)\theta|m_\perp\rangle + \sin(2k+1)\theta|m\rangle. \tag{1.58}$$

Therefore, in order to maximize the probability of measuring the marked state $|m\rangle$, the Grover iterations are repeated to maximize $\sin(2k+1)\theta$, which will occur after $\mathcal{O}(\sqrt{2^n})$ iterations. Notice the key role that superposition plays in this algorithm; having the ability to call the oracle over such superpositions is the key difference to outperform classical algorithms.

1.3 Quantum error correction

1.3.1 Overcoming quantum noise

The goal of quantum error correction is to overcome unwanted quantum processes on encoded information from external sources of noise. In general, quantum systems that are

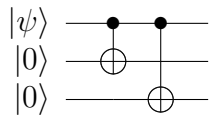


Figure 1.4: Encoding circuit for Shor's 3-qubit code. An arbitrary quantum state $|\psi\rangle$ is coupled to two ancillary qubits through $CNOT$ gates, yielding a state robust against X errors.

used in quantum computation are very sensitive to external parameters. This sensitivity is crucial to controlling such experimental systems, yet has the adverse consequence of being prone to unwanted external sources of noise.

Unlike error correction for classical systems, one of the primary obstacles for quantum error correction is that an arbitrary quantum state cannot be copied, as per the No-cloning Theorem 3. As such, quantum error correction techniques must find alternative methods of redundancy in encoding the quantum information to be protected. Again, entanglement will play an essential role in such encodings.

1.3.1.1 Shor's three-qubit code and stabilizers

One of the simplest forms of classical error correcting codes is the repetition code. Suppose one wanted to protect against arbitrary single bit flips X_i (we use the Pauli X_i operator here which is equivalent to the $\neg_i$ for binary classical codes). If we know that such flips happen with low probability p, then the probability of two such flips will be even more unlikely, being of order p^2. We encode the classical bit x_i by copying it over to other bits before exposing the full string to noise. Then, order to establish the state of the string after being subject to noise, we measure each of the bits and take the majority. Consider the following encoding [125]:

$$0 \to 000 = 0_{\mathcal{E}}, 1 \to 111 = 1_{\mathcal{E}}, \tag{1.59}$$

therefore the probability that an encoded state $x_{\mathcal{E}}$ has a majority of bits that have flipped to $\neg x$ is: $3p^2(1-p) + p^3$ which is strictly less than p for $p < 1/2$.

A similar form of encoding can be used to protect against this classical form of noise for qubits[3]. While arbitrary state cloning is not possible, it is possible to copy the basis

[3]We denote this noise as classical since the Kraus operators for the noise map have representation in only one type of Pauli basis (X in this example).

state representation of a state using a simple unitary transformation composed of $CNOT$ gates, Fig. 1.4 shows an explicit encoding circuit for the following:

$$\begin{aligned} |0\rangle \otimes |00\rangle \rightarrow U_{\mathcal{E}}|000\rangle = |000\rangle, \\ |1\rangle \otimes |00\rangle \rightarrow U_{\mathcal{E}}|100\rangle = |111\rangle. \end{aligned} \tag{1.60}$$

Using the above encoding, an arbitrary qubit $\alpha|0\rangle + \beta|1\rangle$ is mapped to the encoded state $\alpha|000\rangle + \beta|111\rangle$ that will protect against X errors (as in the case of the classical repetition code). The initial qubits in Eq. 1.67 that are always in the state $|0\rangle$ are called the ancillary qubits and are required for any quantum error correction encoding.

It was stated above that the 3-qubit encoding can protect against X errors just as in the classical case, however we know that directly measuring the state of the qubit and taking a majority vote approach will not necessarily work as it will collapse the quantum state onto one of the two codestates $|000\rangle$ or $|111\rangle$ (after correction). However, we would like to *protect* the encoded quantum state, not necessarily perform an operation that would amount to measuring it. Therefore, we require a quantum operation that extracts which state the majority of the qubits are in without directly measuring each of the individual qubits. This can be completed by measuring a sequence of parity measurements of pairs of qubits. Firstly measuring the parity of the first two qubits, followed by qubits 2 & 3. Such measurements correspond to measuring the observables Z_1Z_2 and Z_2Z_3, where Z_i is the Pauli operator on qubit i. The measurements will project onto the "+1" ("-1") eigenspaces thus recording an even (odd) parity of the corresponding qubits. Consider the four possible outcomes for X errors of weight 1 or less:

Error	Final state	Measurement outcomes	
$I(\alpha\|000\rangle + \beta\|111\rangle) \rightarrow$	$(\alpha\|000\rangle + \beta\|111\rangle)$	$+1,\ +1,$	(1.61)
$X_1(\alpha\|000\rangle + \beta\|111\rangle) \rightarrow$	$(\alpha\|100\rangle + \beta\|011\rangle)$	$-1,\ +1,$	(1.62)
$X_2(\alpha\|000\rangle + \beta\|111\rangle) \rightarrow$	$(\alpha\|010\rangle + \beta\|101\rangle)$	$-1,\ -1,$	(1.63)
$X_3(\alpha\|000\rangle + \beta\|111\rangle) \rightarrow$	$(\alpha\|001\rangle + \beta\|110\rangle)$	$+1,\ -1.$	(1.64)

Therefore, by associating each of the single-qubit error possibilities with one of the outcomes from the two parity measurements, we can correct for any single qubit X error by physically reapplying the appropriate X correction according to the measurement outcome. This is our first encounter with the notion of *stabilizers*.

We say that the operators Z_1Z_2 and Z_2Z_3 are stabilizers of the 3-qubit code as they have a trivial (equivalent to the identity operator) action on the codespace. Therefore any Pauli error that anti-commutes with these stabilizers must be an error that maps states

inside the codespace to states orthogonal to the codespace. We will return to this in much greater detail in Section 1.3.3.

Finally, it should be noted, given that the stabilizers of the 3-qubit code are composed of only Z type operators, any Z error will go undetected as such errors will commute with the stabilizers of the code. Of course, the code was designed to protect against X errors (to mimic the classical error correction repetition code), however any weight-1 Z error will result in a logical phase error, namely: $Z_1(\alpha|000\rangle + \beta|111\rangle) = \alpha|000\rangle - \beta|111\rangle$, and similarly for Z_2 and Z_3.

1.3.1.2 Shor's 9-qubit code and code concatenation

The idea behind Shor's 9-qubit code is to re-encode the physical qubits in the 3-qubit code into their own respective version of the 3-qubit code [125]. However, the new encoding is in a rotated basis in order to protect against phase errors. First consider the following rotated version of the 3-qubit code:

$$\begin{aligned} |+\rangle \otimes |00\rangle \rightarrow U_{\mathcal{E}}|+00\rangle = |+++\rangle, \\ |-\rangle \otimes |00\rangle \rightarrow U_{\mathcal{E}}|-00\rangle = |---\rangle. \end{aligned} \tag{1.65}$$

This encoding is equivalent to the previous version of the 3-qubit code, with the final states rotated into the X eigenbasis (this could simply be achieved by applying Hadamard gates to all of the qubits of the code, $H^{\otimes 3}$). Therefore, the action of the Pauli errors are reversed, X errors result in a phase error while single-qubit Z errors can be corrected by measuring the parity of the qubits in the new basis, that is by measuring the observables X_1X_2 and X_2X_3. These also correspond to the new stabilizers of the code.

Shor's idea was to *concatenate* the first encoding of the 3-qubit code with the second rotated version of the encoding of the 3-qubit code, that is to re-encode the qubits from the first encoding. This can be summarized below:

$$\begin{aligned} |0\rangle &\rightarrow U_{\mathcal{E}_1}|000\rangle = |000\rangle \\ &\rightarrow U_{\mathcal{E}_2}\left((|+\rangle + |-\rangle)|00\rangle \otimes (|+\rangle + |-\rangle)|00\rangle \otimes (|+\rangle + |-\rangle)|00\rangle\right) \\ &= (|+++\rangle + |---\rangle) \otimes (|+++\rangle + |---\rangle) \otimes (|+++\rangle + |---\rangle), \\ |1\rangle &\rightarrow U_{\mathcal{E}_1}|100\rangle = |111\rangle \\ &\rightarrow U_{\mathcal{E}_2}\left((|+\rangle - |-\rangle)|00\rangle \otimes (|+\rangle - |-\rangle)|00\rangle \otimes (|+\rangle - |-\rangle)|00\rangle\right) \\ &= (|+++\rangle - |---\rangle) \otimes (|+++\rangle - |---\rangle) \otimes (|+++\rangle - |---\rangle), \end{aligned} \tag{1.66}$$

(1.67)

where normalization factors have been omitted for ease of notation. By concatenating the two codes, this 9-qubit code gains important features from both codes, that is, the ability to protect against both single-qubit X and Z errors[4]. The code can protect against any single-qubit phase error by just considering the effect of the phase error on a single 3-qubit block. Consider the first 3 qubits, the action of a phase error will flip a qubit from $|+\rangle$ to $|-\rangle$, and vice-versa. As such, by measuring the observables associated with the rotated version of the 3-qubit code, we can detect and correct such an error. To correct for the bit flip errors is slightly tricker, as the action of a single X error on a 3-qubit block will go undetected within that block, as with the rotated version of the 3-qubit code which cannot correct against X errors. However, we can then use the property of the first encoding $\mathcal{E}_1$ to identify which blocks of three have undergone a logical X error. Suppose an X error occurred on one of the first 3 qubits, this would be equivalent to the error $X_1X_2X_3$ occurring, which can be thought of as a logical X error having occurred on the first 3 encoded qubits. This error can be differentiated from the same error having occurred on the second or third blocks of 3 qubits by measuring stabilizers that are equivalent to the traditional 3-qubit code, except now in block form. That is, $Z_1Z_2 \to Z_1Z_2Z_3Z_4Z_5Z_6$, and similarly $Z_2Z_3 \to Z_4Z_5Z_6Z_7Z_8Z_9$. Therefore, the overall stabilizers of the 9-qubit code are:

$$X_1X_2,\ X_2X_3, \qquad X_4X_5,\ X_5X_6, \qquad X_7X_8,\ X_8X_9, \tag{1.68}$$

$$Z_1Z_2Z_3Z_4Z_5Z_6, \qquad Z_4Z_5Z_6Z_7Z_8Z_9. \tag{1.69}$$

In order to decode after measuring the different syndrome observables, we proceed similarly to the 3-qubit code case. In the case of detected Z errors, we correct on each 3-qubit block identically to the case of the rotated 3-qubit code. In the case of X errors, we will correct by applying an X operator to a single qubit in the block that has been detected. Recall that an individual error on a single qubit block is equivalent to a logical fault on that block, that is $X_1 \cong X_2 \cong X_3 \cong X_1X_2X_3$. As such, if we apply Pauli X_1 when in fact the error X_2 had occurred, we will be left with the "error" X_1X_2 at the completion of decoding. However, notice that this is a stabilizer of the code, and as such is equivalent to the logical identity and the error has been corrected. This statement is equivalent to saying that the errors X_1, X_2, and X_3 have the same syndrome measurement outcomes, and therefore we can correct for them by applying any of the individual X operators. The fact that multiple errors produce the same syndrome measurements is an example of a *degenerate code*, and not all codes will have this property.

[4]The code presented here has a much higher protection against phase errors Z, being able to protect up to four Z errors.

1.3.1.3 Correcting arbitrary single-qubit noise

Thus far in our discussion of quantum error correction we have addressed the need to correct for single-qubit Pauli errors, however, in reality, noise will be much more complicated.

Since an arbitrary quantum channel is a completely positive map, we can express the action of the channel in its Pauli basis (which form an orthonormal spanning basis for such operators). As such, any single-qubit quantum channel can be expressed as follows:

$$\mathcal{N}(\rho) = \sum_k A_k \rho A_k^\dagger \tag{1.70}$$

$$= \sum_{ijk} (a_{k,i} P_i) \rho (a_{k,j}^* P_j) \tag{1.71}$$

Recall that for an error correcting code that can correct for arbitrary single qubit errors, we know there exists a recovery operator $\mathcal{R}$ such that for all $|\psi\rangle$ in the code and any single qubit Pauli error P_i the following holds: $U_M \Pi_M P_i |\psi\rangle = |\psi\rangle$, where Π_M is the projector obtained from measuring syndrome M, and U_M is the recovery operator that depends on M. Thus given any single-qubit error, there exists a set of recovery unitary operators $\{U_M\}$, depending on the measurement result obtained, that correct for the single-qubit error and map the encoded quantum state back to its noise-free encoding. Consider the action of the recovery operator $\mathcal{R}$ on a state that has undergone arbitrary noise of the form of Eq. 1.71 for a logical state of the form $\rho = \sum_m p_m |\psi_m\rangle\langle\psi_m|$, where all $|\psi_k\rangle$ are members of the

quantum error correcting code:

$$\begin{aligned}
\mathcal{R}\circ\mathcal{N}(\rho) &= \mathcal{R}\circ\left(\sum_{ijk} a_{k,i}a_{k,j}^{*}P_i\rho P_j\right)\\
&= \left(\sum_{ijk} a_{k,i}a_{k,j}^{*}U_M\Pi_M P_i\rho P_j\Pi_M U_M^{\dagger}\right)\\
&= \left(\sum_{ijk} a_{k,i}a_{k,j}^{*}U_M\Pi_M P_i(\sum_{m} p_m|\psi_m\rangle\langle\psi_m|)P_j\Pi_M U_M^{\dagger}\right)\\
&= \left(\sum_{ijkm} a_{k,i}a_{k,j}^{*}p_m U_M\Pi_M P_i|\psi_m\rangle\langle\psi_m|P_j\Pi_M U_M^{\dagger}\right)\\
&= \left(\sum_{ijk} a_{k,i}a_{k,j}^{*}\right)\sum_{m} p_m|\psi_m\rangle\langle\psi_m|\\
&= \sum_{m} p_m|\psi_m\rangle\langle\psi_m| = \rho. \qquad (1.72)
\end{aligned}$$

where the sum over the coefficients $\sum_{ijk} a_{k,i}a_{k,j}^{*} = 1$ for trace preserving channels. Therefore, given an error correcting code along with a decoder that can correct any single-qubit Pauli operator, the code will naturally be protected against any form of single-qubit noise as well!

The 9-qubit code encompasses many important concepts in quantum error correction that will be revisited in great detail in the next few sections. To summarize:

- *Encoding and decoding:* Quantum information is encoded, via an encoding circuit, in order to protect the information in a higher-dimensional (typically highly entangled) subspace. Measurements that do not collapse the state of the encoded information are performed to inform the decoder of the location of the errors. Unitary decoding operations are applied given the information from the measurement.

- *Stabilizers:* Stabilizers are special Pauli operators that preserve the codespace (logically equivalent to the identity operation). The stabilizers form the observables to be measured in order to locate the presence of an error.

- *Code concatenation:* Two error correcting codes can be concatenated to form a larger error correcting code that will have many of the features of both codes. This can be used to increase the protection of the logical qubits.

- *Degenerate codes:* Quantum error correcting codes in which not all errors have a unique syndrome. However, errors can still be corrected because errors sharing the same syndrome form a stabilizer when combined together.
- *Arbitrary single-qubit noise:* Any quantum error correcting code that can protect against any single-qubit Pauli error will also be able to protect against arbitrary single-qubit noise. A similar form of result will hold for codes that can protect against Pauli noise on more than 1 qubit.

1.3.2 Conditions for quantum error correction

In the previous section, two notions of error correcting codes were presented, one which protected against one form of Pauli noise (analogous to classical noise) and one which corrected arbitrary noise on a single qubit. One may ask: "What then are the central conditions for quantum error correction?" We first require a formal definition of an error correcting code.

Definition 12. *A n-qubit **quantum error correcting code** composed of k logical qubits is defined as a pair $(\mathcal{C}, \mathcal{R})$, where $\mathcal{C}$ a 2^k dimensional Hilbert space denoted the codespace, and $\mathcal{R}$ is a recovery channel.*

Definition 13. *Given a set of errors $\mathcal{E}$, the quantum error correcting code $(\mathcal{C}, \mathcal{R})$ is said to correct for the errors $\mathcal{E}$ if the following holds:*

$$\mathcal{R} \circ (E_i|\psi\rangle\langle\psi|E_i^\dagger) = |\psi\rangle\langle\psi|, \ \forall \ |\psi\rangle \in \mathcal{C}, \ E_i \in \mathcal{E}. \tag{1.73}$$

Knill and Laflamme discovered that since errors act by mapping encoded qubits outside the protected Hilbert space $\mathcal{C}$, in order for a recovery operation to be possible orthogonal logical states get mapped to orthogonal subspaces under the action of the error channel. Since no quantum channel can distinguish between non-orthogonal states, if orthogonal logical states are mapped to states that are non-orthogonal, we have no hope of recovering the information in a perfect manner. Moreover, Knill and Laflamme were able to show that not only is this necessary but it is indeed sufficient for the existence of a recovery operator.

Theorem 14 (Knill–Laflamme conditions [88])**.** *Given a set of errors $\mathcal{E}$ and a Hilbert space $\mathcal{C}$, then there exists a recovery operator $\mathcal{R}$ such that the quantum error correcting code $(\mathcal{C}, \mathcal{R})$ corrects for the errors $\mathcal{E}$ if and only if:*

$$\langle\psi_\alpha|E_i^\dagger E_j|\psi_\beta\rangle = c_{ij}\delta_{\alpha\beta}, \ \forall \ |\psi_\alpha\rangle, |\psi_\beta\rangle \in \mathcal{C}, E_i, E_j \in \mathcal{E}, \tag{1.74}$$

where $c_{ij} \in \mathbb{C}$ and $\delta_{\alpha\beta}$ is the Kronecker delta.

Corollary 15 ([88]). *Given a set of errors $\mathcal{E}$ and a Hilbert space $\mathcal{C}$, then there exists a recovery operator $\mathcal{R}$ such that the quantum error correcting code $(\mathcal{C}, \mathcal{R})$ corrects for the errors $\mathcal{E}$ if and only if:*

$$\Pi_{\mathcal{C}} E_i^{\dagger} E_j \Pi_{\mathcal{C}} = c_{ij} \Pi_{\mathcal{C}}, \ \forall \ E_i, E_j \in \mathcal{E}, \tag{1.75}$$

where $\Pi_{\mathcal{C}}$ is the projector onto the Hilbert space $\mathcal{C}$.

1.3.3 The stabilizer formalism

In this section we present what is now known as the stabilizer formalism, developed by Gottesman, which has become the primary language for discussing quantum error correcting codes. Many important concepts are summarized here, for a more detailed description of the founding ideas see Ref. [58].

Definition 16. *Given a subspace $\mathcal{Q} \subseteq \mathcal{H}$, a **stabilizer** S is a Pauli operator such that:*

$$S|\psi\rangle = |\psi\rangle, \ \forall \ |\psi\rangle \in \mathcal{Q}. \tag{1.76}$$

Lemma 17. *Given a subspace $\mathcal{Q} \subseteq \mathcal{H}$, the set of stabilizers form a multiplicative Abelian group $\mathcal{S}$.*

Proof. The identity element of the group is clear as $I_{\mathcal{H}} \in \mathcal{S}$, where $I_{\mathcal{H}}$ is the identity element on the Hilbert space $\mathcal{H}$. Since any Pauli operator has the property $P^2 = I_{\mathcal{H}}$, any stabilizer S is its own inverse. Let $|\psi\rangle \in \mathcal{Q}$, given two stabilizers S_1 and S_2, $S_1 S_2 |\psi\rangle = S_1 |\psi\rangle = |\psi\rangle$, and as such $(S_1 S_2)$ is a stabilizer. Finally, given now a third stabilizer S_3, $S_1(S_2 S_3)|\psi\rangle = |\psi\rangle = (S_1 S_2) S_3 |\psi\rangle$, showing associativity. Therefore the set of stabilizers are a group, what remains to be shown is that the group is Abelian. Since by definition stabilizers are Pauli operators, we can use the following fact about Pauli operators: $S_1 S_2 = \pm S_2 S_1$. However, we cannot have that $S_1 S_2 = -S_2 S_1$ else we would conclude that $|\psi\rangle = S_1 S_2 |\psi\rangle = -S_2 S_1 |\psi\rangle = -|\psi\rangle$ which is a contradiction. □

We have demonstrated that the stabilizers of a given subspace $\mathcal{Q}$ form an Abelian subgroup $\mathcal{S} \subset \mathcal{P}_n$ of the n-qubit Pauli group (assuming $\dim(\mathcal{H}) = 2^n$), however the converse is also true, any Abelian subgroup $\mathcal{S} \subset \mathcal{P}_n$ such that $-I, iI \notin \mathcal{S}$ will form a subspace Q such that $\dim(Q) = \dim(\mathcal{H}) - \dim(\mathcal{S})$.

Definition 18. *A n-qubit **stabilizer group** is an Abelian subgroup $\mathcal{S} \in \mathcal{P}_n$ such that $-I, iI \notin \mathcal{S}$.*

Definition 19. *A stabilizer group $\mathcal{S} \in \mathcal{P}_n$ and recovery operator $\mathcal{R}_\mathcal{S}$ define a* ***stabilizer code*** *$(\mathcal{Q}_\mathcal{S}, \mathcal{R}_\mathcal{S})$, where $\mathcal{Q}_\mathcal{S} = \{|\psi\rangle \in \mathcal{H} \mid S|\psi\rangle = |\psi\rangle \ \forall \ S \in \mathcal{S}\}$. Moreover, given $dim(\mathcal{S}) = 2^{n-k}$, $dim(\mathcal{Q}_\mathcal{S}) = 2^k$, that is the error correcting code is composed of k logical qubits.*

The stabilizer group by definition leaves the codespace invariant and is logically equivalent to the identity operator. Given a codespace $\mathcal{Q}_\mathcal{S}$ encoding k logical qubits, there must be k logical Pauli X and Z operators. By definition, a logical Pauli operator P_L must map any element of $\mathcal{Q}_\mathcal{S}$ to another element of $\mathcal{Q}_\mathcal{S}$, that is for all $|\psi\rangle \in \mathcal{Q}_\mathcal{S}$, $P_L|\psi\rangle = |\psi'\rangle \in \mathcal{Q}_\mathcal{S}$, where $|\psi'\rangle$ does not have to necessarily be different from $|\psi\rangle$. As such, we must have the following for all $|\psi\rangle \in \mathcal{Q}_\mathcal{S}$ and $S \in \mathcal{S}$, $P_L S|\psi\rangle = P_L|\psi\rangle = |\psi'\rangle = S|\psi'\rangle = SP_L|\psi\rangle \Rightarrow P_L S = SP_L$. Therefore, the set of logical Pauli operators are the Pauli operators that commute with the stabilizer group $\mathcal{S}$, that is the centralizer of $\mathcal{S}$. However, in the case of Pauli operators, the centralizer of $\mathcal{S}$ is equivalent to the normalizer of $\mathcal{S}$ and typically we define the logical Pauli operators as the elements from this group.

Definition 20. *Given a stabilizer code composed from the stabilizer group $\mathcal{S}$, the* ***non-trivial*** *(non-identity)* ***logical Pauli operators*** *of the code are given by the elements of the normalizer of $\mathcal{S}$ that are not in $\mathcal{S}$, that is $\mathcal{N}(\mathcal{S})\backslash\mathcal{S}$, where*

$$\mathcal{N}(\mathcal{S}) = \{P \in \mathcal{P}_n \mid PSP^\dagger = S' \in \mathcal{S}\}. \tag{1.77}$$

The normalizer gives a very nice characterization of the form of logical operators that a stabilizer code can have. Moreover, it is very indicative of how well a code protects against noise and defines a very important parameter for stabilizer codes.

Definition 21. *The* ***distance*** *of a stabilizer group $\mathcal{S}$ is defined as:*

$$d = \min\{\text{wt}(P) | P \in \mathcal{P}_n, P \in \mathcal{N}(\mathcal{S})\backslash\mathcal{S}\}, \tag{1.78}$$

where $\text{wt}(P)$ *is the weight of the Pauli operator $P = \otimes_{i=1}^n P_i$, that is the total number of qubits where $P_i \neq I$.*

Theorem 22. *Given a n qubit stabilizer group $\mathcal{S}$, encoding k logical qubits with distance d, there always exists a recovery operator $\mathcal{R}_\mathcal{S}$ such that the associated stabilizer code can correct any error of weight less than or equal to $\lfloor (d-1)/2 \rfloor$. The associated stabilizer code is denoted a $[[n, k, d]]$ stabilizer code.*

Proof. Let $\mathcal{E} = \{E_i\}$ be a set of errors of weight less than or equal to $\lfloor (d-1)/2 \rfloor$. Then we must have that

$$\langle \psi_k | E_i^\dagger E_j | \psi_l \rangle = c_{ij} \delta_{kl}, \ \forall \ |\psi_k\rangle, |\psi_l\rangle \in \mathcal{Q}_\mathcal{S}, E_i, E_j \in \mathcal{E}, \tag{1.79}$$

since else $E_i^\dagger E_j$ would have overlap with a logical operator, yet we know that $E_i^\dagger E_j$ has weight at most $(d-1)$, thus this would contradict the definition of the normalizer. Therefore, for any set of errors $\mathcal{E}$ satisfying the statement of the Theorem, the codespace satisfies the Knill-Laflamme Theorem 14 and the decoder $\mathcal{R}_\mathcal{S}$ exists. □

Theorem 23. *Given a $[[n,k,d]]$ stabilizer code and let $\mathcal{H}_{d-1}$ be a subset of at most $(d-1)$ qubits. Then there always exists a recovery operator $\mathcal{R}_\mathcal{S}$ that corrects for any set of errors $\mathcal{E} = \{E_i\} \in \mathcal{H}_{d-1}$.*

Proof. Since for any E_i and E_j, the support of both is contained in $\mathcal{H}_{d-1}$, as such $\text{wt}(E_i^\dagger E_j) \leq d-1$ and the Knill-Laflamme conditions must trivially hold. □

Therefore, the distance of a quantum error correcting code provides two important notions for correctable errors. Firstly, if the error set is of weight less than half the size of the distance then such errors must be correctable. Secondly, if the location of the errors are known, the quantum error correcting code will be able to correct for up to $(d-1)$ errors.

1.3.4 Fault-tolerant quantum computation

Thus far we have discussed the properties of quantum error correcting codes and how in theory recovery operations can be constructed to map back to the initial logical state. However, in practice, the physical qubits storing the information will not only undergo some form of quantum noise while being stored, but rather both the encoding and recovery operations will be noisy as well. Moreover, logical operations will also result in errors, and it will be important that the set of logical operations do not spread errors in an uncontrollable manner. We will return to the different possible construction of logical gates in Chapter 2 and will define the notion of fault-tolerance here.

Traditionally the idea behind a fault-tolerant operation is that a correctable error remains correctable at the conclusion of the logical operation. We will present a slightly looser notion of fault-tolerance that guarantees that some subset of correctable errors remain correctable at the conclusion of the operation. In introducing this notion, some of the error correction properties of the code will be sacrificed, however, overall quantum state protection will be guaranteed and a fault-tolerance threshold will still exist.

Definition 24. *Given a quantum error correcting code $\mathcal{Q}$, a quantum operation $\mathcal{A}$ is said to be **fault-tolerant** against a set of physical errors $\mathcal{E}$ if the resulting error at the conclusion*

of the operation remains correctable. That is, there exists a recovery operation $\mathcal{R}_E$ such that:

$$\mathcal{R}_E \circ \mathcal{A}_E(\rho) = \mathcal{A}(\rho),\ \forall\ \rho \in \mathcal{Q},\ E \in \mathcal{E}, \tag{1.80}$$

where $\mathcal{A}_E$ is the noisy quantum channel associated with the composition of the ideal operation $\mathcal{A}$ and the error E that may occur at a given point throughout the action of the ideal gate.

The simplest form of fault-tolerant operation is one that does not couple qubits within the error correcting code, thereby not increasing the weight of the errors.

Definition 25. *Given a quantum error correcting code $\mathcal{Q}$, a logical gate G is said to be* ***transversal*** *if there is a representation of G as a product of individual gates. That is, there exists an operation of the form $\otimes_{i=1}^{n} G_i$, such that*

$$\begin{aligned} G\rho G^\dagger =& (\otimes_{i=1}^{n} G_i)\rho(\otimes_{i=1}^{n} G_i^\dagger) \\ &= \otimes_{i=1}^{n} G_i \rho G_i^\dagger,\ \forall\ \rho \in \mathcal{Q}. \end{aligned} \tag{1.81}$$

Importantly, the definition of transversal gates encompasses 2-qubit as well as single-qubit logical gates, namely there will be some codes that will contain a transversal $CNOT$ gate.

Lemma 26. *Given a quantum error correcting code $\mathcal{Q}$, any transversal gate G will be fault-tolerant for the set of correctable errors $\mathcal{E} = \{E_i\}$ of weight $wt(E_i) \leq \lfloor (d-1)/2 \rfloor$.*

Proof. Given any error E_i, that can occur either before or after the action of the gate G, the resulting set of errors that can occur at the conclusion of the gate is given by $\mathcal{E} \cup \mathcal{F}$, where $\mathcal{F} = \{GE_iG^\dagger\}$. However, note that for any codestate $\rho \in \mathcal{Q}$ the action of the modified error will be:

$$\begin{aligned} G\rho G^\dagger &= \otimes_{j=1}^{n} G_j E_i \rho E_i^\dagger G_j^\dagger \\ &= \otimes_{j=1}^{n} G_j E_i G_j^\dagger G_j \rho G_j^\dagger G_j E_i^\dagger G_j^\dagger \\ &= \otimes_{j=1}^{n} G_j E_i G_j^\dagger (G\rho G^\dagger) G_j E_i^\dagger G_j^\dagger, \end{aligned} \tag{1.82}$$

and as such the error $GE_iG^\dagger$ will only have non-trivial action at the locations where E_i will have support, therefore $GE_iG^\dagger$ is logically equivalent to an error of weight at most $\lfloor (d-1)/2 \rfloor$ and as such is correctable. Therefore, the union of correctable errors remain correctable, and G is fault-tolerant for the set of errors of weight $\leq \lfloor (d-1)/2 \rfloor$. □

Fault-tolerant operations are essential as they will allow for the construction of schemes to reduce the logical error rate to arbitrarily small levels, allowing for the possibility of large scale quantum computations. The basic idea of the fault-tolerance threshold theorem is that given a family of codes with increasing distance, there will always be a threshold error rate below which the logical error rate can be made arbitrarily small with only poly-logarithmic overhead in the number of operations. As with many statements about error correction in large systems, there are underlying assumptions that go into the fault-tolerance threshold theorem, most importantly that the individual errors remain bounded in the number of qubits that they influence. Here, the threshold theorem is stated for concatenated codes, yet strong numerical evidence supports the existence of a threshold for topological codes as well. We shall return to this in Chapter 2.

Theorem 27 ([1, 2, 3])**.** *Suppose a quantum error correcting code $\mathcal{Q}$ and a noise map $\mathcal{N} = E_N \circ \cdots \circ E_1$ consisting of the composition of a set of correctable errors $\mathcal{E} = \{E_i\}_{i=1}^N$ each occurring with probability at most p. Then, there exists a* ***fault-tolerance threshold error rate*** *p_{th} such that for any fault-tolerant quantum circuit $\mathcal{A}$ and target logical error rate ϵ, there exists a quantum circuit with overhead $\mathcal{O}(\text{poly}(\log 1/\epsilon))$ for all $p < p_{th}$.*

Proof. Given a quantum error correcting code $\mathcal{Q}$ and a set of correctable errors $\mathcal{E}$ occurring with probability p, then given the fault-tolerant operation $\mathcal{A}$, the probability of the resulting error being uncorrectable at the conclusion of the operation $\mathcal{A}$ is bounded by:

$$p^{(1)} \leq \sum_{k=2}^{N} \binom{N}{k} p^k (1-p)^{N-k} \tag{1.83}$$

$$\leq \sum_{k=2}^{N} \binom{N}{k} p^2 (1-p)^{N-2} \tag{1.84}$$

$$\leq 2^N p^2 (1-p)^{N-2} \tag{1.85}$$

$$\leq 2^N p^2, \tag{1.86}$$

where N is the cardinality of the set of correctable errors $\mathcal{E}$ that occur with probability p. Notice that the logical error rate will be smaller than the physical error rate below a threshold $p < p_{th} = 1/2^N$. Now, the logical information can be re-encoded by concatenating the code with itself, resulting in a new quantum code where each logical qubit is composed of multiple logically encoded qubits. After applying error correction to each of the individual blocks, errors are projected onto the logical subspace as either the identity map (in the case of correctable errors) or a logical fault (in the case of uncorrectable errors). Therefore,

the new error correcting code will have a logical error rate bounded by the same equation as given above with the physical error rate p replaced by the logical error rate $p^{(1)}$:

$$p^{(2)} \leq 2^N (p^{(1)})^2 \tag{1.87}$$
$$\leq 2^N (2^N p^2)^2 \tag{1.88}$$
$$= 2^{3N} p^4 \tag{1.89}$$
$$\leq \left(\frac{p}{p_{th}}\right)^3 p. \tag{1.90}$$

By continuing to concatenate using the same code, the logical error rate can be continually suppressed in a doubly exponential manner. Generally, the logical error rate at the m-th concatenation level will be upper bounded by:

$$p^{(m)} \leq 2^N \left(\left(\frac{p}{p_{th}}\right)^{2^{m-1}-1} p\right)^2 \tag{1.91}$$
$$= \left(\frac{p}{p_{th}}\right)^{2^m-1} p. \tag{1.92}$$

Therefore, the logical error rate decreases double-exponentially with respect to the fraction p/p_{th}, thus allowing for efficient overhead with respect to the exponential increase in qubits as m increases. The overall number of concatenation levels required to reach a target error rate ϵ will thus be:

$$\left(\frac{p}{p_{th}}\right)^{2^m-1} p \leq \epsilon \tag{1.93}$$
$$\Rightarrow 2^m - 1 \leq \log_{p/p_{th}} \epsilon \tag{1.94}$$
$$\Rightarrow m \leq C \log\log \frac{1}{\epsilon}. \tag{1.95}$$

Given that each concatenation level consists of encoding the qubits of the previous level in a quantum error correcting code, assuming the original code has n qubits, the total overhead in qubit number for a code with m levels of concatenation is:

$$n^m = n^{C \log\log \frac{1}{\epsilon}} = A \cdot \text{poly}(\log \frac{1}{\epsilon}) \tag{1.96}$$

□

Chapter 2

Transversality and fault-tolerant quantum error correction

2.1 Fault-tolerant universal gate sets

2.1.1 Transversal gate sets

In Section 1.3, the notion of a fault-tolerant operation was introduced. An important set of naturally fault-tolerant operations are traversal gates, that is the set of logical gates of the form $G = \otimes_{i=1}^{n} G_i$. Transversal gates are of interest for both theoretical fault-tolerance purposes but also for their potential ease of application in an experimental setting. The ability for transversal gates to implement a logical gate by addressing each qubit individually, in one time step, without having to use ancillary systems or complicated coupling, makes them the most desirable gate in terms of reducing the overhead associated with fault-tolerance.

Definition 28. *Given a $[[n, k, d]]$ qubit stabilizer code $\mathcal{S}$, we define the the parity check matrix $K_{\mathcal{S}}$ as a $(2n) \times (n-k)$ binary matrix where each row is associated with one of the $(n-k)$ stabilizer generators of the code. Given a stabilizer generator S, the associated row has an entry of 1 in the first n columns for every location where S has an X or Y, and an entry of 1 in columns $n+1$ to $2n$ for every location where S has a Y or Z. If the stabilizer generators can be partitioned into two sets of X and Z generators, then the parity check matrix can be separated into two submatricies $K_{\mathcal{S}_X}$ and $K_{\mathcal{S}_Z}$.*

Example 29. *The parity check matrix for the Shor 9-qubit code, whose stabilizers are given in Eqs. 1.68–1.69, is:*

$$K_{\mathcal{S}} = \left(\begin{array}{ccccccccc|ccccccccc} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right) \tag{2.1}$$

Notice that the parity check matrix of the code can be partitioned into two submatricies for both the X and Z type.

Definition 30. *Let $\mathcal{C}_1$ be a $[n, k_X, d_X]$ code that corrects for X errors, defined by the stabilizer group $\mathcal{S}_Z$ (single square brackets for the code parameters signifying this is equivalent to a classical code). Let $\mathcal{C}_2$ be a $[n, k_Z, d_Z]$ code that corrects for Z errors, defined by the stabilizer group $\mathcal{S}_X$. Suppose $\mathrm{Ker}(K_{\mathcal{S}_X}) \subseteq \mathrm{Ker}(K_{\mathcal{S}_Z})$ and $d = \min(d_1, d_2)$ (where $\mathrm{Ker}(A)$ is the Kernel of A), then $\mathcal{C}_1$ and $\mathcal{C}_2$ define a $[[n, k_X - k_Z, d]]$ **CSS stabilizer code**, where the stabilizer group $\mathcal{S}$ can be partitioned into two subgroups containing operators of X and Z type, that is $\mathcal{S} = \mathcal{S}_X \cup \mathcal{S}_Z$.*

The CSS code construction allows for the creation of a quantum error correcting code from two classical codes, each aimed to address the errors of X or Z type. The statement that $\mathrm{Ker}(K_{\mathcal{S}_X}) \subseteq \mathrm{Ker}(K_{\mathcal{S}_Z})$ is equivalent to stating that the classical code defined to correct against Z errors must be contained in the one defined to correct against X errors. The logical X and Z Pauli operators will also consist only of X and Z Pauli operators, respectively, as each of the classical codes will have a logical operator with even overlap with the stabilizers of their respective types. As such, codewords of the CSS codes can be constructed in a simple manner.

Lemma 31. *Given a $[[n, k, d]]$ CSS code with stabilizers $\mathcal{S} = \mathcal{S}_X \cup \mathcal{S}_Z$ and stabilizer*

generators $\mathcal{S}_X = \langle G_i^X \rangle$, $\mathcal{S}_Z = \langle G_i^Z \rangle$, then the codestates can be expressed as follows:

$$|0\cdots 0\rangle_L = \frac{1}{\sqrt{|\mathcal{S}_X|}} \prod_i (I + G_i^X)|0\rangle^{\otimes n} \tag{2.2}$$

$$= \frac{1}{\sqrt{|\mathcal{S}_X|}} \sum_{s_x \in \mathcal{S}_X} |s_x\rangle, \tag{2.3}$$

$$|j_1 \cdots j_k\rangle_L = \left(\prod_{m=1}^{k} X_{L,m}^{j_m} \right) \frac{1}{\sqrt{|\mathcal{S}_X|}} \prod_i (I + G_i^X)|0\rangle^{\otimes n}. \tag{2.4}$$

Proof. The codestates of a stabilizer code are defined to be the states that are eigenstates of the stabilizer group $\mathcal{S}$. Equivalently, $|\psi\rangle$ is a codestate if it is stabilized by all of the stabilizer generators of the code. Consider first the Z type stabilizer generators and their action on the state $|0\cdots 0\rangle_L$ defined above:

$$G_j^Z |0\cdots 0\rangle_L = \frac{1}{\sqrt{|\mathcal{S}_X|}} G_j^Z \prod_i (I + G_i^X)|0\rangle^{\otimes n} \tag{2.5}$$

$$= \frac{1}{\sqrt{|\mathcal{S}_X|}} \prod_i (I + G_i^X) G_j^Z |0\rangle^{\otimes n} \tag{2.6}$$

$$= \frac{1}{\sqrt{|\mathcal{S}_X|}} \prod_i (I + G_i^X)|0\rangle^{\otimes n}, \tag{2.7}$$

$$\tag{2.8}$$

since G_j^Z commutes with all the X type stabilizer generators by definition of being a stabilizer group. Moreover, the action of G_j^Z on the state $|0\rangle^{\otimes n}$ is trivial since G_j^Z is composed of only Z operators. Now, consider the action of the X operators:

$$G_j^X |0\cdots 0\rangle_L = \frac{1}{\sqrt{|\mathcal{S}_X|}} G_j^X \prod_i (I + G_i^X)|0\rangle^{\otimes n} \tag{2.9}$$

$$= \frac{1}{\sqrt{|\mathcal{S}_X|}} \left(\prod_{i \neq j} (I + G_i^X) \right) G_j^X (I + G_j^X)|0\rangle^{\otimes n} \tag{2.10}$$

$$= \frac{1}{\sqrt{|\mathcal{S}_X|}} \prod_i (I + G_i^X)|0\rangle^{\otimes n}, \tag{2.11}$$

$$\tag{2.12}$$

where the terms in the product can be reordered in any way due to commutativity, and we have used the simple identity that $G_j^X(I + G_j^X) = I + G_j^X$. Therefore, the state $|0\cdots 0\rangle_L$ must be an eigenstate of the stabilizer group, and moreover all other codestates can be generated from the logical X operators as by definition they preserve the codespace and will act non-trivially on the state $|0\cdots 0\rangle_L$ as the logical X operators cannot be generated from the stabilizer group $\mathcal{S}_X$. □

Lemma 32. *The CNOT gate can be implemented transversally for all CSS codes. Given two encoded codeblocks of the same code $\mathcal{Q}$, the transversal $CNOT$ has the following action on the computational logical basis states:*

$$CNOT^{\otimes n}|i_1\cdots i_k\rangle|j_1\cdots j_k\rangle = |i_1\cdots i_k\rangle|(i_1\oplus j_1)\cdots(i_k\oplus j_k)\rangle. \tag{2.13}$$

Proof. This result will be proved by considering the action of the transversal $CNOT$ gate on the stabilizers of the codestates. The stabilizers of the state $|i_1\cdots i_k\rangle|j_1\cdots j_k\rangle$ are:

$$\begin{aligned}
\mathcal{S}_X\otimes I^{\otimes n} \qquad & I^{\otimes n}\otimes\mathcal{S}_X && (2.14)\\
\mathcal{S}_Z\otimes I^{\otimes n} \qquad & I^{\otimes n}\otimes\mathcal{S}_Z && (2.15)\\
(-1)^{i_1}Z_{L,1}\otimes I^{\otimes n} \qquad & I^{\otimes n}\otimes(-1)^{j_1}Z_{L,1} && (2.16)\\
\vdots \qquad & \vdots &&\\
(-1)^{i_k}Z_{L,k}\otimes I^{\otimes n} \qquad & I^{\otimes n}\otimes(-1)^{j_k}Z_{L,k}. && (2.17)
\end{aligned}$$

The action of the $CNOT$ gate on the Pauli basis can be summarized as follows (from qubit 1 to 2):

$$\begin{aligned}
CNOT: XI &\longrightarrow XX && (2.18)\\
IX &\longrightarrow IX && (2.19)\\
ZI &\longrightarrow ZI && (2.20)\\
IZ &\longrightarrow ZZ. && (2.21)
\end{aligned}$$

Therefore the action of the transversal $CNOT$ gate on any X stabilizer S_X:

$$\begin{aligned}
S_X\otimes I^{\otimes n} &\longrightarrow S_X\otimes S_X && (2.22)\\
I^{\otimes n}\otimes S_X &\longrightarrow I^{\otimes n}\otimes S_X, && (2.23)
\end{aligned}$$

however by multiplying the two resulting outputs together, these stabilizers are equivalent to the original X stabilizers of the code. Similarly for any given Z stabilizer S_Z:

$$\begin{aligned}
S_Z\otimes I^{\otimes n} &\longrightarrow S_Z\otimes I^{\otimes n} && (2.24)\\
I^{\otimes n}\otimes S_Z &\longrightarrow S_Z\otimes S_Z, && (2.25)
\end{aligned}$$

and again by multiplying the two stabilizers the original Z stabilizers are recovered, therefore the final state must be a codestate. Finally, consider the action of the gate on the logical operators:

$$(-1)^{i_m} Z_{L,m} \otimes I^{\otimes n} \longrightarrow (-1)^{i_m} Z_{L,m} \otimes I^{\otimes n} \tag{2.26}$$
$$I^{\otimes n} \otimes (-1)^{j_m} Z_{L,m} \longrightarrow (-1)^{j_m} Z_{L,m} \otimes Z_{L,m}, \tag{2.27}$$

the mapped logical operators are equivalent to the following by multiplying the second line with the first:

$$(-1)^{i_m} Z_{L,m} \otimes I^{\otimes n} \tag{2.28}$$
$$(-1)^{i_m + j_m} I^{\otimes n} \otimes Z_{L,m}. \tag{2.29}$$

Therefore, the stabilizers of the output state are given by:

$$\begin{aligned}
\mathcal{S}_X \otimes I^{\otimes n} \qquad & I^{\otimes n} \otimes \mathcal{S}_X && (2.30)\\
\mathcal{S}_Z \otimes I^{\otimes n} \qquad & I^{\otimes n} \otimes \mathcal{S}_Z && (2.31)\\
(-1)^{i_1} Z_{L,1} \otimes I^{\otimes n} \qquad & I^{\otimes n} \otimes (-1)^{i_1 + j_1} Z_{L,1} && (2.32)\\
\vdots \qquad & \vdots \\
(-1)^{i_k} Z_{L,k} \otimes I^{\otimes n} \qquad & I^{\otimes n} \otimes (-1)^{i_k + j_k} Z_{L,k}, && (2.33)
\end{aligned}$$

which are the stabilizers of the computational basis state $|i_1 \cdots i_k\rangle|(i_1 \oplus j_1) \cdots (i_k \oplus j_k)\rangle$. □

Example 33. *The 7-qubit Steane code is a* $[[7, 1, 3]]$ *CSS code with the following stabilizer generators [128]:*

$$\begin{aligned}
XXXXIII \qquad & ZZZZIII \\
XXIIXXI \qquad & ZZIIZZI \\
XIXIXIX \qquad & ZIZIZIZ, && (2.34)
\end{aligned}$$

and logical operators $X_L = X^{\otimes 7} \cong IIIIXXX$*, and* $Z_L = Z^{\otimes 7} \cong IIIIZZZ$*.*

Lemma 34. *The Clifford gate set is transversal for the 7-qubit Steane code.*

Proof. In order to show that the full set of Clifford gates is transversal for the 7-qubit code, it suffices to show that a generating set of gates are transversal for the code. As shown above, the $CNOT$ gate is transversal for the 7-qubit code as it is a CSS code.

The Hadamard gate is transversal as the X and Z stabilizers of the code are symmetric and Hadamard has the action of mapping X to Z and vice versa, as well under the action of the Hadamard $H^{\otimes 7}X^{\otimes 7}H^{\otimes 7} = Z^{\otimes 7}$, therefore concluding the correct logical operation on the logical Pauli operators.

Finally, to complete the generating set for the Clifford group, consider the action of the transversal phase gate, that is the gate $S = |0\rangle\langle 0| + i|1\rangle\langle 1|$. Since the phase gate is diagonal it preserves the Z gate under conjugation and the action on the X gate results in: $SXS^\dagger = Y$. Therefore the X stabilizers will transform as follows:

$$\begin{aligned} XXXXIII &\longrightarrow YYYYIII \\ XXIIXXI &\longrightarrow YYIIYYI \\ XIXIXIX &\longrightarrow YIYIYIY. \end{aligned} \tag{2.35}$$

However, note that the transformed stabilizers are a product of the original X and Z stabilizers, and as such the stabilizer group remains the same. Finally, the logical X operator transforms as: $S^{\otimes 7}X^{\otimes 7}S^{\dagger\otimes 7} = Y^{\otimes 7} = (-iZX)^{\otimes 7} = iZ^{\otimes 7}X^{\otimes 7} = -Y^{\otimes 7}$. Therefore, $S^{\otimes 7}$ results in the logical application of $S^\dagger$, which is still a generating gate for the Clifford gate set. □

Note that $S^{\otimes 7}$ being logically equivalent to $S^\dagger$ for the 7-qubit code is the first example we have seen of a transversal gate applying a logical gate that is not equivalent to the physical underlying gate (albeit very related in that it is the inverse of the applied gate). Moreover, the proof of the above statement for the transversal phase gate could have also been obtained from considering the weight of the computational basis states that are in the codewords of the logical states of the code $|0\rangle_L$, $|1\rangle_L$. Namely, for the 7-qubit code the logical $|0\rangle_L$ state is composed of computational basis states with weight $0 \mod 4$ and the

logical $|1\rangle_L$ state is composed of basis states of weight 3 mod 4:

$$|0\rangle_L = \frac{1}{2^{3/2}}(I+G_1^X)(I+G_2^X)(I+G_3^X)|0\rangle^{\otimes 7} \tag{2.36}$$

$$= \frac{1}{2^{3/2}}(I+G_1^X)(I+G_2^X)(|0\rangle^{\otimes 7}+|1010101\rangle) \tag{2.37}$$

$$= \frac{1}{2^{3/2}}(I+G_1^X)(|0\rangle^{\otimes 7}+|1010101\rangle+|1100110\rangle+|0110011\rangle) \tag{2.38}$$

$$= \frac{1}{2^{3/2}}(|0\rangle^{\otimes 7}+|1010101\rangle+|1100110\rangle+|0110011\rangle + |1111000\rangle+|0101101\rangle+|0011110\rangle+|1001011\rangle), \tag{2.39}$$

$$|1\rangle_L = \frac{1}{2^{3/2}}(|1\rangle^{\otimes 7}+|0101010\rangle+|0011001\rangle+|1001100\rangle + |0000111\rangle+|1010010\rangle+|1100001\rangle+|0110100\rangle). \tag{2.40}$$

$$\tag{2.41}$$

Example 35. *The 5-qubit perfect code is a* $[[5,1,3]]$ *quantum error correcting code with the following stabilizer generators [95]:*

$$\begin{array}{ll} XZZXI & IXZZX \\ XIXZZ & ZXIXZ, \end{array} \tag{2.42}$$

with logical operators $X_L = X^{\otimes 5}$*, and* $Z_L = Z^{\otimes 5}$*.*

Notice that the 5-qubit code has stabilizers that are cyclic permutations of one another. As such, the Pauli twirl operator K will be a transversal gate for the code:

$$K = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & i \\ 1 & -i \end{pmatrix}, \tag{2.43}$$

$$KXK^\dagger = -Y, \tag{2.44}$$

$$KYK^\dagger = -Z, \tag{2.45}$$

$$KZK^\dagger = X. \tag{2.46}$$

Lemma 36. *The twirl operator* K *is transversal for the perfect* $[[5,1,3]]$ *code.*

Proof. First consider the action of the transversal twirl operation $K^{\otimes 5}$ on the stabilizers. Due to the cyclic symmetry of the stabilizer generators and of the transversal operation, without loss of generality we can consider the transformation of a single stabilizer generator.

Showing that it remains in the stabilizer group will be sufficient to showing that the group remains unchanged.

Consider the following:

$$\begin{aligned} K^{\otimes 5}(XZZXI)K^{\otimes 5} &= YXXYI \\ &= (XIXZZ)(ZXIXZ), \end{aligned} \tag{2.47}$$

therefore the transformed stabilizer is a product of two of the original stabilizers of the group and thus remains in the stabilizer group. Therefore we conclude that $\mathcal{S}$ remains unchanged.

Finally, it must be checked whether the logical operation implements the correct mapping of the logical X and Z operators. It is fairly straightforward to check:

$$K^{\otimes 5}(XXXXX)K^{\otimes 5} = -YYYYY = -Y_L, \tag{2.48}$$

$$K^{\otimes 5}(ZZZZZ)K^{\otimes 5} = XXXXX = X_L, \tag{2.49}$$

$$\tag{2.50}$$

thus completing the proof. □

Example 37. *The 15-qubit Reed-Muller code is a* $[[15, 1, 3]]$ *code with the following stabilizers [89, 129]:*

$$\begin{array}{ll} IIIIIIIXXXXXXXX & IIIIIIIZZZZZZZZ \\ IIIXXXXIIIIXXXX & IIIZZZZIIIIZZZZ \\ IXXIIXXIIXXIIXX & IZZIIZZIIZZIIZZ \\ XIXIXIXIXIXIXIX & ZIZIZIZIZIZIZIZ \\ IIIIIIIIIIZZZZ & IIIIIZZIIIIIZZ \\ IIIIIIIIZZIIZZ & IIIIZIZIIIIIZIZ \\ IIIIIIIZIZIZIZ & IIZIIIZIIIZIIIZ, \end{array} \tag{2.51}$$

and the logical Pauli operators are $X_L = X^{\otimes 15}$ *and* $Z_L = Z^{\otimes 15}$.

Lemma 38. *The CNOT and T gates are transversal for the 15-qubit code.*

Proof. The $CNOT$ is transversal since the 15-qubit code is a CSS code. In order to show the transversality of the T gate, rather than show the preservation of the stabilizer group, it

will be shown that the action of the transversal T has the desired mapping for the stabilizer states $|0\rangle_L$ and $|1\rangle_L$. As shown in Lemma 31, the logical $|0\rangle_L$ state can be expressed as:

$$|0\rangle_L = \frac{1}{\sqrt{|\mathcal{S}_X|}} \sum_{s_x \in \mathcal{S}_X} |s_x\rangle, \tag{2.52}$$

where s_x is a binary string representing the locations of Pauli X operators for every element $S_X \in \mathcal{S}_X$. It can be shown that every element in $\mathcal{S}_X$ is of weight-8, except for the identity operator of weight-0. Therefore, all elements in the sum will be of states of weight-8, and as such will remain invariant under the action of $T^{\otimes 15}$:

$$\begin{aligned} T^{\otimes 15}|0\rangle_L &= \frac{1}{\sqrt{|\mathcal{S}_X|}} \sum_{s_x \in \mathcal{S}_X} T^{\otimes 15}|s_x\rangle \\ &= \frac{1}{4}\left(|0\rangle^{\otimes 15} + \sum_{s_x \in \mathcal{S}_X s_x \neq \vec{0}} e^{i8\pi/4}|s_x\rangle\right), \qquad (2.53) \\ &= \frac{1}{4} \sum_{s_x \in \mathcal{S}_X} |s_x\rangle, \qquad (2.54) \end{aligned}$$

therefore $T^{\otimes 15}|0\rangle_L = |0_L\rangle$. On the other hand, due to the fact that $X_L = X^{\otimes 15}$, the logical state $|1\rangle_L$ will be a sum over computational basis states with weight 7 and 15, thus resulting in the following action:

$$\begin{aligned} T^{\otimes 15}|1\rangle_L = T^{\otimes 15}X_L|0\rangle_L &= \frac{1}{\sqrt{|\mathcal{S}_X|}} \sum_{s_x \in \mathcal{S}_X} T^{\otimes 15}X^{\otimes 15}|s_x\rangle \\ &= \frac{1}{\sqrt{|\mathcal{S}_X|}} \sum_{s_x \in \mathcal{S}_X} T^{\otimes 15}\left|\vec{1} \oplus s_x\right\rangle \\ &= \frac{1}{4}\left(e^{i15\pi/4}|1\rangle^{\otimes 15} + \sum_{s_x \in \mathcal{S}_X s_x \neq \vec{0}} e^{i7\pi/4}\left|\vec{1} \oplus s_x\right\rangle\right), \qquad (2.55) \\ &= \frac{e^{i7\pi/4}}{\sqrt{|\mathcal{S}_X|}} \sum_{s_x \in \mathcal{S}_X} \left|\vec{1} \oplus s_x\right\rangle \\ &= e^{-i\pi/4}|1\rangle_L. \qquad (2.56) \end{aligned}$$

Therefore, $T^{\otimes 15}|1\rangle_L = e^{-i\pi/4}|1\rangle_L$, which means that the transversal T gate results in a logical action of $T_L^\dagger$. One could equivalently apply the logical T gate by applying $T^\dagger$ transversally. □

The above three codes exhibit all different classes of transversal gates, however none of the transversal gate sets for a given error correcting code are universal for quantum computation.

Definition 39. *A finite gate set $\mathcal{G} = \{G_1, G_2, \ldots, G_n\}$ is called a universal gate set for a Hilbert space $\mathcal{H}$ if for any unitary transformation U and error rate ϵ there exists a decomposition using $\mathcal{O}(\log^c(\frac{1}{\epsilon}))$ gates in $\mathcal{G}$, where c is a constant.*

In fact, it was shown that any transversal gate set for a given quantum error correcting code can only generate a finite gate set and thus cannot be universal. A similar result was obtained for quantum error correcting codes generated from the group $GF(4)$ (stabilizer codes belong to this class). These results are summarized in the theorem below.

Theorem 40 (Eastin–Knill Theorem [52, 142])**.** *Given a quantum error correcting code $\mathcal{Q}$, then the set of transversal gates $\mathcal{G}_T$ cannot be universal for quantum computation.*

This result creates a problem for fault-tolerant theory as transversal operations form the most natural realization of fault-tolerant operations, and the idea of using a gate that is non-transversal for a given code would seem to break fault-tolerance. We will return to using non-transversal gates later in this chapter, however there are methods to circumvent the lack of a universal gate using special ancillary state preparation.

2.1.2 Magic state distillation

2.1.2.1 Gate teleportation

In Section 1.2.1.6, the state teleportation algorithm was introduced using an entangled qubit between two parties to transport the information from one party to another. It turns out that there is a very similar scheme called *gate teleportation* that will be useful in fault-tolerant quantum computation. The scheme was developed by Gottesman and Chuang and is presented below [61].

Quantum gate teleportation is identical to quantum state teleportation from Alice's perspective. What changes is that prior to receiving Alice's message, Bob performs the unitary U on his half of the entangled pair and follows this by applying the unitary $UPU^\dagger$ on his qubit based on the resulting measurement from Alice (where P is the Pauli operation that would have been performed in the usual state teleportation scheme). At first this seems to provide no advantage, Bob is simply applying the unitary, undoing it, applying

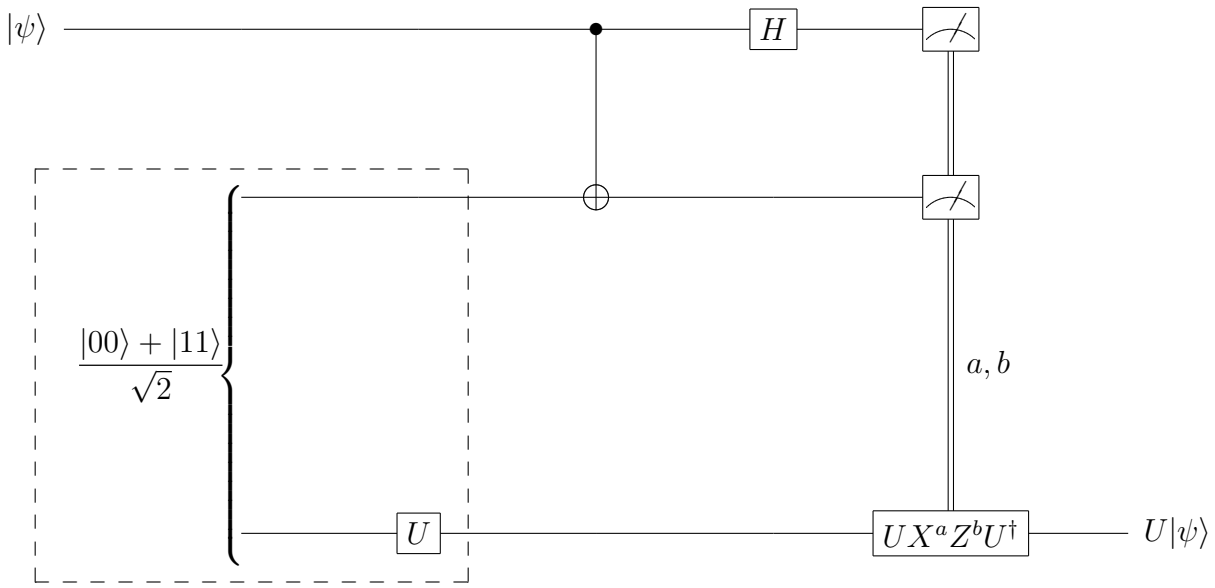


Figure 2.1: Protocol for quantum gate teleportation [61]. Alice and Bob each share one half of a Bell pair and Alice begins with the arbitrary state used in the computation. Alice performs a Bell measurement on her two qubits, while Bob applies the unitary U to his half of the Bell pair. Alice then sends the result of the measurement to Bob who in turn performs a controlled unitary $UX^aZ^bU^\dagger$ that depends on the measurement result (a, b). The resulting final state on Bob's side is $U|\psi\rangle$.

the Pauli, and applying the unitary again. Why not wait until the state teleportation scheme has completed before applying his unitary? The important observation is that the action of the unitary U is on the Bell state that Alice and Bob share and this is particularly important when the unitary is noisy!

Taking a step back and thinking about the different elements of state teleportation, there is the preparation of the entangled state between Alice and Bob, the measurement in the Bell basis for Alice, and the controlled Pauli gate application for Bob depending on Alice's measurement and message. Gate teleportation contains the same basic elements, except that the prepared state is no longer necessarily a Bell pair, but rather a different entangled state, Alice's measurement is the same, and Bob now performs a different controlled unitary, which could potentially be easier than applying U itself! This is the key aspect, that the application of U is pushed to the state preparation, and Bob now has to perform the gate $UPU^\dagger$ on his copy of the state which may be easier to perform than U itself.

2.1.2.2 The Clifford hierarchy

As eluded to in the previous section, gate teleportation becomes useful primarily in situations when performing the gate $UPU^\dagger$ is easier than performing U. This has its most obvious impact in the implementation of unitaries from the Clifford hierarchy.

Definition 41. *Let $\mathcal{C}_n^{(1)} := \mathcal{P}_n$ be the Pauli group on n qubits as well as the first level in the **Clifford hierarchy**. The higher levels of the Clifford hierarchy are then defined recursively as follows:*

$$\mathcal{C}_n^{(l)} := \{U \in \mathbb{U}(\mathcal{H}^{\otimes n}) | UPU^\dagger \in \mathcal{C}_n^{(l-1)},\ \forall\ P \in \mathcal{P}_n\}, \tag{2.57}$$

*where $\mathbb{U}(\mathcal{H}^{\otimes n})$ is the set of unitaries in the n qubit Hilbert space. The set of operators in $C_n^{(2)}$ are denoted the **Clifford gates**, mapping Pauli operators to Pauli operators.*

Note that from the definition of the Clifford hierarchy, unitaries that reside in the Clifford hierarchy imply that the unitary that Bob has to apply, $UPU^\dagger$, is one level lower in the hierarchy. This fact becomes particularly important in the case when $U \in \mathcal{C}_1^{(3)}$ that is a single qubit gate at the third level of the hierarchy, then every element in the gate teleportation scheme remains Clifford except for the state preparation protocol (note that measurement in the computational basis is considered a Clifford operation here, as it is included in the class of operations that are efficiently simulatable using a classical computer by the Gottesman-Knill Theorem [59]).

Relating this notion back to quantum error correction, we have already seen examples of a code that has a generating set of transversal gates for the Clifford group, that is $\{H, S, CNOT\}$ in the case of the 7-qubit code. Also, computational basis measurement can be performed transversally in such a code using techniques such as Shor error correction, Steane error correction, or Knill error correction [126, 130, 87]. Moreover, adding any single qubit gate from outside the Clifford group will generate a universal gate set for quantum computation. Therefore, by applying gate teleportation to a unitary $U \in \mathcal{C}_1^{(3)}$ will allow for the application of a universal set of operations, assuming that one can prepare the logical state $(I \otimes U)(|00\rangle + |11\rangle/\sqrt{2}$ reliably since $UPU^\dagger$ is Clifford and transversal.

2.1.2.3 Distilling the magic state

It has been shown in the last few sections that the problem that the Eastin-Knill Theorem 40 poses on the impossibility of transversal universality can be sidestepped by having the ability to prepare a special logical state of the form $(I \otimes U)(|00\rangle + |11\rangle)/\sqrt{2}$, where U is a gate from $\mathcal{C}_1^{(3)}$ and the underlying code can implement the full Clifford group transversally[1]. Preparing any eigenstate of a stabilizer group can be done in a fault-tolerant manner using fault-tolerant measurement techniques, which can be done to arbitrary precision assuming the underlying gates are below threshold. Such states shall be referred to as stabilizer states. However, the state $(I \otimes U)(|00\rangle + |11\rangle)/\sqrt{2}$ is not a stabilizer state and thus given an error correcting code with only transversal Clifford operations, it is unclear how such a state can be prepared in a fault-tolerant manner.

The most commonly used gate to complete the universal gate set with the Clifford operations is the $\pi/8$-gate, or T gate:

$$T = \begin{pmatrix} e^{-i\pi/8} & 0 \\ 0 & e^{i\pi/8} \end{pmatrix} = e^{-i\pi/8} \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} \tag{2.58}$$

. Up to global phases, the T gate induces a phase of $\pi/4$ on the $|1\rangle$ state, we shall refer to this version of the gate (without global phase) for the remainder of the thesis. The following identities should be noted for the T gate:

$$TZT^\dagger = Z \tag{2.59}$$

$$TXT^\dagger = \frac{1}{\sqrt{2}}(X + Y), \tag{2.60}$$

[1]While thus far only the 7-qubit code has been discussed in relation to transversal Clifford operations, this will hold for another very important family of topological error correcting codes, the color codes.

and it can be verified that the operator $(X+Y)/\sqrt{2}$ is indeed in the Clifford group and as such $T \in \mathcal{C}_1^{(3)}$. Therefore, $\mathcal{C}_n^{(2)} + T$ is universal for quantum computation and moreover, by using gate teleportation, universal quantum computation can be obtained from reliably preparing the state $(I \otimes T)\ (|00\rangle + |11\rangle/\sqrt{2}$, or more simply given that $CNOT$ is assumed to be fault-tolerant and as such have low noise, preparing the state $T(|0\rangle + |1\rangle)/\sqrt{2}) = (|0\rangle + e^{i\pi/4}|1\rangle)/\sqrt{2}$ is sufficient.

To recap, assuming the error correcting code that is being used allows for the application of all Clifford operations in a transversal manner, and thus fault-tolerant, the error on such gates can be made arbitrarily small. Unfortunately, this is not sufficient for universal quantum computation, and gate teleportation would allow to complete the universal gate set assuming one could fault-tolerantly prepare the state $(|0\rangle + e^{i\pi/4}|1\rangle)/\sqrt{2}$.

This section reviews the 15-qubit state distillation scheme, originally presented by Bravyi and Kitaev [32]. Other distillation schemes do exist that exhibit different fault-tolerant properties, we refer to them at the end of this section. State distillation begins with the preparation of many copies of the noisy logical state, denoted ρ. Defining the state $|H\rangle = (|0\rangle + e^{i\pi/4}|1\rangle)/\sqrt{2}$, note that $|H\rangle$ is an "+1" eigenstate of the Clifford operator $TXT^\dagger$ while $|H_\perp\rangle = (|0\rangle - e^{i\pi/4}|1\rangle)/\sqrt{2}$ is a "-1" eigenstate. As such, by applying the probabilistic map of the identity and $TXT^\dagger$ with even weight, we can dephase the noisy state ρ in this basis (recall that $TXT^\dagger$ is Clifford and as such can be applied with high accuracy):

$$\rho \to \rho_H = \frac{1}{2}(\rho + TXT^\dagger \rho TXT^\dagger) = (1-\epsilon)|H\rangle\langle H| + \epsilon|H_\perp\rangle\langle H_\perp|, \tag{2.61}$$

where ϵ is now a single parameter that characterizes the level of noise in the initial encoded state.

The idea behind $|H\rangle$-type magic state distillation is that by taking 15 copies of the noisy state, the overlap of the joint state with the codespace of the 15-qubit Reed-Muller code will yield a state with higher fidelity in the $|H\rangle$ state assuming ϵ is below some threshold value. Moreover, in order to project the joint state onto the 15-qubit codespace, the decoder of the 15-qubit code along the computational basis measurements need to be applied, all of which belong to the Clifford group and can be applied with high accuracy. Let Π_{15} be the projector onto the eigenspace of the 15-qubit code and $\mathcal{D}_{15}$ be the decoder of the code, this projection can be implemented using Clifford operations and post-selection. Then the resulting state after the projection and decoding will be diagonal in the $\{|H\rangle, |H_\perp\rangle\}$

basis [32]:

$$\mathcal{D}_{15} \circ \left(\Pi_{15} \rho_H^{\otimes 15} \Pi_{15}\right) = (1-\epsilon')|H\rangle\langle H| + \epsilon'|H_\perp\rangle\langle H_\perp|, \tag{2.62}$$

$$\epsilon' = \frac{1 - 15(1-2\epsilon)^7 + 15(1-2\epsilon)^8 - (1-2\epsilon)^{15}}{2(1+15(1-2\epsilon)^8)}, \tag{2.63}$$

where the states are post-selected on having projected onto the codespace of the 15-qubit code, which occurs with probability:

$$P_{\text{succes}} = \frac{1 + 15(1-2\epsilon)^8}{16}. \tag{2.64}$$

The output state will therefore be a less noisy state assuming that $\epsilon' < \epsilon$, which occurs for all $\epsilon < 0.14$. Moreover, by repeating the procedure, multiple copies of the less noisy state can be fed back into the same distillation process and the error rate can be reduced to arbitrarily small levels in a very rapidely. In fact, the error rate of the distilled state is suppressed double-exponentially in the number of levels of distillation, overcoming the exponential growth in the number of noisy qubits required for such a distillation. In practice, it will be rare that a computation would require more than three levels of distillation. The process is called *magic state distillation* as not only can the $|H\rangle$ state be distilled allowing for the application of a universal set of quantum gates, the distillation can be done using only Clifford operations in the first place!

Is the problem of universal fault-tolerant quantum computation then solved? Magic state distillation certainly provides a means towards such a universal set of operations. However, the primary drawback of magic state distillation is that while it is technically efficient in terms of the rate at which the noise is suppressed, the number of noisy *logical* states that need to be prepared is still very large, for even a few distillation levels. Moreover, in gate compilation schemes using the aforementioned universal gate set, typically the T gate is applied approximately half of the time [86], therefore there will always have to be an active ancillary factory ready for each logical qubit. Despite many efforts to reduce the overhead in magic state distillation [104, 80, 31, 79], studies have assessed that in schemes where magic state distillation may be most useful, such as in completing the universal gate set for the Toric code [85], the overhead in terms of physical qubit number would be close to 90% of the total physical qubit space [55]. Therefore, developing techniques to circumvent the need for magic state distillation and being able to complete the universal gate set has become a very active area of research in recent years.

2.2 Universal fault-tolerance via code concatenation

The ideas presented in this section are based on the research that appeared in Ref. [78], copyrighted by the American Physical Society. I developed the idea for this project and both authors contributed to the final scheme which was developed to realize the research goal of the project. I wrote the manuscript and both parties contributed to the editing of the work for publication.

2.2.1 Overview

This work presents an alternative approach to obtaining a universal set of transversal gates by using concatenated quantum error correcting codes, rather than state distillation. The scheme proposed uses two different quantum error correcting codes, with differing sets of transversal operations, in concatenation. We argue that by sacrificing the full distance of the concatenated quantum error correcting code, we can exploit the transversal properties of both quantum codes to produce a set of operations that, while not globally transversal, provide a means for universal fault-tolerant quantum gates. In this work we shall focus on protecting against arbitrary single-qubit errors, however, we provide a brief description of how such a scheme could be generalized to correct against t errors. Recently, Paetznick and Reichardt [114] have proposed a similarly motivated work on universal quantum fault-tolerance without the preparation of special ancillary states and their idea was further developed to a topological setting by Bombín [23]. In their scheme, additional transversal measurements and error correction are introduced after the action of the transversal Hadamard gate in order to recover the codespace. The presented scheme differs from such a construction in that it does not require an additional round of error correction since the logical gates do not disrupt the codespace as they are not necessarily transversal. This comes at the expense of requiring an additional concatenated code for protection. Additionally, there has been research that focused on obtaining a set of fault-tolerant operations to transfer between different quantum error correcting codes, yet such schemes do not yield a set of universal operations [131, 69, 46].

2.2.2 Concatenated QEC

We say a logical gate G on a quantum error correcting code $\mathcal{C}$ is called t-transversal if G interacts with at most t locations of the underlying qubits composing the code $\mathcal{C}$. The general concatenated error correcting scheme is as follows: the qubits that we desire to

protect against errors are encoded into a quantum error correcting code $\mathcal{C}_1$. In this work, we shall require the code distance of $\mathcal{C}_1$ to be at least 3, so that it can correct arbitrary single-qubit errors. The qubits that make up the code $\mathcal{C}_1$ are subsequently encoded into a second code $\mathcal{C}_2$, which again will be required to have distance of at least 3. The general layout of the scheme is summarized in Figure 2.2. As we are focusing on codes that correct for an arbitrary single-qubit error, we shall refer to a transversal gate for a given code as a gate which is 1-transversal and any gate not having this form as non-transversal.

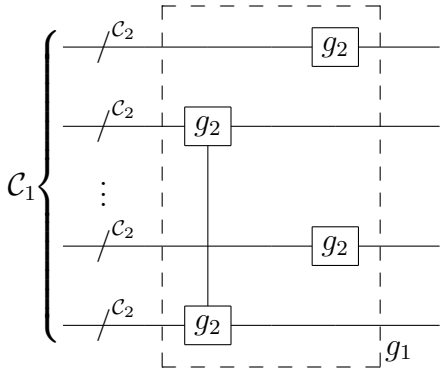


Figure 2.2: General construction of a logical gate for a concatenated error correction scheme. The qubit of information is encoded in a quantum error correcting code $\mathcal{C}_1$, whose qubits are in turn encoded into a code $\mathcal{C}_2$. As such, the logical gate g_1 (given by the boxed region) on the encoded space $\mathcal{C}_1$ will be composed of multiple logical gates g_2 on the $\mathcal{C}_2$ codeblocks.

The important properties for the quantum error correcting codes $\mathcal{C}_1$ and $\mathcal{C}_2$ for the implementation of universal fault-tolerant quantum logic are as follows: 1. For any logical gate that is non-transversal in $\mathcal{C}_1$, there must exist an application of this logical gate using gates that are transversal in $\mathcal{C}_2$, 2. The recovery operations (syndrome measurement and error correction operations) on $\mathcal{C}_1$ and $\mathcal{C}_2$ must be globally transversal (in the full concatenated codespace).

Since there exists no quantum error correcting code that exhibits a full set of transversal quantum gates [52, 142], there will always be at least one gate in a given universal gate set that will couple qubits that make up the error correcting code, leading to the possibility of bad error propagation. Consider the first level of encoding $\mathcal{C}_1$, the non-transversal gate can lead to a propagation of errors, however, if we are now encoding each of the qubits making up the code $\mathcal{C}_1$ into a further error correcting code, the propagation from a single

to multiple physical faults will not necessarily lead to a propagation of logical faults if the errors are sufficiently sparse.

Specifically, the first requirement of the concatenated quantum error correction scheme stipulates that every non-transversal gate in the code $\mathcal{C}_1$ can be implemented using transversal gates in the code $\mathcal{C}_2$. The non-transversality of a given gate will cause the propagation of a single physical fault between different logical qubits in $\mathcal{C}_1$. The implementation of the non-transversal $\mathcal{C}_1$ gates will govern the propagation of the physical errors between the qubits. Therefore, we require the gates that make up the logical gate on $\mathcal{C}_1$, themselves logical gates for the code $\mathcal{C}_2$, to be transversal in the encoded space $\mathcal{C}_2$. By imposing such a restriction, a single error occurring in the non-transversal gate application in $\mathcal{C}_1$ will propagate to at most a single physical error in each of the logical qubits forming $\mathcal{C}_1$, which themselves are encoded blocks of $\mathcal{C}_2$. This is precisely the property one is after in a fault-tolerant quantum computation, that a single physical error will propagate to at most a single physical error on encoded codeblocks, allowing for the correction of such errors.

Given a choice of codes $\mathcal{C}_1$ and $\mathcal{C}_2$, not all gates of the universal gate set will be transversal in $\mathcal{C}_2$. By the properties outlined above, any logical gate in $\mathcal{C}_1$ that uses gates from $\mathcal{C}_2$ that are not transversal in its construction, must be transversal in $\mathcal{C}_1$. In performing such a gate, a single fault on a particular $\mathcal{C}_2$ codeblock could propagate to multiple errors within this codeblock and could lead to a logical $\mathcal{C}_2$ fault in the codeblock where the error occurred. However, a single logical fault on one of the $\mathcal{C}_2$ codeblocks will not yield a global logical fault on $\mathcal{C}_1$, as such a code can correct for arbitrary logical faults on one of its encoding logical qubits.

The concatenation scheme therefore protects all gates in the universal gate set. The scheme circumvents the result claiming that no universal gate set can be implemented transversally [52, 142] by not implementing the gates in a strict transversal manner. Rather, the gates are implemented such that errors spread to locations that are further protected by an additional code through concatenation.

How is error correction then applied? We shall describe the error correction properties that are required after the application of two types of logical quantum gates, those that are non-transversal in $\mathcal{C}_1$ yet use an application of transversal $\mathcal{C}_2$ gates, and the application of logical gates that are transversal in $\mathcal{C}_1$, whose individual block gates are non-transversal in $\mathcal{C}_2$. In the case of the former, the important property of the error correction is that it does not couple qubits within the codeblocks of $\mathcal{C}_2$, as the application of the gate could propagate a single fault into multiple single faults on each of the $\mathcal{C}_2$ codeblocks. If the error correction procedure propagates errors within the $\mathcal{C}_2$ codeblocks, then single errors on each codeblock will propagate to multiple errors on each codeblock, thus possibly leading to logical errors

on multiple codeblocks, therefore causing a global logical fault. As such, it is very important that the error syndrome measurement and correction be performed transversally on each of the $\mathcal{C}_2$ codeblocks. Error correction at the $\mathcal{C}_1$ level is not necessary after the application of this type of logical gate, as the errors propagate within the codeblocks and the scheme is constructed in a way that all such errors on the codeblocks are recoverable as long as only a single fault occurs in the application of the gate.

The error correction procedure after the implementation of the transversal gate in $\mathcal{C}_1$ (using non-transversal $\mathcal{C}_2$ gates) will require an additional level of error correction. As in the application of the non-transversal $\mathcal{C}_1$ gates, error correction on each of the $\mathcal{C}_2$ codeblocks is first applied. As the application of the logical gate on $\mathcal{C}_1$ uses non-transversal $\mathcal{C}_2$ gate applications, a single (correctable) error on a particular $\mathcal{C}_2$ codeblock can propagate to a non-correctable set of errors on that given codeblock. As such, performing the $\mathcal{C}_2$ error correction on that codeblock will introduce a logical error (if the error were to occur during the $\mathcal{C}_2$ error correction process itself then this error will be weight one). However, as mentioned above, if only a single logical $\mathcal{C}_2$ error has occurred, the logical fault introduced by the error correction will be correctable using error correction procedure on $\mathcal{C}_1$. However, it is important that the error correction procedure on $\mathcal{C}_1$, which is a logical error correction procedure as it acts on logically encoded states in $\mathcal{C}_2$, is itself globally transversal. As such, errors that could occur during error correction will not propagate to multiple physical errors that could be detrimental upon the application of further logical computation.

2.2.3 Example: A 105-qubit quantum error correcting code

A simple example of the scheme outlined in this work involves two of the most well studied quantum error correcting codes, $\mathcal{C}_1$ will be the 7-qubit Steane code [128] and $\mathcal{C}_2$ the 15-qubit Reed-Muller code [89]. The 15-qubit Reed-Muller code has the following set of transversal gates: $\{T, CNOT\}$, where each logical gate is achieved by applying the gate itself to each of the qubits (or $T^\dagger$ in the case of the logical T gate). The missing gate from the universal gate set is the Hadamard gate. The 7-qubit Steane code (corresponding to $\mathcal{C}_1$) has the following set of transversal gates: $\{S, H, CNOT\}$, where $S = |0\rangle\langle 0| + i|1\rangle\langle 1|$. Each logical gate is achieved by applying an individual gate to each of the qubits, or pair of qubits (in the case of applying logical S, one applies $S^\dagger$ to each of the qubits). As such, $\mathcal{C}_1$ can implement gates from the Clifford group transversally, yet are missing the T gate from the universal gate set that can be implemented transversally.

The concatenated code is 7 blocks of 15 qubits, totalling 105 qubits, encoding 1 qubit of information. As both quantum codes share the property that all Pauli gates, the S

phase gate, and the $CNOT$ gate can be implemented logically by applying the gate to each qubit, or pair of qubits, then the globally logical version of these gates for the 105 qubit code are also achieved by applying the corresponding gate to each qubit, or pair of qubits, of the full 105 qubit code. Additionally, all syndrome measurements (which will correspond to the measurement of Pauli observables) will be transversal within the code, as well as the Pauli corrections.

The logical T gate is achieved by combining logical gates on the different $\mathcal{C}_2$ codeblocks, which as shown in Figure 2.3 is not transversal in the $\mathcal{C}_1$ code, yet uses gates that are all transversal within the 15 qubit $\mathcal{C}_2$ codeblocks. As explained in the previous section, a single error in the implementation of the logical gate can propagate to multiple errors (a maximum of 3 for this particular gate application) yet will be distributed such that there are only single errors on each $\mathcal{C}_2$ codeblock. The error correction procedure measures the syndromes on each of the $\mathcal{C}_2$ codeblocks individually, which corresponds to measuring the 14 syndromes corresponding to the 15-qubit code. The Pauli error correction operations are then applied to correct for the errors that occurred during the application of the logical T gate. As such, the concatenated code can correct for an arbitrary weight-1 error that occurs during the implementation of the logical T gate.

In order to implement the logical H, one applies the logical H_{15} on each of the $\mathcal{C}_2$ codeblocks, as such it is transversal in the encoded states that form the code $\mathcal{C}_1$, yet each individual H_{15} is not transversal in its implementation on the $\mathcal{C}_2$ codeblocks. A single error that occurs in one of the individual applications of the H_{15} gates could propagate to multiple errors within the codeblock, leading to possible logical errors. However, if only one such error occurs the full quantum code will still be protected. After the action of the gate, error correction is applied to each of the $\mathcal{C}_2$ codeblocks, possibly resulting in correction causing a $\mathcal{C}_2$ logical fault. However, if only one such logical $\mathcal{C}_2$ fault occurs, subsequent global error correction at the logical $\mathcal{C}_1$ level will detect such an error. The $\mathcal{C}_1$ error correction involves measuring the 6 stabilizers of the 7-qubit Steane code, where each stabilizer is now a logical stabilizers composed of X_{15} or Z_{15} operators. However, as such operators are transversal for the 15-qubit code, they can be measured in a transversal way. The maximal weight of the stabilizers measured for $\mathcal{C}_1$ code is 32, since each of the logical X_{15} gates involve 8 X gates on the $\mathcal{C}_2$ codeblock, and the weight of the 7-qubit X stabilizers is 4. Error correction for the $\mathcal{C}_1$ level will then be completed by performing logical Pauli error correcting operations on affected $\mathcal{C}_2$ codeblocks. The measurement of the $\mathcal{C}_1$ stabilizers will be the most expensive error correction step due to the high weight of the stabilizers.

As described, the concatenated code can correct for any weight one error. However, it is worth noting that if one used a straight concatenation of the two codes to protect

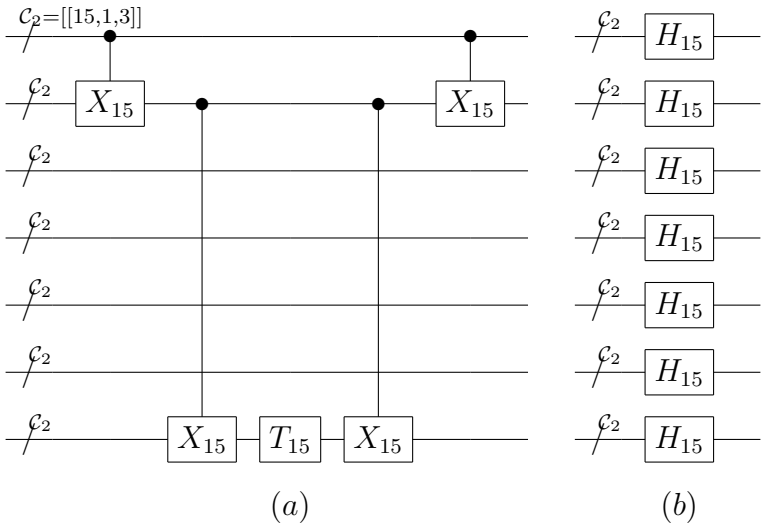


Figure 2.3: (*a*) Logical T gate for the Steane code $\mathcal{C}_1$, composed of logical $CNOT_{15}$ and T_{15} gates on the $\mathcal{C}_2$ codeblocks. These gates are transversal in $\mathcal{C}_2$, and therefore only propagate errors to different codeblocks, without propagating within a given $\mathcal{C}_2$ codeblock. (*b*) Logical H gate for the Steane code $\mathcal{C}_1$, implemented using logical H_{15} gates on each of the $\mathcal{C}_2$ codeblocks. Note that the individual H_{15} are non-transversal on each codeblock.

against quantum noise, the concatenated code will be a [[105,1,9]] quantum error correcting code, that is, it would protect against 4 arbitrary errors. In this fault-tolerant scheme, we are sacrificing the larger distance of a straight concatenation scheme in order to protect against arbitrary single qubit errors when performing logical gates.

2.2.4 Summary & outlook

In this work we have proposed a method for universal quantum fault-tolerance using concatenated error correcting codes. The full distance of the concatenated scheme is sacrificed in order to establish a set of universal quantum gates that are robust to a smaller set of errors. The transversal properties of the two different error correction schemes are exploited to limit the propagation of errors to either be sufficiently sparse, only a small number of errors per encoded codeblock, or limiting all errors to be contained within a single codeblock.

The scheme described in this work could be adapted to account for quantum error

correcting codes $\mathcal{C}_1$ and $\mathcal{C}_2$ that correct against arbitrary weight t errors. The key properties of universal gate sets developed for such a concatenation scheme would be modified such that given a gate which is not t-transversal in $\mathcal{C}_1$, the logical gates in $\mathcal{C}_2$ which form such a gate must be t-transversal in $\mathcal{C}_2$ when applied in composition. Additionally, similar requirements for the quantum error correction operations would be necessary. The error correction operations should be 1-transversal as to not possibly propagate errors that occur during the error correction process to multiple errors that could be detrimental at the next stage of computation. The presented scheme could be generalized to arbitrary distance codes by concatenating the higher distance 2D and 3D color code, of which the 7-qubit Steane and 15-qubit Reed-Muller are the lowest non-trivial code in each respective dimension.

At first it may seem that the presented scheme would have quite a low fault-tolerance threshold due to the high number of physical qubits for such a small global distance. The non-transversal nature of the logical Hadamard gate on each of the 15-qubit codeblocks will introduce many potential fault locations which would seem to indicate a high probability of logical errors arising unless the physical error rate is very low. However, as shown in the next section, the fault-tolerance threshold can actually be higher than expected due to the form of logical noise for each of the respective logical gates. As such, the threshold is comparable to many other concatenated schemes [113]. Moreover, it should be noted that in the presented example, the protection of the code $\mathcal{C}_2$ is really only needed on three of the codeblocks, the logical qubits that are involved in the non-transversal T gate application. Therefore, the 105-qubit code could, in theory, be replaced by a 49-qubit encoding (3 encoded blocks of 15 qubits, and 4 single qubits) and still have a fault-tolerant set of universal logical gates [2]. This provides a means to reduce the number of faulty locations and increases the theoretical threshold for some noise models [37].

2.3 Thresholds for universal concatenated codes

The material from this section is based on the journal article of Ref. [38], copyrighted by the American Physical Society. In this work, Christopher Chamberland and myself were the primary investigators of the research, developed the framework of the fault-tolerant numerical tests and wrote of the manuscript. I was involved in the development of the simulation code, and advised Christopher Chamberland in the writing of much of the code

[2] I would like to acknowledge Jonas Anderson, Chris Cesare, and Guillaume Duclos-Cianci for interesting discussions regarding this construction as well as a recent parallel work that outlines this encoding [110]

in the latter part of the project. All three authors of Ref. [38] contributed to the analysis of the results and the editing of the manuscript.

2.3.1 Fault-tolerance threshold under depolarizing noise

In this work, we establish a lower bound on the fault-tolerant threshold for the 105-qubit universal concatenated code under depolarizing noise. We show that the dual protection coming from the concatenation of two different error correcting codes provides more than just minimal fault-tolerant protection, it also serves as a means for logical error suppression at the second level of concatenation (and above). We believe that this provides new insights in the development of quantum error correcting codes and emphasizes an important principle: to logically protect the quantum gates that are most present in the fault-tolerant architecture.

The key property of fault-tolerant architectures is the presence of an *asymptotic threshold*. For concatenated coding schemes, the asymptotic threshold is the physical error rate p_{th} such that for physical error rates $p < p_{th}$ the logical error rate can be made arbitrarily small for sufficiently large number of concatenation levels (and the overall time/space resource overhead scales as $\mathcal{O}(\text{poly}(\log(A/\epsilon))A)$, where A would be the required resources for a noiseless circuit).

All currently known fault-tolerant schemes for quantum logic require active error correction between logical gates. Error correction steps are interleaved between the implementation of various fault-tolerant gates. In this study, fault-tolerant syndrome measurement and error correction is implemented using Steane's method [130], see Fig. 2.4. At a given concatenation level, each component of the logical circuit (gates and error detection/measurement) will be themselves composed of many operations from the previous level of concatenation. These components include state preparation and measurement, logical gates and memory locations. We consider a depolarizing model for each physical location (level-0) in the circuit. Depolarizing noise is modelled in a similar manner to that of Paetznick and Reichardt's study of the 23-qubit Golay code [113]. Each single qubit gate (including resting qubits) undergoes Pauli noise with probability $p/4$ for each Pauli operation, and each two-qubit gate undergoes two-qubit Pauli noise with probability $p/16$ for each non-trivial two-qubit Pauli. Under this noise, state preparation in the stabilizer Z (X) basis is flipped from $|0\rangle$ ($|+\rangle$) to $|1\rangle$ ($|-\rangle$) with probability $p/2$. Similarly, measurement in the stabilizer Z (X) basis is flipped with probability $p/2$.

As first proposed by Aliferis *et al.* [2] we analyze logical gates by considering the whole as an extended rectangle (*exRec*), that is the logical gate itself along with its leading (LEC)

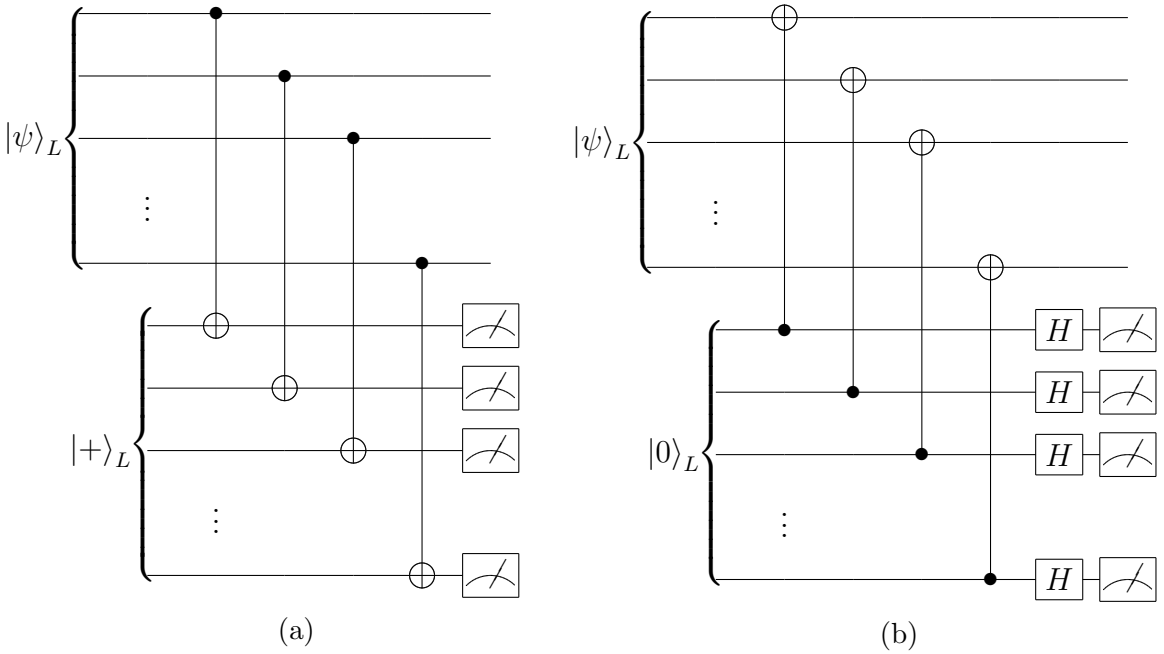


Figure 2.4: Steane error correction scheme for detecting (a) X and (b) Z errors. A logical stabilizer state is prepared fault-tolerantly (not shown in Figure) and is coupled to the logical state using a transversal $CNOT$. The transversal $CNOT$ has no effect on the logical state, however errors of the desired type are propagated onto the ancillary state for detection.

and trailing (TEC) error correction circuits (see Figure 2.5). In order to characterize the rate at which logical errors occur, we define *malignant* error events. Let $|\psi_1\rangle$ be a single or two-qubit logical state obtained by applying ideal decoders immediately after the LEC circuit and $|\psi_2\rangle$ the state obtained by applying ideal decoders immediately after the TEC. We define the event mal_E as $|\psi_2\rangle = EU|\psi_1\rangle$ where E is a single or two-qubit logical Pauli error and U is the desired gate. We denote the malignant logical error E present at the output of the circuit by mal_E. In what follows we will be interested in obtaining estimates of the probability that the event mal_E occurs for the CNOT, Hadamard and T gate.

We use Monte-Carlo sampling in order to determine the probability of each malignant event given an underlying physical depolarizing model. Given N simulations of the logical gate G at a physical error rate p, we track the number of malignant faults $a_E(p)$ of each error type E, and estimate the probability of a given logical fault as $\Pr[\text{mal}_E|G,p] = a_E/N$. The estimate of $\Pr[\text{mal}_E|G,\ p]$ improves as the number of iterations N increases by reducing

Figure 2.5: Extended rectangle consisting of leading and trailing error correcting circuits implementing the desired logical gate G.

the standard deviation. Using a least-squares fitting to determine the error probability as a function of input depolarizing error rate, we can determine the *pseudo-threshold* for each of the logical operations for our error-correcting code. For a level-1 exRec encoding the logical gate G, we define the pseudo-threshold as the crossing point $p = p_G^{(1)}(p)$, where $p_G^{(1)}(p) = \sum_{E_i} \Pr[\text{mal}_{E_i}|G, p]$ for all possible logical Pauli errors E_i for a given logical gate G. Intuitively, the pseudo-threshold corresponds to the error rate below which the logical error rate at level-1 is guaranteed to be lower than the physical error rate. In all previously studied error correction codes, the pseudo-threshold was conjectured to be an upper bound on the asymptotic threshold [133, 113]. In this work we show that this intuitive bound does not necessarily have to hold and that the asymptotic threshold can be much larger than the pseudo-threshold. To our knowledge this is the first exhibition of this type of logical error behaviour and is fundamentally related to the structure of the underlying 105-qubit error correcting code.

At each location of the level-one exRec, errors are introduced following the depolarizing noise model with noise strength p. Since the logical gates in question are fault-tolerant, a logical fault can only occur if a sequence of failures occur at the physical level. Namely, we can upper bound the failure probability for each logical fault E as follows:

$$\Pr[\text{mal}_E^{(1)}|G, p] \leq \sum_{k=\lceil \frac{d^*}{2} \rceil}^{L_G} c(k) p^k =: \Gamma_G^{(1)}, \tag{2.65}$$

where the coefficients $c(k)$ are positive integers that parametrize the number of possible weight-k errors that can lead to a logical fault, L_G is the total number of circuit locations in the logical gate G, and d^* characterizes the minimal distance of a given logical gate (that is $\lceil d^*/2 \rceil$ is the minimum weight error that must occur to produce a logical fault). For example in the 105-qubit code, the logical CNOT gate has $d^* = 9$, while the Hadamard and T logical gates have $d^* = 3$ since they sacrifice some of the distance of the code due to the fact that they are not globally transversal. As was shown in [113], the polynomial $\Gamma^{(1)}(p)$ is monotone non-decreasing making its construction straightforward with the role of upper bounding the logical error probabilities of all the logical operations G at level-1.

We can then generalize this notion to the level l concatenation level, where each of the physical locations are replaced by logical exRec locations of the $(l-1)$ level. Taking the

worst case error rate for the $(l-1)$ logical components, the error rate of logical gates at the l-th concatenation level can be bounded as follows:

$$\Pr[\mathrm{mal}_E^{(l)}|G,p] \leq \sum_{k=\lceil \frac{d^*}{2} \rceil}^{L_G} c(k) \left(\Gamma_G^{(l-1)}\right)^k =: \Gamma_G^{(l)}, \tag{2.66}$$

where the polynomials given by the coefficients $c(k)$ remain the same as the logical gate is composed of the same operations, just replacing physical locations with logical exRecs from the previous concatenation level.

Finally, we generalize a claim of Ref. [113] required to show the suppression of errors for level-2 and higher concatenation levels when below the fault-tolerance threshold p_{th}.

Lemma 42. *Suppose the error rate of a logical gate G at the l-th concatenation level can be upper bounded as follows:*

$$Pr[mal_E^{(l)}|G,p] \leq \Gamma_G^{(l)} = \sum_{k=\lceil \frac{d^*}{2} \rceil}^{L_G} c(k) \left(\Gamma_G^{(l-1)}\right)^k. \tag{2.67}$$

If the upper bounding error polynomial satisfies the following $\Gamma^{(l)} \leq \epsilon \Gamma^{(l-1)}$ for $0 \leq \epsilon \leq 1$, then the following holds:

$$Pr[mal_E^{(m)}|G,p] \leq \Gamma_G^{(m)} \leq \epsilon^{\sum_{r=0}^{m-l} \lceil \frac{d^*}{2} \rceil^r} \Gamma_G^{(l-1)}, \tag{2.68}$$

where $m > l$, and d^* is the minimal distance of the encoded state throughout the logical application of the gate G.

Proof. We shall show this result by induction. Therefore, consider first the case of $m = l+1$. By definition $\Pr[\mathrm{mal}_E^{(l+1)}|G,p]) \leq \Gamma_G^{(l+1)}$, for all logical errors E. In order to show the right side of the inequality given in Eq. 2.68 consider the expansion of $\Gamma_G^{(l+1)}$ as a sum over

failures of the gates at the (l)-th level, and use the claim that $\Gamma_G^{(l)} \leq \epsilon \Gamma_G^{(l-1)}$.

$$\begin{aligned}
\Gamma_G^{(l+1)} &= \sum_k c(k) \left(\Gamma_G^{(l)}\right)^k \\
&\leq \sum_k c(k) \left(\epsilon \Gamma_G^{(l-1)}\right)^k \\
&= \epsilon^{\lceil \frac{d^*}{2} \rceil} \sum_k c(k) \epsilon^{k - \lceil \frac{d^*}{2} \rceil} \left(\Gamma_G^{(l-1)}\right)^k \\
&\leq \epsilon^{\lceil \frac{d^*}{2} \rceil} \sum_k c(k) \left(\Gamma_G^{(l-1)}\right)^k \\
&= \epsilon^{\lceil \frac{d^*}{2} \rceil} \Gamma_G^{(l)} \\
&\leq \epsilon^{\lceil \frac{d^*}{2} \rceil + 1} \Gamma_G^{(l-1)}.
\end{aligned}$$

We used the fact that all of the $c(k)$ coefficients in the expansion are positive and due to the fault-tolerance of the logical gates, errors of order smaller than $\lceil d^*/2 \rceil$ are correctable and therefore $c(k) = 0 \; \forall \; k < \lceil d^*/2 \rceil$.

To complete the proof, we assume the induction hypothesis for level m and show for level $(m+1)$:

$$\begin{aligned}
\Gamma_G^{(m+1)} &= \sum_k c(k) \left(\Gamma_G^{(m)}\right)^k \\
&\leq \sum_k c(k) \left(\epsilon^{\sum_{r=0}^{m-l} \lceil \frac{d^*}{2} \rceil^r} \Gamma_G^{(l-1)}\right)^k \\
&\leq \epsilon^{\lceil \frac{d^*}{2} \rceil \sum_{r=0}^{m-l} \lceil \frac{d^*}{2} \rceil^r} \sum_k c(k) \left(\Gamma_G^{(l-1)}\right)^k \\
&= \epsilon^{\sum_{r=1}^{m+1-l} \lceil \frac{d^*}{2} \rceil^r} \sum_k c(k) \left(\Gamma_G^{(l-1)}\right)^k \\
&= \epsilon^{\sum_{r=1}^{m+1-l} \lceil \frac{d^*}{2} \rceil^r} \Gamma_G^{(l)} \\
&\leq \epsilon^{\sum_{r=0}^{m+1-l} \lceil \frac{d^*}{2} \rceil^r} \Gamma_G^{(l-1)},
\end{aligned}$$

thus completing the induction proof. □

The importance of this Lemma cannot be understated, it is key to the double-exponential suppression of the error rate as the concatenation level increases, which is the underlying

property required from the logical error rate in order to guarantee a ploy-logarithmic overhead in computational resources for fault-tolerant quantum computation. We explore the behaviour of these polynomials further for the 105-qubit code in the upcoming sections.

2.3.2 Concatenated 105-qubit thresholds

2.3.2.1 Error type analysis

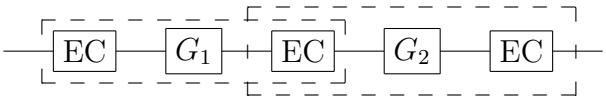
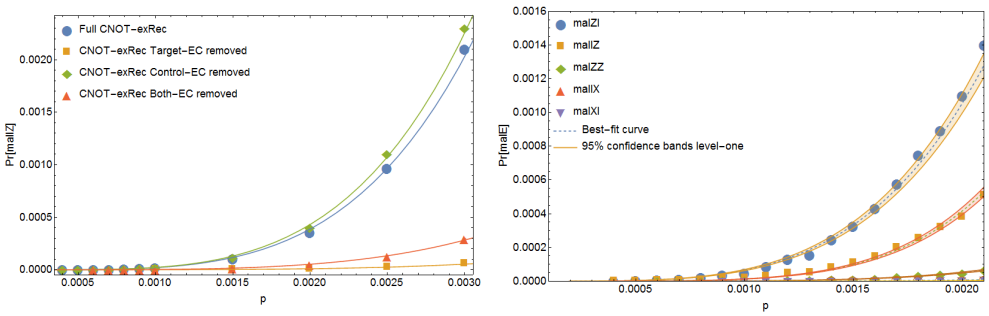


Figure 2.6: An example of shared EC's between two consecutive level-one exRecs

In computing the probability of obtaining a logical error E at the second level of concatenation for an error rate p and gate G ($\Pr[\text{mal}_E^{(2)}|G,p]$), each level-one exRecs in the level-two circuit was treated as a physical independent location with a redefined noise model given by the polynomials $\Gamma^{(1)}_{G,E}(p)$. For example, a level-one CNOT gate in a level-two simulation would be treated as a physical CNOT gate. A two-qubit Pauli error would be inserted with a probability upper bounded by the polynomials obtained in Eq. 2.66 with $l = 2$ instead of the probability arising from the depolarizing noise model. To be consistent with Eq. 2.66, the notation is chosen such that $\Gamma^{(l)}_G$ is the upper bounding polynomial at level-l for all error types E whereas $\Gamma^{(l)}_{G,E}$ is the upper bounding polynomial for the particular error type E.

As can be seen in Fig. 2.6, a level-two simulation will typically contain many level-one exRecs with overlapping ECs and so it is not entirely correct to treat them independently. If two level-one exRecs share ECs, then the rectangle that precedes the other one is replaced with a faulty gate only if it is still incorrect after the shared ECs have been removed. As was shown in Ref. [113], we must calculate probability of a logical fault for both complete and incomplete exRecs where one or more TECs have been removed and take the polynomial that bounds all cases, see Fig. 2.7a for an example. We would also like to point out that for single qubit gates, exRec's without a TEC always had a lower probability of obtaining a logical fault (for any error type) compared to the case where the TEC was kept. This can be understood from the fact that the TEC adds more locations and hence more ways for errors to be introduced at the output of the circuit.



(a) (b)

Figure 2.7: (a) Polynomials upper bounding the event mal_{IZ} for either the full level-one CNOT exRec or the level-one CNOT exRec with one or both TEC's removed. The polynomial upper bounding the event mal_{IZ} will upper bound all the curves in the above figure. (b) Polynomials upper bounding the level-one CNOT exRec for the different logical error types.

The polynomials in Fig. 2.7b upper bound the probability of obtaining a logical error at the first level of concatenation of the CNOT exRec. Each curve corresponds to a different error type (error types that are not displayed occur with a probability less than 10^{-7} for all sampled physical error rates). Note that the upper bounds on logical Z malignant events are significantly higher than their X counterpart. This is primarily due to the fact that the 15-qubit Reed-Muller code offers better protection against X errors.

The polynomials of Fig. 2.9a upper bound the probability of obtaining a logical X, Z or Y error for the level-one Hadamard exRec. The circuit encoding the logical H, see Fig. 2.8 on the 15-qubit codeblocks is very sensitive to input Z errors. Any Z error propagating through the target qubit of the CNOT gates will result in a logical X error on the 15-qubit codeblock. The latter is the main reason for a larger upper bound on the event mal_X compared to the event mal_Z even though the 15-qubit Reed-Muller code provides better protection against X errors.

Fig. 2.3 illustrates the T gate circuit construction for the 105-qubit concatenated code, where qubits that are inactive at a given time step would undergo memory noise. Notice that compared to the Hadamard circuit construction in Fig. 2.8, there are much fewer locations where errors can propagate leading to a logical fault on multiple codeblocks. Since the 105-qubit code is more efficient at correcting X errors, and X errors propagating through a physical T gate location transforms as $TXT^\dagger = X(I + iZ)/\sqrt{2}$ (leading to a Z error contribution), we expect the probability of obtaining a logical Z error at the output of the T gate circuit to be much higher than the probability of obtaining a logical X error. In

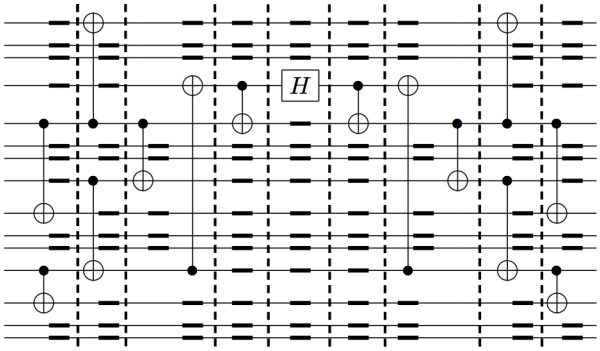


Figure 2.8: Logical Hadamard H circuit for $[[15, 1, 3]]$ Reed-Muller code. The bold dark lines represent resting qubits subject to storage errors. The dotted vertical lines are used to separate the time steps for which gates are applied in parallel. Logical H for the 105-qubit code is implemented fault-tolerantly by applying each non-fault-tolerant logical H gates in parallel.

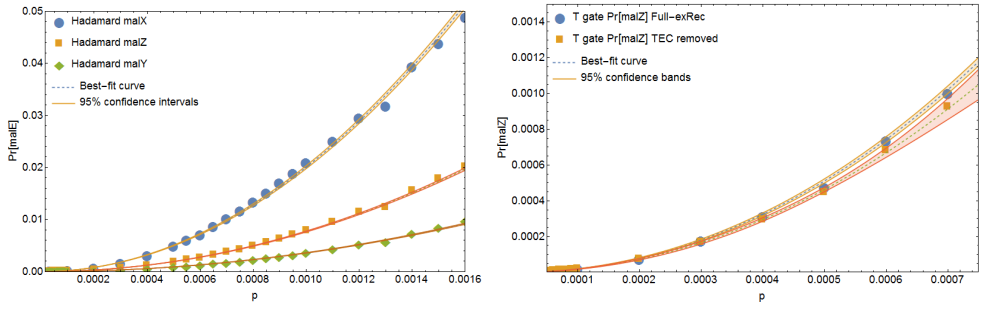
fact, our simulations showed that the level-one logical X error probability could be upper bounded by 10^{-7} for all considered error rates.

2.3.2.2 Threshold analysis

In order to lower bound the level-1 pseudo-threshold, the probability of all logical error types are summed for each of the logical gates and bounded as in Eq. 2.65. The resulting polynomials are compared to the input physical error rate and their crossing point determines the pseudo-threshold, see Fig. 2.11. The resulting values are presented in Table 2.1.

The pseudo-threshold values for a gate G given in Table 2.1 were obtained from the crossing point between the physical error rate p and the curves $p_G^{(1)}(p) = \sum_{E_i} \Pr[\text{mal}_{E_i}|G, p]$ for all possible logical Pauli errors E_i. The plots on the left column of Fig. 2.11 illustrates the crossing point for the logical CNOT, Hadamard and T gate. The CNOT gate has the largest pseudo-threshold value of $(2.11 \pm 0.02) \times 10^{-3}$ due to the double protection from the CSS 7-qubit Steane code and the 15-qubit Reed-Muller code. On the other hand, the Hadamard gate has the lowest pseudo-threshold value of $(4.47 \pm 0.29) \times 10^{-5}$ due to the sensitivity of the encoding circuit on the 15-qubit codeblocks to input Z errors.

Following Lemma 42, a lower bound for the asymptotic threshold value for a particular gate G is given by the the intersection between the polynomials upper bounding the probability of obtaining a logical error E at the first and second level of concatenation ($\Gamma_{G,E}^{(1)}$ and



(a) (b)

Figure 2.9: (a) Polynomials upper bounding the events mal_X, mal_Z and mal_Y for the level-one Hadamard exRec. Input Z errors are most likely to result in a logical X error on a 15-qubit codeblock which explains why the event mal_X is most likely to occur. (b) Polynomials upper bounding the event mal_Z for the level-one T gate exRec. Note that the logical error probabilities for the event mal_X and mal_Y are too small to be displayed.

$\Gamma^{(2)}_{G,E}$). Note that the error type E in the asymptotic threshold calculation is chosen such that the intersection between $\Gamma^{(1)}_{G,E}$ and $\Gamma^{(2)}_{G,E}$ occurs at the smallest physical error rate. For the CNOT gate (Fig. 2.11b) this is given by $E = ZI$, for the Hadamard gate (Fig. 2.11d) it is $E = X$ and for the T gate (Fig. 2.11f) it is $E = Z$.

It is important to observe that the CNOT pseudo-threshold is nearly two orders of magnitude larger than the Hadamard pseudo-threshold. Furthermore, all other operations in our circuits (resting qubits, measurement in the X and Z basis and state preparations) are upper bounded by level one polynomials that have larger pseudo-thresholds than CNOT. The dominant set of errors leading to logical faults in the level-1 Hadamard gate is a result of input errors from the LEC as well as failures in the CNOT gates within the 15-qubit Hadamard codeblocks. These components are composed of only memory, CNOT, X and Z basis state preparation and measurement locations. Since the level-1 logical error probability of these gates will be much smaller in the level-2 Hadamard exRec, detrimental faults will be much less likely to occur. Hence, there will be error rates p above the pseudo-threshold $p_{1,H}$ such that the level-2 error polynomials characterizing the logical error rate will be below the level-1 bounding polynomial,

$$\Gamma^{(2)}(p) \leq \Gamma^{(1)}(p), \ \forall \ p \leq p_{2,H}, \tag{2.69}$$

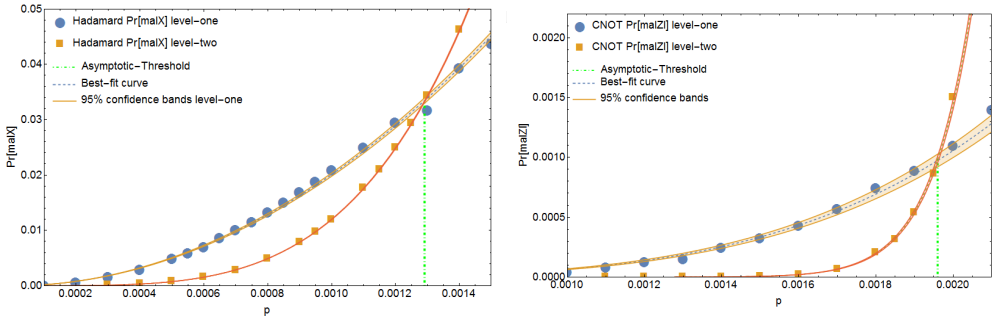
where $p_{2,H} > p_{1,H}$. The error rate $p_{2,H}$ is the threshold rate below which all level-2 logical gates have a lower error rate compared to the level-1 logical error rate. As shown in

	Pseudo-Threshold	Asymptotic threshold
CNOT gate	$(2.11 \pm 0.02) \times 10^{-3}$	$(1.95 \pm 0.01) \times 10^{-3}$
T gate	$(4.89 \pm 0.11) \times 10^{-4}$	$(1.58 \pm 0.02) \times 10^{-3}$
Hadamard gate	$(4.47 \pm 0.29) \times 10^{-5}$	$(1.28 \pm 0.02) \times 10^{-3}$
105-qubit	$\mathbf{(4.47 \pm 0.29) \times 10^{-5}}$	$\mathbf{(1.28 \pm 0.02) \times 10^{-3}}$
23-qubit Golay	$\mathbf{(1.73) \times 10^{-3}}$	$\mathbf{(1.32) \times 10^{-3}}$

Table 2.1: Lower bounds for the pseudo and asymptotic threshold results for the Hadamard, T gate and CNOT gates. The Hadamard asymptotic-threshold is larger than its pseudo-threshold resulting from the double protection of the CNOT gates as seen by the high CNOT pseudo-threshold. In bold, the overall thresholds for the 105-qubit and 23-qubit codes are compared.

Ref. [113] and argued in the previous section, the value $p_{2,H}$ serves as a lower bound for the asymptotic threshold p_{th}.

In previous studies of asymptotic thresholds for the Golay and 7-qubit CSS codes, the CNOT exRec provided a lower bound on the threshold value since it contained the largest amount of locations relative to all the other gates in the universal gate set [2, 113, 47]. Since the CNOT exRec is itself composed entirely of gates that are transversal, as the error rate approaches the pseudo-threshold value, certain malignant events (for example, the probability of getting a logical ZI error at the output of the CNOT circuit, as can be seen in Fig. 2.7b) become more likely to occur than the level-zero probabilities determined from the depolarizing noise model. Recall that the pseudo-threshold was conjectured to be an upper bound on the asymptotic threshold value. However, it is the CNOT locations that are the leading contributors to logical errors. Consequently, the pseudo-threshold of the CNOT gate, as opposed to the H and T gates, will be the limiting factor to the asymptotic threshold. As argued above, this will give rise to reduced logical error rates of the H and T gates at the second level of concatenation, and using Eq. 2.68, a lower bound for the asymptotic threshold p_{th} can be determined. The plots in Fig. 2.10 illustrate the level-1 and level-2 polynomials upper bounding the logical error rates at the first and second level for the Hadamard and CNOT gate circuits, see Fig. 2.11 for the corresponding T gate plots. As expected, the CNOT exRec contains a lower asymptotic threshold value given by $(1.95 \pm 0.01) \times 10^{-3}$. The Hadamard exRec limits the threshold value of the 105-qubit code to be $(1.28 \pm 0.02) \times 10^{-3}$. Interestingly, the level-two polynomials satisfy Eq. 2.68 for error rates nearly 30 times larger than their corresponding level-one polynomials. This is a distinctive feature of the 105-qubit concatenated scheme and clearly demonstrates the impact of having an exRec primarily composed of gates which are transversal in both



(a) (b)

Figure 2.10: Probability of logical error as function of physical error rate for the level-1 and level-2 logical (a) Hadamard and (b) CNOT. The crossing point of the fitted curve allows for the determination of a lower bound for the asymptotic threshold for each of the logical gates. The CNOT gate exhibits a much lower logical error rate than the Hadamard at the first level.

codes with much larger pseudo-threshold rates. The asymptotic threshold derived for the 105-qubit code compares favourably to the $[[23, 1, 7]]$ Golay code studied under the same depolarizing error model and metric for gate failures under malignant set counting [113]. This scheme does not require magic state distillation in order to achieve fault-tolerance and may lead to reduced overhead [55]. Determining the resource overhead remains an interesting open problem.

It should be noted that the shift in the crossing point for different concatenation levels in the logical H and T gate (Figs. 2.11d and 2.11f) may at first glance violate the assumption that the polynomial coefficients $c(k)$ from Lemma 42 are the same at all levels. However, one of the assumptions of the polynomials were that the logical error rate of all locations at the previous level have the same error rate, and thus contribute equally in a potential error chain. The fact that CNOT is in fact less noisy than other gates in the regime between the H (and T) pseudo-threshold and asymptotic CNOT threshold means that certain error chains are further suppressed and as such the logical error rate is lower than the worst case bound set by the polynomials. The CNOT crossing points (Fig. 2.11b) are uniform across all levels, indicating that the true logical error rate is very close to the worst-case bound.

An interesting feature can be observed from the plots on the right column of Fig. 2.11. Notice that the polynomial upper bounding the event mal_{ZI} at the third level of concatenation for the logical CNOT gate $\Gamma^{(3)}_{CNOT,ZI}(p)$ intersects $\Gamma^{(1)}_{CNOT,ZI}(p)$ at the asymptotic threshold value $(1.95 \pm 0.01) \times 10^{-3}$. The reason is that for higher error rates than

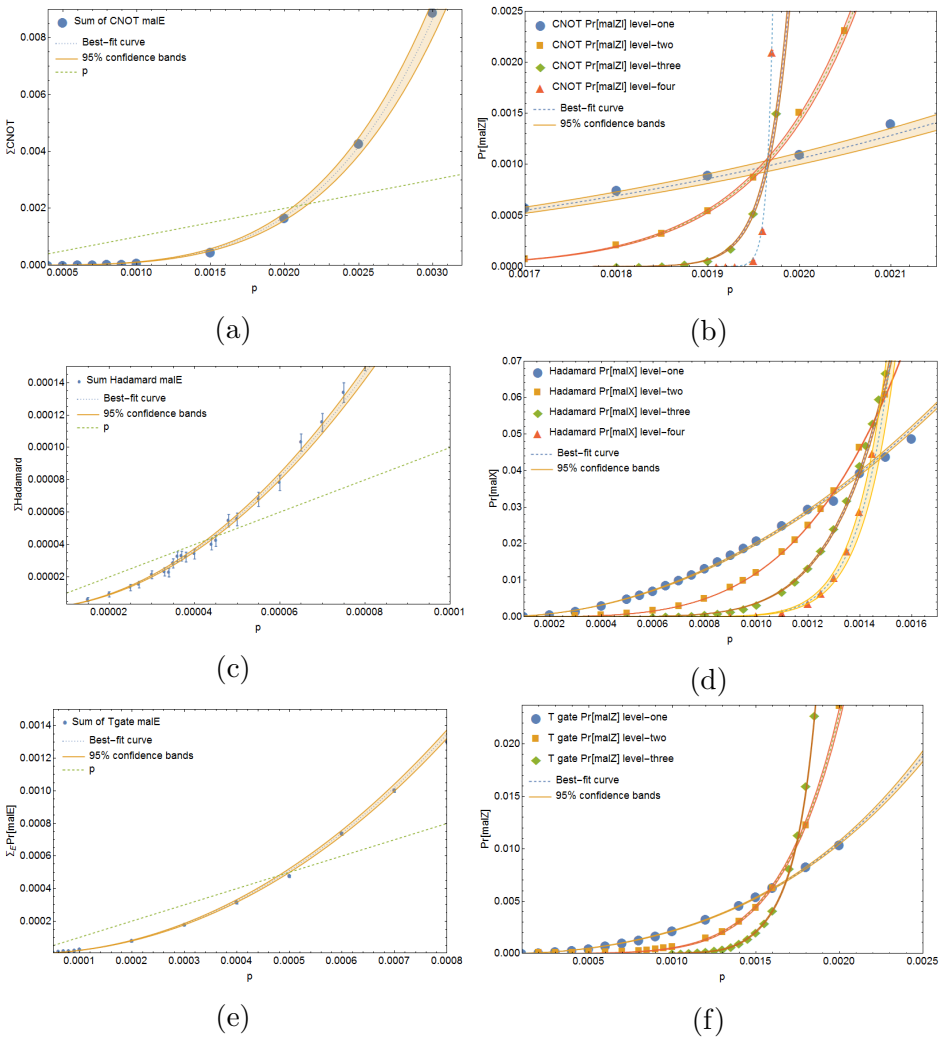


Figure 2.11: The plots on the left column illustrate the probability of logical error as function of physical error rate for logical (a) CNOT, (c) Hadamard and (e) T gate. The crossing point of the fitted curve allows for the determination of the level-1 pseudo-threshold for each of the logical gates. The CNOT pseudo-threshold is the largest among all three gates due to the double protection of the 7-qubit and 15-qubit code. The plots on the right column illustrate the polynomials upper bounding the probability of obtaining a logical error E for the first, second and third level of concatenation. The crossing point between the level-one and level-two polynomials determine the asymptotic threshold for the gate under consideration. For the logical CNOT gate (b), it is the event mal_{ZI} which limits the threshold value. For the logical gate H (d), mal_X limits the threshold value. Lastly, for the logical T gate (f), mal_Z limits the threshold value.

the asymptotic threshold value, the level-two CNOT exRecs (in the level three simulation) are more likely to fail than the level-one CNOT exRecs (in the level two simulation). Consequently, there is a higher probability of obtaining a logical fault at the output of the CNOT exRec. However, for the logical H and T gate exRecs, $\Gamma^{(3)}_{G,E}(p)$ intersects $\Gamma^{(1)}_{G,E}(p)$ at an error rate which is larger than the asymptotic threshold value for these particular gates ($(1.28 \pm 0.02) \times 10^{-3}$ for H and $(1.58 \pm 0.01) \times 10^{-3}$ for T). Consider the logical Hadamard gate (the following argument applies equally well to the T gate). For error rates p that are between the H and CNOT asymptotic threshold values, $(1.28 \pm 0.02) \times 10^{-3} \leq p \leq (1.95 \pm 0.01) \times 10^{-3}$, the level-two Hadamard exRecs in the level-three simulation will be more likely to fail than at the previous level of concatenation. However, this will be compensated by all of the level-two CNOT exRecs in the level-three simulation which will be less likely to fail than at the previous level (since p is below the CNOT asymptotic threshold value). Above the error rate where $\Gamma^{(3)}_{H,X}(p)$ intersects $\Gamma^{(1)}_{H,X}(p)$ ($p = 1.44 \times 10^{-3}$), the level-two Hadamard exRecs will be noisy enough such that the probability of obtaining a logical X error will be larger than at the previous level. Therefore, by considering the crossing points of the logical error rates for higher concatenation levels, a better lower-bound for the asymptotic threshold can be established. However, in order to fairly compare the performance of the concatenated scheme with the Golay code [113], we emphasized the lower bound obtained from the crossing point of the first and second concatenation levels.

2.3.3 Summary & outlook

In this work, we established the first rigorous lower bound on the asymptotic threshold for the concatenated 105-qubit code. We show that the pseudo-threshold value of $(4.47 \pm 0.29) \times 10^{-5}$ arising from the H gate is significantly improved at higher levels of concatenation yielding a lower bound on the asymptotic threshold value of $(1.28 \pm 0.02) \times 10^{-3}$. The increase in asymptotic threshold is primarily due to the relatively high threshold of the logical CNOT gate. We believe that this non-traditional behaviour of logical error probabilities at higher concatenation levels is an interesting property of the studied scheme and points to an interesting direction for future error correction research. Due to the high concentration of CNOT gates for the purposes of error detection, we believe that tailoring codes to correct for logical errors in encoded CNOT gates at the expense of perhaps noisy single qubit gates would allow for higher asymptotic thresholds for concatenated codes. Furthermore, an interesting direction for future research is to establish similar thresholds for other types of concatenated codes, such as the 49-qubit code outlined at the conclusion of the previous section. Additionally, studying the performance of the concatenated

scheme under more complex noise models would be of theoretical and practical interest, as in Refs. [66, 64, 65]. Moreover, determining the resource overhead of these schemes in comparison to those based on magic state distillation is subject to an active study [37] and minimizing such overheads may have significant impact on the development of quantum computing architectures.

2.4 Gate restrictions for transversal operations in stabilizer codes

This section is based on the work that is presented in Ref. [7], copyrighted by Rinton Press. Both authors contributed to the development of the research direction, ideas and writing of the manuscript, and editing of the manuscript.

2.4.1 Overview

As seen in the previous sections, the role of transversal gates is an important one for the implementation of logical gates in a fault-tolerant manner. We have already seen examples of codes that can implement gates from the Clifford hierarchy transversally, as well as the T gate transversally for the 15-qubit code. While such gates are sufficient for universal quantum computation (albeit of course with the caveat that no single gate can have a universal set of transversal gates), a natural question to ask is whether other codes with different classes of transversal gates exist. The Reed-Muller code family points to a class of transversal operations for fractional roots of unity of higher powers of 2, that is of the form $e^{i\pi Z/2^c}$ [96], however other classes of operations are not known. Moreover, recent results in the area of quantum gate decomposition have focused on expressing an arbitrary single-qubit quantum gate as a sequence of Hadamard (H) and V gates, where $V = \mathrm{diag}(1+2i, 1-2i)/\sqrt{5}$ [21]. Therefore, the discovery of quantum error correcting codes that allow for the application of V in a transversal manner could potentially lead to adaptation of the above mentioned techniques for universal fault-tolerant gate application without state distillation for these proposed gate decompositions.

Recently, a parallel work by Pastawski and Yoshida [115] showed many exciting results pertaining to fault-tolerant operations in topological stabilizer codes. They also proved that families of stabilizer codes with a finite loss threshold must have transversal gates in the Clifford hierarchy. Furthermore, they show that higher loss thresholds impose greater

restrictions on the level in the Clifford hierarchy at which transversal gates can be implemented. The result presented in this section does not need a finite loss threshold or a family of codes to be applicable. Additionally, this result applies to transversal gates between two like codes; however, this result only applies to qubits and restricts the transversal gates to being in the Clifford hierarchy (although it does not specify the level). A parallel work has also shown that transversal logical gates for qubit stabilizer codes must belong to the Clifford hierarchy, and that the level of the Clifford hierarchy which can be implemented is bounded as a function of the number of physical qubits and code distance [20].

The main result of this work is that for quantum qubit stabilizer codes, the *only* diagonal gates that can be implemented transversally using physical diagonal gates are those whose entries along the diagonal are of the form $e^{i\pi c/2^k}$, for some power of k depending on the choice of code. This result holds both for single and two-qubit gates, and moreover we show that all such gates must be contained within the Clifford hierarchy. Moreover, as Zeng, Cross, and Chuang showed [142], any transversal non-trivial single-qubit logical gate for a qubit stabilizer code must result from the application of diagonal gates along with local Clifford operations and potential swapping of qubits. Therefore, our result classifies all transversal single-qubit logical gate operations up to local Clifford equivalences and relabelling of qubits. Additionally, our result classifies all transversal diagonal single-qubit logical gates that can map one stabilizer code to another stabilizer code. It is worth noting that the Reed-Muller family of quantum codes provides a means to implement any of these diagonal transversal gates, where changing to higher order in the code family allows for the implementation of diagonal logical gates with finer angles, all of which are in the Clifford hierarchy and of the form $e^{i\pi c/2^k}$. Finally, we show that the two-qubit transversal gates that can be applied for a given quantum stabilizer code must belong to the same level of the Clifford hierarchy as the single qubit transversal logical gates that can be applied for such a code. As such, there is no increased computational logical power by considering two-qubit transversal gates and we conjecture this result to generalize to multi-qubit gates.

2.4.2 Single qubit transversal Z rotations

In this work we aim to classify the set of logical transversal gates for a stabilizer code, that is logical gates that are composed of individual unitary gates on each of the underlying qubits of the code, $U = \bigotimes_{j=1}^{n} U_j$. We begin by noting an important result from Zeng *et al.* [142] where it was shown that unitary, single-qubit logical transversal operators in qubit stabilizer codes must be of the form:

$$U = L\left(\bigotimes_{j=1}^{n} \mathrm{diag}(1, e^{i\pi\theta_j})\right) R^\dagger P_\pi. \tag{2.70}$$

Here $L, R^\dagger$ are transversal Clifford operations and P_π is a coordinate permutation (a set of SWAP gates). Notice that if an $[[n,k,d]]$ stabilizer code exists which implements U transversally, then up to local Clifford equivalences, an $[[n,k,d]]$ stabilizer code exists which implements $U = \bigotimes_{j=1}^{n} \mathrm{diag}(1, e^{i\pi\theta_j})$ transversally. In this work we look at the restrictions on these diagonal, transversal gates.

We begin by focusing on $Z(\theta)$ rotations, that is rotations about the Z-axis by some angle θ. For qubits, this rotation is given by a diagonal matrix

$$A = \begin{bmatrix} e^{i\pi\theta_1} & 0 \\ 0 & e^{i\pi\theta_2} \end{bmatrix} = e^{i\pi\theta_1} \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi(\theta_2-\theta_1)} \end{bmatrix}. \tag{2.71}$$

Up to a global phase, we therefore need only consider rotations of the form

$$A = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi\theta} \end{bmatrix} \equiv Z(\theta), \tag{2.72}$$

where we are using the above equation as the definition of a single-qubit $Z(\theta)$ rotation of angle $\pi\theta$ (we shall assume for the remainder of this work that the angular rotations are rational multiples of π, as discussed in detail below).

In this work, we study constraints on transversal implementations of logical $Z(\theta)$ rotations. A transversal Z rotation is defined as

$$Z_T(\theta) := Z(\theta_1) \otimes Z(\theta_2) \otimes ... \otimes Z(\theta_n). \tag{2.73}$$

Before considering the most general form of transversal gate outlined above, we first focus on the case when all physical qubits undergo the same rotation θ, that is, we require that the logical implementation be *strongly transversal*, $Z_L(\theta') = Z(\theta)^{\otimes n}$, with n being the number of physical qubits. While each single-qubit rotation is a rotation by the same angle, we do not require that the logical Z applies the same rotation to the logical qubit.

2.4.2.1 CSS codes

We begin by focusing on the case of CSS codes $\mathcal{S} = \langle G_1^X, \ldots, G_{|G_X|}^X, G_1^Z, \ldots, G_{|G_Z|}^Z \rangle$, where $|G_X|$ and $|G_Z|$ refer to the number of X and Z stabilizers. Additionally, a CSS code

defines logical X and Z operators composed only of X and Z operators, again respectively. Recall the definition of the codestates presented in Lemma 31. The codestates can then be expanded in terms of each of the X generators:

$$|0_L\rangle = |g_0\rangle + \sum_{i_1} |g_{i_1}\rangle + \sum_{i_1<i_2} |g_{i_1} \oplus g_{i_2}\rangle + \cdots + \sum_{i_1<\cdots<i_{|G_X|-1}} |g_{i_1} \oplus \cdots \oplus g_{i_{|G_X|-1}}\rangle + |g_1 \oplus \ldots \oplus g_{|G_X|}\rangle, \tag{2.74}$$

$$|1_L\rangle = |g_L\rangle + \sum_{i_1} |g_L \oplus g_{i_1}\rangle + \sum_{i_1<i_2} |g_L \oplus g_{i_1} \oplus g_{i_2}\rangle + \ldots + |g_L \oplus g_1 \oplus \ldots \oplus g_{|G_X|}\rangle, \tag{2.75}$$

where g_i is a n-bit binary string representing the location of the X operators in the generator G_i^X, g_0 is the all-0 vector and g_L is a binary vector representing X_L. In order to classify the set of transversal Z-axis rotations that can implement a logical operation we derive a set of conditions restricting the rotation angles due to constrains arising from the form of the binary vectors g_i.

Theorem 43. *Given a $[[n, l, d]]$ error-detecting CSS code Q (distance $d \geq 2$). A strongly transversal rotation, that is a gate of the form $Z(\theta)^{\otimes n}$, that implements a non-identity logical gate for Q is restricted to be composed of individual rotations of the form $Z(a/2^k)$, where $a, k \in \mathbb{N}$.*

Proof. Consider the action of the strongly transversal rotation $Z(\theta)^{\otimes n}$ on the codestates given in Equations 2.74–2.75. Given that $Z(\theta)^{\otimes n}$ will preserve computational basis states, other than introducing a complex phase, then in order for the transversal rotation to implement a logical operation each term in the expansion of $|0_L\rangle$ must obtain an identical phase (in this case a trivial phase of 1 since $Z(\theta)^{\otimes n}|g_0\rangle = |g_0\rangle$). Similarly, all states in the expansion of $|1_L\rangle$ must obtain an identical phase, a non-trivial complex phase in order for

the logical operator to be a non-identity gate, as follows:

$$\begin{aligned}
Z(\theta)^{\otimes n}|0_L\rangle &= |g_0\rangle + Z(\theta)^{\otimes n}\left(\sum_{i_1}|g_{i_1}\rangle + \sum_{i_1<i_2}|g_{i_1}\oplus g_{i_2}\rangle + ... + |g_1\oplus ...\oplus g_{|G_X|}\rangle\right)\\
&= |g_0\rangle + \sum_{i_1} e^{i\pi\theta|g_{i_1}|}|g_{i_1}\rangle + \sum_{i_1<i_2} e^{i\pi\theta|g_{i_1}\oplus g_{i_2}|}|g_{i_1}\oplus g_{i_2}\rangle\\
&\quad + ... + e^{i\pi\theta|g_1\oplus ...\oplus g_{|G_X|}|}|g_1\oplus ...\oplus g_{|G_X|}\rangle\\
&= |g_0\rangle + \sum_{i_1}|g_{i_1}\rangle + \sum_{i_1<i_2}|g_{i_1}\oplus g_{i_2}\rangle + ... + |g_1\oplus ...\oplus g_{|G_X|}\rangle,\\
Z(\theta)^{\otimes n}|1_L\rangle &= e^{i\pi\theta|g_L|}\left(|g_L\rangle + \sum_i |g_L\oplus g_i\rangle + ... + |g_L\oplus g_1\oplus ...\oplus g_{|G_X|}\rangle\right),
\end{aligned}$$

where $|x|$ denotes the weight of the binary string x. The constraint that each of the states in the expansion of $|0_L\rangle$ must obtain the same phase, and similarly for $|1_L\rangle$, can be summarized by the following set of modular algebraic constraints:

$$\begin{array}{rl}
\theta|g_{i_1}| = 0 \bmod 2 & \theta|g_L| = a \bmod 2\\
\theta|g_{i_1}\oplus g_{i_2}| = 0 \bmod 2 & \theta|g_L\oplus g_{i_1}| = a \bmod 2\\
\vdots & \vdots\\
\theta|g_1\oplus ...\oplus g_{|G_X|}| = 0 \bmod 2 & \theta|g_L\oplus g_1\oplus ...\oplus g_{|G_X|}| = a \bmod 2,\\
 & \forall\ 0<i_1<i_2<...<i_{|G_X|-1}\le |G_X|,
\end{array}$$

where $\theta,\ a\in\mathbb{R}$ and $|.|\in\mathbb{N}$. The constraints on the left correspond to those for the $|0_L\rangle$ state and those on the right the $|1_L\rangle$ state.

We begin by making some observations on the above equations to rule out certain values of θ.

1. First, notice that if θ is irrational these equations can never be satisfied since $n\theta = p\equiv 0 \bmod 2 \implies \theta = \frac{2t}{n}\in\mathbb{Q}$. We can therefore restrict our attention to rational angles ($\theta = \frac{p}{q}\in\mathbb{Q}$). Without loss of generality, we can assume this fraction is irreducible and in the range $(0,2]$. Moreover, we must have that the weight of the binary strings is a multiple of q, that is $\frac{p}{q}|\cdot| = 0 \mod q \implies |\cdot| = 0 \mod q$.

2. We can express these constraints as conditions on overlap similarly to Bravyi and Haah [31] by noting that

$$|g_1\oplus ...\oplus g_n| = \sum_{i=1}^{n}|g_i| - 2\sum_{i<j}|g_i\wedge g_j| + ... + (-2)^{n-1}|g_1\wedge ...\wedge g_n|, \tag{2.76}$$

where $\wedge$ is the bitwise AND.

With these observations, we can express the constraints as:

$$\begin{aligned}
|g_{i_1}| &= 0 \bmod q \\
|g_{i_1}| + |g_{i_2}| - 2|g_{i_1} \wedge g_{i_2}| &= 0 \bmod q \\
&\vdots \\
\sum_{i_1}^{n} |g_{i_1}| - 2 \sum_{i_1 < i_2} |g_{i_1} \wedge g_{i_2}| + ... + (-2)^{|G_X|-1} |g_1 \wedge ... \wedge g_{|G_X|}| &= 0 \bmod q
\end{aligned}$$

$$\begin{aligned}
|g_L| &= b \bmod q \\
|g_L| + |g_{i_1}| - 2|g_L \wedge g_{i_1}| &= b \bmod q \\
&\vdots \\
\forall\ 0 < i_1 < i_2 < ... < i_{|G_X|-1} &\leq |G_X|
\end{aligned}$$

We can see that these equations are not independent since the requirement that $|g_i| = 0 \bmod q$, implies

$$|g_{i_1}| + |g_{i_2}| - 2|g_{i_1} \wedge g_{i_2}| = 0 \bmod q \implies 2|g_{i_1} \wedge g_{i_2}| = 0 \bmod q. \tag{2.77}$$

Using this, we can express the above constraints as overlap conditions

$$\begin{aligned}
|g_{i_1}| &= 0 \bmod q, \\
2|g_{i_1} \wedge g_{i_2}| &= 0 \bmod q \\
4|g_{i_1} \wedge g_{i_2} \wedge g_{i_3}| &= 0 \bmod q \\
&\vdots \\
(2)^{|G_X|-1} |g_1 \wedge ... \wedge g_{|G_X|}| &= 0 \bmod q \\
|g_L| &\neq 0 \bmod q \\
2|g_{i_1} \wedge g_L| &= 0 \bmod q \\
4|g_{i_1} \wedge g_{i_2} \wedge g_L| &= 0 \bmod q \\
&\vdots \\
(2)^{|G_X|} |g_1 \wedge ... \wedge g_{|G_X|} \wedge g_L| &= 0 \bmod q \\
\forall\ 0 < i_1 < i_2 < ... < i_{|G_X|-1} &\leq |G_X|,
\end{aligned}$$

We have also dropped the minus sign since it has no effect. For the logical operator to be nontrivial, we have assumed that $b \neq 0$. Notice that the $0 \bmod q$ conditions are independent constraints.

Observe that if q has only even prime factors (*i.e.* $q = 2^t$ for some integer t) then all higher-order overlap conditions will, at some point, become trivial. For example, if $q = 2^k$ overlap conditions will be trivial for any $k+1$ or more rows and Reed-Muller codes exist which have any $Z(1/2^k)$ gate transversally. In fact, since the transversal gates form a group, Reed-Muller codes exist which have any $Z(c/2^k)$ (where c is an integer) gate transversally. Therefore the existence of transversal gates is already solved in the positive for that case.

In what follows we will assume that q has at least one odd prime factor and that $|g_L| \neq 0 \bmod q_o$ for at least one such q_o (we will choose this q_o). If $|g_L| = 0 \bmod q_o$ for all odd prime factors, again assuming $\theta = p/q$ then $Z(\theta)^{\otimes n}|1_L\rangle = e^{i\pi|g_L|p/q}|1_L\rangle = e^{i\pi c/2^k}|1_L\rangle$ for some positive integer k, where $c \equiv |g_L| \bmod 2^k$. In this case, the odd prime factors add nothing, and we could apply the same logical operator using $Z(1/2^k)^{\otimes n}$ instead of $Z(p/q)^{\otimes n}$, which reduces to the case we considered above. Therefore, we assume $|g_L| \neq 0 \bmod q_o$ for at least one such q_o and note that in this case all overlap conditions are nontrivial. We can write $q = q_o \cdot q_{P/o}$ where $q_{P/o}$ is the product of the other prime factors of q. Since $|g| = 0, 1 \bmod q \implies |g| = 0, 1 \bmod q_o$, we can write a weaker set of overlap conditions as

$$\begin{aligned}
|g_{i_1}| &= 0 \bmod q_o, \\
|g_{i_1} \wedge g_{i_2}| &= 0 \bmod q_o \\
|g_{i_1} \wedge g_{i_2} \wedge g_{i_3}| &= 0 \bmod q_o \\
&\vdots \\
|g_1 \wedge ... \wedge g_{|G_X|}| &= 0 \bmod q_o \\
|g_L| &\neq 0 \bmod q_o \\
|g_{i_1} \wedge g_L| &= 0 \bmod q_o \\
|g_{i_1} \wedge g_{i_2} \wedge g_L| &= 0 \bmod q_o \\
&\vdots \\
|g_1 \wedge ... \wedge g_{|G_X|} \wedge g_L| &= 0 \bmod q_o \\
\forall\ 0 < i_1 < i_2 < ... < i_{|G_X|-1} &\leq |G_X|.
\end{aligned}$$

In what follows, we will attempt to find the smallest binary matrix (in terms of number of rows) which satisfies all of the weaker overlap conditions.

Let S_X be the parity check matrix for the X stabilizers, that is S_X is a binary matrix where each row is given by independent binary strings g_i, $i \in \{1, ..., |G_X|\}$. In the statement

of the Theorem we are assuming the code is an error-detecting code with distance $d \geq 2$, therefore each column of S_X must have at least one non-zero entry as this is a necessary and sufficient condition for Z error detection. We will consider the case of a single logical operator and show that no nontrivial matrix S_X with at least one X logical operator exists, such that all rows satisfy the overlap conditions derived above [3].

Let us now try to find the smallest number of rows in S_X such that the overlap conditions are satisfied. We begin by defining the variable $w_{\vec{\mu}}$, where $\vec{\mu} = (\mu_1\ \mu_2 \ldots\ \mu_{|G_X|})$ is a $|G_X|$ bit binary vector, as follows:

$$w_{\vec{\mu}} = \left|\left(\wedge_{(\mu_i=1)} g_i\right) \wedge \left(\wedge_{(\mu_i=0)} \neg g_i\right) \wedge g_L\right|, \tag{2.78}$$

that is $w_{\vec{\mu}}$ is the weight of the bitwise AND of g_L along with all binary vectors g_i such that $\mu_i = 1$ and $\neg g_j$ (negation of g_j) such that $\mu_j = 0$. For example if $|G_X| = 2$, that is S_X is composed of just two rows, then $w_{(10)} = |g_1 \wedge (\neg g_2) \wedge g_L|$, $w_{(01)} = |(\neg g_1) \wedge g_2 \wedge g_L|$, and $w_{(11)} = |g_1 \wedge g_2 \wedge g_L|$. While this definition may at first seem non-intuitive, we choose such a definition in order to consider quantities from the constraints on g_i above. For example, again in the case of $|G_X| = 2$, the constraint $|g_1 \wedge g_L| = 0 \mod q_o$ can be expressed as $w_{(11)} + w_{(10)} = 0 \bmod q_o$ by the straightforward binary string relationship $|g_1 \wedge g_L| = |g_1 \wedge g_2 \wedge g_L| + |g_1 \wedge (\neg g_2) \wedge g_L|$.

Using the definition of $w_{\vec{\mu}}$ we can arrive at a contradiction to the set of overlap conditions. First consider the case of $w_{\vec{\mu}}$ where $\vec{\mu} = (11\ \ldots 1)$ is the all-1 vector. Therefore, by the above constraints,

$$w_{(11\ldots1)} = |g_1 \wedge \ldots \wedge g_{|G_X|} \wedge g_L| = 0 \bmod q_o.$$

Now consider the overlap condition consisting of $|G_X| - 1$ binary vectors g_i, that is $|g_{i_1} \wedge \ldots \wedge g_{i_{|G_X|-1}} \wedge g_L| = 0 \bmod q_o$. Without loss of generality, consider the case of the first $|G_X| - 1$ vectors g_i. Then the following identity holds:

$$|g_1 \wedge \ldots \wedge g_{|G_X|-1} \wedge g_L| = w_{(11\ldots11)} + w_{(11\ldots10)} = 0 \bmod q_o,$$

and therefore $w_{(11\ldots10)} = 0$, where equality for the remainder of the proof is mod q_o. Thus, without loss of generality $w_{\vec{\mu}} = 0$ for all vectors $\vec{\mu}$ of weight $|G_X| - 1$. We can now recursively prove the same statement for any weight $\vec{\mu}$. Suppose $w_{\vec{\mu}} = 0$ for all $|\vec{\mu}| > k$ and consider without loss of generality the bitwise AND of the first k vectors g_i and g_L,

$$|g_1 \wedge \ldots \wedge g_k \wedge g_L| = \sum_{i=k+1}^{|G_X|} \sum_{\mu_i=0}^{1} w_{(1\ldots1\mu_{k+1}\ldots\mu_{|G_X|})} = 0 \bmod q_o.$$

[3] This part of our proof uses techniques developed in [31].

However, since all of the terms in the above sum are equal to 0 by the induction hypothesis except for the term where $\mu_i = 0 \; \forall \; i > k$, we arrive at the conclusion that $w_{(\vec{1}_k \vec{0}_{|G_X|-k})=0}$, that is $w_{\vec{\mu}} = 0$ when the only the first k entries of $\vec{\mu}$ are non-zero. Since the choice of which k entries were non-zero was arbitrary, we arrive at the conclusion that $w_{\vec{\mu}} = 0 \; \forall |\vec{\mu}| \geq k$. We have therefore proven the induction statement that $w_{\vec{\mu}} = 0 \; \forall \; \vec{\mu}$. Finally, in order to arrive at the contradiction to the satisfiability of the vector constrains, note that $|g_L| = \sum_{\vec{\mu}} w_{\vec{\mu}} = 0$, which contradicts the condition for a non-trivial gate $|g_L| \neq 0 \mod q_o$ Therefore no such q_o exists.

Finally, it is worth noting that the set of constraints are only dependant on the X type stabilizer generators and their parity check matrix S_X, and are not dependant on the number of logical operators. Therefore, additional logical qubits (and thus constraints for the different binary strings $g_{X_{L,i}}$ of the different logical $X_{L,i}$ operators would only further constrain the problem and reduce the set of angles. As such, this result holds for multiple logical qubits as well. □

Remark 1. We made the assumption that the logical X operator was composed of a set of individual X operators on a collection of qubits characterized by the bit string g_L, where $g_L(i) = 1$ if X_L performs the operation X at qubit i. However in theory, X_L could also be comprised of Z (or Y) operations as well. A particular Z (or Y) gate could introduce a phase on some of the state vectors in the expansion of the logical $|1_L\rangle$, yet these phases must be preserved by the action of $Z(\theta)^{\otimes n}$. Since these diagonal rotations will not change the form of the computational basis state, they will only introduce a phase. In that manner, the presence of Z (or Y) operations in the logical X_L gate will not change the set of algebraic conditions for the physical rotations $Z(\theta)$.

Remark 2. We made the assumption that the individual rotation on the physical qubits, $Z(\theta)$, were of the form $\text{diag}(1, e^{i\theta})$, however in full generality the diagonal gates can be of the form $\text{diag}(e^{i\varphi}, e^{i\theta})$. The resulting conditions on the transformation of the logical states $|0_L\rangle$ and $|1_L\rangle$ will have the form:

$$\begin{aligned}
Z(\theta)^{\otimes n}|0_L\rangle &= Z(\theta)^{\otimes n}\left(|g_0\rangle + \sum_{i_1}|g_{i_1}\rangle + \sum_{i_1<i_2}|g_{i_1}\oplus g_{i_2}\rangle + ... + |g_{i_1}\oplus ... \oplus g_{|G_X|}\rangle\right) \\
&= e^{i\varphi n}|g_0\rangle + \sum_{i_1} e^{i\theta|g_{i_1}|+i\varphi(n-|g_i|)}|g_{i_1}\rangle + \sum_{i_1<i_2} e^{i\theta|g_{i_1}\oplus g_{i_2}|+i\varphi(n-|g_{i_1}\oplus g_{i_2}|)}|g_{i_1}\oplus g_{i_2}\rangle \\
&\qquad + \cdots + e^{i\theta|g_{i_1}\oplus ...\oplus g_{|G_X|}|+i\varphi(n-|g_{i_1}\oplus ...\oplus g_{|G_X|}|)}|g_{i_1}\oplus ... \oplus g_{|G_X|}\rangle \\
&= e^{i\varphi n}\left(|g_0\rangle + \sum_{i_1}|g_{i_1}\rangle + \sum_{i_1<i_2}|g_{i_1}\oplus g_{i_2}\rangle + ... + |g_{i_1}\oplus ... \oplus g_{|G_X|}\rangle\right)
\end{aligned}$$

and

$$Z(\theta)^{\otimes n}|1_L\rangle = e^{i\varphi n + i(\theta-\varphi)|g_L|}\left(|g_L\rangle + \sum_{i_1}|g_L \oplus g_{i_1}\rangle + ... + |g_L \oplus g_{i_1} \oplus ... \oplus g_{|G_X|}\rangle\right).$$

The constraints can then be shown to have the form:

$$\begin{aligned}
(\theta-\varphi)|g_{i_1}| &= 0 \bmod 2 \\
2(\theta-\varphi)|g_{i_1} \wedge g_{i_2}| &= 0 \bmod 2 \\
&\vdots \\
2^{|G_X|-1}(\theta-\varphi)|g_{i_1} \wedge \ldots \wedge g_{|G_X|}| &= 0 \bmod 2 \\
\\
(\theta-\varphi)|g_L| &\neq 0 \bmod 2 \\
2(\theta-\varphi)|g_L \wedge g_{i_1}| &= 0 \bmod 2 \\
&\vdots \\
2^{|G_X|}(\theta-\varphi)|g_L \wedge g_{i_1} \wedge ... \wedge g_{|G_X|}| &= 0 \bmod 2 \\
\forall\ 0 < i_1 < i_2 < ... < i_{|G_X|-1} \leq |G_X|,
\end{aligned}$$

which are the same constrains on the difference of the phases $(\theta-\varphi)$ as the case when $\varphi = 0$. Therefore, an arbitrary global phase can be introduced on the individual rotations of the form $\mathrm{diag}(1, e^{i\theta})$ which are allowed in the CSS construction.

It is worth noting that the restriction on the set of rotations that can be applied to the individual qubits of a CSS code will impose a restriction on the set of logical rotations that can be applied. This shows a strong connection to the Clifford hierarchy. The Clifford hierarchy is defined recursively, where the first level of the hierarchy on n qubits is defined as the Pauli operators on n qubits, denoted $\mathcal{C}_n^{(1)} = \mathcal{P}_n$. Higher levels ($k \geq 2$) of the Clifford hierarchy are then defined as follows:

$$\mathcal{C}_n^{(k)} = \{U \in U(2^n) \mid UPU^\dagger \in \mathcal{C}_n^{(k-1)}\ \forall P \in \mathcal{P}_n\},$$

that is, a unitary U in the k-th level of the Clifford hierarchy maps by conjugation the Pauli operators on n qubits to an element in the $(k-1)$-th level of the Clifford hierarchy. Namely, the second level of the Clifford hierarchy is the Clifford operators, mapping Pauli operators to Pauli operators. It is worth noting that each level of the Clifford hierarchy contains all lower levels of the Clifford hierarchy, that is $\mathcal{C}_n^{(p)} \subsetneq \mathcal{C}_n^{(q)}$, if $p < q$.

Proposition 1. *Let $A = Z(\theta)$ be a diagonal single-qubit operator. If $\theta = c/2^k$, for any integer $k \geq 0$ where θ is in its most reduced form, then $A \in \mathcal{C}_1^{(k+1)}$. Otherwise, A is not in the Clifford hierarchy, that is $A \notin C_1^{(k)}$ for all k.*

Proof. Consider the action of conjugation of the operator $A = Z(\theta)$ on the single qubit Pauli matrix X, the action on Pauli Z is trivial due to the commutation of diagonal matrices. Consider the recursive construction of the matrices A_p defined as: $A_p = A_{p-1} X A_{p-1}^\dagger$, where $A_0 = A$. Notice the following:

$$
\begin{aligned}
A_1 = A_0 X A_0^\dagger &= \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi\theta} \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & e^{-i\pi\theta} \end{pmatrix} = \begin{pmatrix} 0 & e^{-i\pi\theta} \\ e^{i\pi\theta} & 0 \end{pmatrix}, \\
A_2 = A_1 X A_1^\dagger &= \begin{pmatrix} 0 & e^{-i\pi\theta} \\ e^{i\pi\theta} & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & e^{-i\pi\theta} \\ e^{i\pi\theta} & 0 \end{pmatrix} = \begin{pmatrix} 0 & e^{-2i\pi\theta} \\ e^{2i\pi\theta} & 0 \end{pmatrix}, \\
&\vdots \\
A_p = A_{p-1} X A_{p-1}^\dagger &= \begin{pmatrix} 0 & e^{-2^p i\pi\theta} \\ e^{2^p i\pi\theta} & 0 \end{pmatrix}.
\end{aligned}
$$

If $A \in \mathcal{C}_1^{(k+1)}$ for some $k \geq 0$, then by definition $A_1 \in \mathcal{C}_1^{(k)}$, $A_2 \in \mathcal{C}_1^{(k-1)}$, $\ldots$, $A_k \in \mathcal{C}^{(1)} = \mathcal{P}_1$. However, notice by the form of A_k that $A_k = X \Leftrightarrow \theta = c/2^{k-1}$, $A_k = Y \Leftrightarrow \theta = c/2^k$, and $A_k \neq Z \ \forall \ \theta$, where the angle θ is in its most reduced form. □

Corollary 44. *Strongly transversal logical gates $Z(\theta)^{\otimes n}$ on CSS stabilizer codes must be composed of individual rotations that are an element of the Clifford hierarchy, that is $Z(\theta) \in \mathcal{C}_1^{(k)}$, for some value of k. Moreover, the logical gate that is implemented must also be an element of Clifford hierarchy on the logically encoded subspace.*

Proof. The first statement follows from Theorem 43 and Proposition 1. The second statement follows by considering the action of the individual rotations on the logical states written out in their expansion in terms of the computational basis. Since the individual phases must be of the form $\theta = c/2^k$, any logical phase will be a multiple of such a phase. □

2.4.2.2 Stabilizer codes

We have proven that the only possible strongly transversal operations for CSS codes must be composed of rotations of the form $Z(a/2^k)$. The proof relied upon the fact that for CSS

codes it is trivial to write logical states expressed as a sum of states in the computational basis such that orthogonal logical states do not share common computational states in their sums. Namely, if $\mathcal{S}_X$ represents the stabilizer group formed by all the X stabilizer generators $\mathcal{S}_X = \langle G_i^X \rangle$ of a given CSS code $\mathcal{C}_\mathcal{S}$, then the logical basis states can be expressed as follows:

$$|0_L\rangle = \frac{1}{2^{|\mathcal{S}_X|/2}} \prod_i (I + G_i^X)|0\rangle^{\otimes n} = \frac{1}{\sqrt{|\mathcal{S}_X|}} \sum_{s \in \mathcal{S}_X} |s\rangle,$$
$$X_{L_i}|0_L\rangle = \frac{1}{2^{|\mathcal{S}_X|/2}} \prod_i (I + G_i^X) X_{L_i}|0\rangle^{\otimes n} = \frac{1}{\sqrt{|\mathcal{S}_X|}} \sum_{s \in \mathcal{S}_X} \left| s \otimes g_{X_{L_i}} \right\rangle,$$

where X_{L_i} is the logical X operators composed of individual X operators. Importantly, for two orthogonal states $X_{L_i}|0_L\rangle$ and $X_{L_j}|0_L\rangle$, any two elements of the sum will be orthogonal, that is $\langle s \otimes g_{X_{L_i}} | s' \otimes g_{X_{L_j}} \rangle = 0, \ \forall \ s, \ s' \in \mathcal{S}_X$. This fact is important for showing the restriction of the transversal Z rotations as it implies that any transversal phase rotation must introduce the same phase on all elements of the sum in order to remain in the codespace. Therefore, in order to impose the same argument for restricting the phase in the case of stabilizer codes, we must find a basis for a general stabilizer code that has the same properties with respect to non-overlapping computational basis states when written out as a sum over computational basis states. The following Lemma 45 and Corollary 46 will show such a construction.

Lemma 45. *Given a $[[n, k, d]]$ stabilizer code $\mathcal{C}_\mathcal{S}$, one can always find a set of 2^k logical basis states of the following form:*

$$|\psi_m\rangle = \sum_l i^{a_{m,l}} |m_l\rangle, \tag{2.79}$$

where $a_{m,l}$ is an integer and m_l is an n-bit binary string such that two different logical basis states cannot share any elements in the computation basis expansion. That is, given $|\psi_p\rangle, \ |\psi_q\rangle$ such that $p \neq q$ then $\langle p_s | q_t \rangle = 0 \ \forall \ s, t$.

Proof. Let the stabilizer generators of $\mathcal{S}$ be given by $\langle G_i \rangle_{i=1}^{n-k}$, and the logical Pauli operators be given by $X_{L,j}$, $Z_{L,j}$ for $1 \leq j \leq k$, satisfying $X_{L,j} Z_{L,l} = (-1)^{\delta_{jl}} Z_{L,l} X_{L,j}$. There must exist at least one computational basis state that has non-zero overlap with the stabilizer codespace $\mathcal{C}_\mathcal{S}$. Without loss of generality, we assume that $|0\rangle^{\otimes n}$ is such a state. Then, the

following state is a codestate of $\mathcal{C}_\mathcal{S}$,

$$|\phi\rangle = \frac{1}{2^{(n-k)/2}} \prod_{i=1}^{n-k} (I + G_i)|0\rangle^{\otimes n} = \frac{1}{2^{(n-k)/2}} \sum_i S_i |0\rangle^{\otimes n}$$
$$= \frac{1}{2^{(n-k)/2}} \sum_i |s_i\rangle,$$

where $\{S_i\}$ are the set of all stabilizers generated by $\langle G_i \rangle_{i=1}^{n-k}$ and we have defined the state $|s_i\rangle = S_i|0\rangle^{\otimes n}$. Consider the action of two anti-commuting logical Pauli operators $X_{L,1}$, $Z_{L,1}$, on the state $|\phi\rangle$. We know that $|\phi\rangle$ cannot be an eigenstate of both operators, as no state can be a joint eigenstate of two anti-commuting operators. Therefore, we can consider the following two cases: either $|\phi\rangle$ is an eigenstate of one of the operators, or $|\phi\rangle$ is not an eigenstate of either operator. We shall consider the case of the former first.

Without loss of generality, assume that $Z_{L,1}|\phi\rangle = |\phi\rangle = |\psi_1\rangle$ and $X_{L,1}|\phi\rangle = |\psi_2\rangle \neq \alpha|\psi_1\rangle$ (where α is a global phase). Consider the action of $X_{L,1}|\phi\rangle$:

$$X_{L,1}|\phi\rangle = \frac{1}{2^{(n-k)/2}} \prod_{i=1}^{n-k} (I + G_i) X_{L,1}|0\rangle^{\otimes n}$$
$$= \frac{1}{2^{(n-k)/2}} \sum_i S_i \left|g_{X_{L,1}}\right\rangle,$$

if $\left|g_{X_{L,1}}\right\rangle = |s_j\rangle$ for some j then after the action of the sum over stabilizer operators, the final state $X_{L,1}|\phi\rangle = \alpha|\phi\rangle$ would be a contradiction. Therefore, the state $\left|g_{X_{L,1}}\right\rangle$ must be a computational basis state that is not present in the expansion of $|\phi\rangle$, and moreover, each element of the state $|\psi_2\rangle$ must have zero overlap with the state $|\psi_1\rangle$,

$$|\psi_2\rangle = X_{L,1}|\phi\rangle = \frac{1}{2^{(n-k)/2}} \sum_i \left|g_{X_{L,1}} \oplus s_i\right\rangle.$$

Therefore, two states of the form of Equation 2.79 have been constructed. Consider now the action of the next pair of anti-commuting logical Paulis $X_{L,2}$, $Z_{L,2}$ on the state $|\psi_1\rangle$. Again, since $|\psi_1\rangle$ cannot be a joint eigenstate of both operators, without loss of generality, assume $X_{L,2}|\psi_1\rangle = |\psi_3\rangle \neq \alpha|\psi_1\rangle$. Moreover, it must be that $|\psi_3\rangle \neq \alpha|\psi_2\rangle$ or else the following would be true: $X_{L,1}X_{L,2}|\psi_1\rangle = \alpha X_{L,1}|\psi_2\rangle = \alpha|\psi_1\rangle$, which would imply that $|\psi_1\rangle$ is an eigenstate of two anti-commuting operators, $Z_{L,1}$ and $X_{L,1}X_{L,2}$, which results in a

contradiction. Therefore, we can express the state $|\psi_3\rangle$ as follows:

$$|\psi_3\rangle = X_{L,2}|\psi_1\rangle = \frac{1}{2^{(n-k)/2}} \prod_{i=1}^{n-k} (I + G_i) X_{L,2}|0\rangle^{\otimes n}$$
$$= \frac{1}{2^{(n-k)/2}} \sum_i \left|g_{X_{L,2}} \oplus s_i\right\rangle,$$

where each state in the computational basis expansion must have zero overlap with the states $|\psi_1\rangle$ and $|\psi_2\rangle$. Finally. consider the action of the same anti-commuting pair on the state $|\psi_2\rangle$. As will be shown below, it does not matter which we choose, and thus, without loss of generality, we assume it to be the operator $Z_{L,2}$. First note that if $Z_{L,2}|\psi_2\rangle = \alpha|\psi_1\rangle$, then $|\psi_1\rangle$ would be the joint eigenstate of two anti-commuting Paulis, $Z_{L,1}$ and $X_{L,1}Z_{L,2}$, which is a contradiction. Moreover, if $Z_{L,2}|\psi_2\rangle = \alpha|\psi_3\rangle$ then again $|\psi_1\rangle$ would be the joint eigenstate of two anti-commuting Paulis, $Z_{L,1}$ and $X_{L,1}X_{L,2}Z_{L,2}$. Therefore, the state $Z_{L,2}|\psi_2\rangle = |\psi_4\rangle$ must have zero overlap with the previous established states and can be expressed as follows:

$$|\psi_4\rangle = Z_{L,2}|\psi_2\rangle = \frac{1}{2^{(n-k)/2}} \prod_{i=1}^{n-k} (I + G_i) Z_{L,2} X_{L,1}|0\rangle^{\otimes n}$$
$$= \frac{1}{2^{(n-k)/2}} \sum_i \left|g_{X_{L,1}} \oplus g_{Z_{L,2}} \oplus s_i\right\rangle.$$

Notice the form of $|\psi_3\rangle$ and $|\psi_4\rangle$. By taking the previous states $|\psi_1\rangle$ and $|\psi_2\rangle$ and a pair of non-commuting logical Paulis, for each state in the previous level, we can construct a new state by applying the logical Pauli for which it is not an eigenstate. One can continue the same constructive process for preparing states of the form of Equation 2.79 by taking the m-th pair of anti-commuting logical operators and the 2^{m-1} previous constructed states, thereby creating another 2^{m-1} set of orthogonal states, following similar constraints as laid out above. Applying this to all pairs of logical Pauli gates for the given code, 2^k basis states for the codespace can be constructed.

In the case when the state $|\phi\rangle$ is not an eigenstate of either of the first two logical Pauli gates, $Z_{L,1}$ and $X_{L,1}$, the following modifications have to be made. Let $|\psi_1\rangle = Z_{L,1}|\phi\rangle$ and $|\psi_2\rangle = X_{L,1}|\phi\rangle$. If $|\psi_1\rangle = |\psi_2\rangle$ then by redefining the logical Pauli $\tilde{Z_{L,1}} = X_{L,1}Z_{L,1}$ and $\left|\tilde{\psi_1}\right\rangle = |\phi\rangle$, we recover the original case where $\tilde{Z_{L,1}}\left|\tilde{\psi_1}\right\rangle = \left|\tilde{\psi_1}\right\rangle = |\phi\rangle$ and $|\psi_2\rangle = X_{L,1}|\phi\rangle$. Therefore, the final case to consider is where $|\psi_1\rangle \neq \alpha|\psi_2\rangle$. In this case, they must not have overlapping states in the computational basis, and their expansion can be written as

follows:

$$|\psi_1\rangle = Z_{L,1}|\phi\rangle = \frac{1}{2^{(n-k)/2}} \sum_i \left|g_{Z_{L,1}} \oplus s_i\right\rangle,$$
$$|\psi_2\rangle = X_{L,1}|\phi\rangle = \frac{1}{2^{(n-k)/2}} \sum_i \left|g_{X_{L,1}} \oplus s_i\right\rangle.$$

Again, as in the previous case, consider the action of the pair of logical Pauli gates $X_{L,2}$ and $Z_{L,2}$ on the state $|\psi_1\rangle$. Without loss of generality, assume that $|\psi_1\rangle$ is not an eigenstate of $X_{L,2}$. Unlike the previous case, it is now possible that $X_{L,2}|\psi_1\rangle = |\psi_2\rangle$. However, if this holds, then redefining $\tilde{Z_{L,1}} = X_{L,1}Z_{L,1}X_{L,2}$ and $\left|\tilde{\psi_1}\right\rangle = |\phi\rangle$ we recover the original case with $\tilde{Z_{L,1}}\left|\tilde{\psi_1}\right\rangle = \left|\tilde{\psi_1}\right\rangle$ and $X_{L,2}\left|\tilde{\psi_1}\right\rangle = |\psi_3\rangle$, as well as all redefined operators satisfying the appropriate commutation relations. Otherwise, we can conclude that $X_{L,2}|\psi_1\rangle = |\psi_3\rangle$ and must be orthogonal to the two previous states as well as have the following form:

$$|\psi_3\rangle = X_{L,2}|\psi_1\rangle = \frac{1}{2^{(n-k)/2}} \sum_i \left|g_{Z_{L,1}} \oplus g_{X_{L,2}} \oplus s_i\right\rangle.$$

Therefore, continuing in the same manner as in the previous case, we can construct the set of 2^k logical basis states of the form given by Equation 2.79. □

Corollary 46. *Suppose $\mathcal{C}_\mathcal{S}$ is an n-qubit stabilizer containing k logical qubits. Given 2^k states $|\varphi_m\rangle \in \mathcal{C}_\mathcal{S}$ whose expansion in terms of the computational basis states are all non-overlapping, then these states must be of the form*

$$|\varphi_m\rangle = \sum_l i^{a_{m,l}} |m_l\rangle.$$

Proof. Since all 2^k states are elements of $\mathcal{C}_\mathcal{S}$, they must be convex combinations of any basis chosen for $\mathcal{C}_\mathcal{S}$. Choose the basis given by the states from Lemma 45. Then, if any of the $|\varphi_m\rangle$ were a convex combination of states from such a basis, there must be at least one overlapping state relative to the individual states in its computational basis state expansion. Otherwise the dimension of the logical Hilbert space would be too small to fit all of these logical states. □

We know by Lemma 45, that the computational basis state expansion of $|1_L\rangle$ will be a sum of states such that each state differs from those in the representation of $|0_L\rangle$. Moreover, the gate $Z(\theta)^{\otimes n}$ will preserve all of these basis states, potentially introducing relative phases

between the elements of the sum, however by Corollary 46 the resulting states must also form a basis for the stabilizer code, and in particular for the case of an automorphism the states must form the same logical basis. We can now proceed to proving the set of transversal Z rotations for stabilizer codes in a very similar manner to Theorem 43.

Proposition 2. *A nontrivial qubit stabilizer code (distance $d \geq 2$) can only have strongly transversal Z rotations which are of the form $Z(a/2^k)$.*

Proof. We can represent a general string of Pauli matrices as a binary string using the associations: $\{I \to 00, X \to 10, Y \to 11, Z \to 01\}$. We will write an n-qubit Pauli string as a $2n$-bit string $f = (g|h)$. Here we have separated the string into the two substrings of n-bits (an X (g) and Z (h) substring). We can express the expansion of $|0_L\rangle$ and $|1_L\rangle$ in terms of binary strings of the X operators as follows (without loss of generality assume that $|0\rangle^{\otimes n}$ is in the stabilizer codespace):

$$\begin{aligned}
|0_L\rangle &= (I + Z_L)\prod_i (I + G_i)|0\rangle^{\otimes n} \\
&= |g_0\rangle + (i)^{a_{Z_L}}|g_{Z_L}\rangle + \sum_{i_1}((i)^{a_{i_1}}|g_{i_1}\rangle + (i)^{a_{Z_L\oplus i_1}}|g_{Z_L}\oplus g_{i_1}\rangle) \\
&\quad + \sum_{i_1<i_2}((i)^{a_{i_1\oplus i_2}}|g_{i_1}\oplus g_{i_2}\rangle + (i)^{a_{Z_L\oplus i_1\oplus i_2}}|g_{Z_L}\oplus g_{i_1}\oplus g_{i_2}\rangle) \\
&\quad + \ldots + ((i)^{a_{i_1\oplus\cdots\oplus i_{|G|}}}|g_{i_1}\oplus\cdots\oplus g_{i_{|G|}}\rangle + (i)^{a_{Z_L\oplus i_1\oplus\cdots\oplus i_{|G|}}}\left|g_{Z_L}\oplus g_{i_1}\oplus\cdots\oplus g_{i_{|G|}}\right\rangle),
\end{aligned} \tag{2.80}$$

and

$$\begin{aligned}
|1_L\rangle &= X_L|0_L\rangle \\
&= (i)^{a_{X_L}}|g_{X_L}\rangle + (i)^{a_{X_L\oplus Z_L}}|g_{X_L}\oplus g_{Z_L}\rangle \\
&\quad + \sum_{i_1}((i)^{a_{X_L\oplus i_1}}|g_{X_L}\oplus g_{i_1}\rangle + (i)^{a_{X_L\oplus Z_L\oplus i_1}}|g_{X_L}\oplus g_{Z_L}\oplus g_{i_1}\rangle) + \\
&\quad + \ldots + ((i)^{a_{X_L\oplus i_1\oplus\cdots\oplus i_{|G|}}}|g_{X_L}\oplus g_{i_1}\oplus\cdots\oplus g_{i_{|G|}}\rangle \\
&\quad + (i)^{a_{X_L\oplus Z_L\oplus i_1\oplus\cdots\oplus i_{|G|}}}\left|g_{X_L}\oplus g_{Z_L}\oplus g_{i_1}\oplus\cdots\oplus g_{i_{|G|}}\right\rangle).
\end{aligned} \tag{2.81}$$

Here g_0 is the "all-zeros" string, g_i, g_{X_L}, g_{Z_L} is a binary string corresponding to the location of the X Pauli operators in the respective operators G_i, X_L, Z_L, and $\oplus$ is bitwise XOR. We have added the additional projector $(I+Z_L)$ in order to project to the eigenspace of Z_L

(which may contain X operators). There are relative phases between the different states as the Y and Z Pauli operators from the stabilizers G_i may introduce phases, these phases are characterized by the term $(i)^{a_{\vec{\mu}}}$, where $a_{\vec{\mu}}$ is an integer setting the phase for the particular ket $|\vec{\mu}\rangle$. Importantly, the computational basis states in the sum are all a function of the bit strings g_i which encode the location of X (or Y) operators in the stabilizer G_i. If $|0\rangle^{\otimes n}$ is not in the codespace, another computational basis state can be chosen and the proof would follow identically.

The effect of applying a $Z(\theta)^{\otimes n}$ rotation to the computational basis state composed of the string g will be:

$$Z(\theta)^{\otimes n}|g\rangle = e^{i\pi\theta|g|}|g\rangle. \tag{2.82}$$

Using the result of Lemma 45 we see that for $Z(\theta)^{\otimes n}$ to be a valid logical operator it cannot introduce relative phases between the states in the expansions of Eqs. 2.80 and 2.81, and as such must introduce the same phase to all the states in the expansion (there can be a relative phase between $|0_L\rangle$ and $|1_L\rangle$ however).

For the CSS codes, we assumed that $X_L(Z_L)$ consisted of single qubit unitaries X and I (Z and I). In this case we make no such assumptions,

$$\begin{aligned}
\theta|g_{i_1}| &= 0 \bmod 2 \\
\theta|g_{i_1} \oplus g_{i_2}| &= 0 \bmod 2 \\
&\vdots \\
\theta|g_{i_1} \oplus ... \oplus g_{i_{|G|}}| &= 0 \bmod 2
\end{aligned}$$

$$\begin{aligned}
\theta|g_{X_L}| &\neq 0 \bmod 2 \\
\theta|g_{X_L} \oplus g_{i_1}| &\neq 0 \bmod 2 \\
&\vdots \\
\theta|g_{X_L} \oplus g_{i_1} \oplus ... \oplus g_{i_{|G|}}| &\neq 0 \bmod 2
\end{aligned}$$

$$\begin{aligned}
\theta|g_{Z_L}| &= 0 \bmod 2 \\
&\forall 0 < i_1 < i_2 < ... < i_{|G|} \leq |G|.
\end{aligned}$$

The additional requirement is from $[Z_L, Z(\theta)^{\otimes n}] = 0$. Otherwise $Z(\theta)^{\otimes n}|0_L\rangle = |0_L\rangle \neq Z(\theta)^{\otimes n} Z_L|0_L\rangle$.

These constraints are the same (actually slightly more constraining) as the constraints from the proof of Theorem 43 and the proof carries through analogously. □

Corollary 47. *Given a nontrivial qubit stabilizer code (distance $d \geq 2$), a strongly transversal diagonal logical gate $U = \bigotimes_i Z(\theta)$ can only implement a logical diagonal gate in the Clifford hierarchy, that is $U = Z_L(c/2^k) \in \mathcal{C}_1^{(k+1)}$.*

Proof. Consider the action of the transversal rotation on the logical basis states for the the stabilizer code chosen according to Lemma 45, that is:

$$\bigotimes_{i=1}^{n} Z(\theta)|\psi_m\rangle = \sum_l i^{a_{m,l}} \bigotimes_{i=1}^{n} Z(\theta)|m_l\rangle = \sum_l i^{a_{m,l}} e^{i\pi\theta|m_l|}|m_l\rangle. \tag{2.83}$$

Therefore, the action of $U = \prod_i Z(\theta)$ will result in orthogonal logical basis states to remain orthogonal by the following:

$$\begin{aligned}
\langle\psi_n|U|\psi_m\rangle &= (\sum_p (-i)^{a_{n,p}}\langle n_p|)(\sum_l i^{a_{m,l}} e^{i\pi\theta|m_l|}|m_l\rangle) && (2.84)\\
&= \sum_{l,p} (-i)^{a_{n,p}} i^{a_{m,l}} e^{i\pi\theta|m_l|}\langle n_p|m_l\rangle && (2.85)\\
&= 0, && (2.86)
\end{aligned}$$

since by the choice of basis $\langle n_p|m_l\rangle \; \forall \; l, p$ for $m \neq n$. As such, for this choice of logical basis the logical gate is by definition diagonal. By Proposition 2, the only allowable rotations are of the form $Z(\theta) = Z(a/2^k)$ and the logical gate must introduce the same phase to all the computational basis states in the expansion of the logical states. Therefore, the logical rotation must have the following form:

$$\begin{aligned}
\bigotimes_{i=1}^{n} Z(a/2^k)|\psi_m\rangle &= \sum_l i^{a_{m,l}} e^{i\pi|m_l|a/2^k}|m_l\rangle && (2.87)\\
&= e^{i\pi|m_1|a/2^k} \sum_l i^{a_{m,l}}|m_l\rangle && (2.88)\\
&= e^{i\pi c/2^k} \sum_l i^{a_{m,l}}|m_l\rangle. && (2.89)
\end{aligned}$$

Therefore, since the phases are restricted to be of the form $e^{i\pi c/2^k}$ the logical gate must be in the $(k+1)$-th level of the Clifford hierarchy. □

2.4.2.3 Relaxing strong transversality

We have proven a restriction on the set of diagonal strongly transversal gates for stabilizer codes, however there is no physical reason requiring all qubits in the code have the same rotation applied to them. Namely, each physical qubit could in theory undergo a different rotation. In this subsection, we show that this added freedom will still not allow an increased freedom in the set of diagonal rotations that one can implement.

First, notice that if the transversal operator includes the identity anywhere, it will have no effect on that qubit and therefore, we can formulate the overlap conditions on a new code with that qubit removed. Unlike puncturing a code we are not actually removing the qubit from the code, it is simply not included in the overlap conditions. In what follows, we will assume this process has been implemented, and no identity operators remain. We can do this without any difficulties since our overlap conditions make no use of the commuting properties of stabilizer generators. To prove this more general case we will introduce a new tool: the *decompression lemma*.

Lemma 48. *If an $[[n, k, d]]$ code exists with a transversal $Z_T(\theta) = Z(\theta_1) \otimes Z(\theta_2) \otimes ... \otimes Z(m\theta_n)$ gate, then there exists an $[[n + m - 1, k, 2]]$ code with a transversal $Z'_T(\theta) = Z(\theta_1) \otimes Z(\theta_2) \otimes ... \otimes (Z(\theta_n)^{\otimes m})$ gate.*

Proof. If a code admits a transversal operation $Z_T(\theta) = Z(\theta_1) \otimes Z(\theta_2) \otimes ... \otimes Z(m\theta_n)$, this code's X stabilizer generators and logical operators clearly satisfy the overlap conditions for the transversal Z operator. Now, if we take the last column of the check matrix and repeat it m times, we have a new code which has distance two since a repeated column in the check matrix creates a weight two logical operator. It is easy to see that $Z'_T(\theta) = Z(\theta_1) \otimes Z(\theta_2) \otimes ... \otimes (Z(\theta_n)^{\otimes m})$ satisfies the same overlap conditions on the new code that Z_T satisfied for the original code, and it follows that $Z'_T(\theta)$ implements the same logical operation as $Z_T(\theta)$. Here we have not specified the Z stabilizer generators and it should be noted that in the new code obtained after applying the decompression lemma, there will be $m - 1$ new Z stabilizer generators. □

Before proceeding to the general statement for transversal gates, we present a Lemma that rules out the possibility of irrational angles contributing to the logical gate.

Lemma 49. *Suppose Q is a $[[n, k, d]]$ error-detecting stabilizer code ($d \geq 2$) with a transversal logical gate of the form $Z_T(\theta) = \bigotimes_{i=1}^{n} Z(\theta_i)$ such that $\theta_i \neq p_i/q_i$ for $i \in \Gamma$, where Γ is an arbitrary set of integers from the set $[1, \ldots, n]$. Then, one can replace the irrational*

angles by the identity gate and obtain the same logical gate. That is, let

$$\beta_i = \begin{cases} 0 & i \in \Gamma \\ \theta_i & i \notin \Gamma \end{cases},$$

then $\forall\ |\psi\rangle \in Q$, $\bigotimes_{i=1}^{n} Z(\theta_i)|\psi\rangle = \bigotimes_{i=1}^{n} Z(\beta_i)|\psi\rangle$.

Proof. We allow each $Z(\theta_i)$ to be a Z rotation about any angle, not just a rational angle. Without loss of generality we can assume each θ_i is in the range $(-1, +1)$ and $\theta_i \neq 0$ (since we can just use a new code with that qubit removed). Now we have a transversal gate of the form:

$$Z_L(\theta') := Z(\theta_1) \otimes Z(\theta_2) \otimes ... \otimes Z(\theta_n). \tag{2.90}$$

We will also assume that at least one of the angles is irrational as we have already solved the rational case. Constraints from $Z_L(\theta')|0_L\rangle = |0_L\rangle$ restrict as follows:

$$\begin{aligned} \vec{\theta} \cdot g_{i_1}^T &= 0 \\ \vec{\theta} \cdot (g_{i_1} \oplus g_{i_2})^T &= 0 \\ &\vdots \\ \vec{\theta} \cdot (g_{i_1} \oplus ... \oplus g_{i_{|G_X|}})^T &= 0, \end{aligned}$$

while constraints from $Z_L(\theta')|1_L\rangle = e^{i\pi\theta}|1_L\rangle$ provide the following:

$$\begin{aligned} \vec{\theta} \cdot g_{X_L}^T &= \theta \\ \vec{\theta} \cdot (g_{X_L} \oplus g_{i_1})^T &= \theta \\ &\vdots \\ \vec{\theta} \cdot (g_{X_L} \oplus g_{i_1} \oplus ... \oplus g_{i_{|G_X|}})^T &= \theta \\ & \forall 0 < i_1 < i_2 < ... < i_{|G_X|} \leq |G_X|. \end{aligned}$$

Here the equality is taken over the Real numbers if at least one term in the sum $\vec{\theta} \cdot g_i^T$ is irrational, otherwise the equality is modulo some integer as before.

Some observations:

1. If $\theta_i = \frac{p}{q}\theta_j$, then if $\theta_i - \frac{p}{q}\theta_j = 0 \implies \frac{\theta_i}{q}(q-p) = 0 \implies \theta'(q-p) = 0$. Here $\theta' = \theta_i/q$. We can use the decompression lemma to create a new code where Z_L applies $Z(\theta')$ to $p+q$ qubits.

2. If $\theta_i \neq \frac{p}{q}\theta_j$, then $\theta_i + \theta_j = 0$ iff $\theta_i = 0$ and $\theta_j = 0$. Notice that θ_i and θ_j could be two irrational numbers which are not proportional or an irrational and a rational number (which by definition are not proportional).

3. We can use these observations to reorder the qubits in the code (and possibly apply the decompression lemma to create a new code) to write $\vec{\theta}$ as $Z(1/q) \otimes ... \otimes Z(1/q) \otimes Z(\theta_1) \otimes ... \otimes Z(\theta_1) \otimes Z(\theta_2)....$ Here the $Z(1/q)$ are from the rational part of Z_L (with q a common denomonator) and $Z(\theta_i)$ are the irrational part of Z_L. We have used the decompression lemma to express proportional irrational angles as the same θ_i. Each different i corresponds to irrational angles which are not proportional.

4. Using the second observation we see that the rational angles and each set of proportional irrational angles must individually satisfy the above constraints. We have already discussed the allowable solutions given rational angles. In what follows we will show that no nontrivial solutions exist given irrational angles.

For each θ_i we will have constraints from $Z_L(\theta_i)|0_{L|\theta_i}\rangle = |0_{L|\theta_i}\rangle$ such that,

$$\begin{aligned}
\vec{\theta_i} \cdot g_{i_1|\theta_i}^T &= 0 \\
\vec{\theta_i} \cdot (g_{i_1|\theta_i} \oplus g_{i_2|\theta_i})^T &= 0 \\
&\vdots \\
\vec{\theta_i} \cdot (g_{i_1|\theta_i} \oplus ... \oplus g_{i_{|G_X|}|\theta_i})^T &= 0,
\end{aligned}$$

while constraints from $Z_L(\theta_i)|1_{L|\theta_i}\rangle = e^{i\pi\theta}|1_{L|\theta_i}\rangle$ provide the following,

$$\begin{aligned}
\vec{\theta_i} \cdot g_{X_L|\theta_i}^T &= \theta \\
\vec{\theta_i} \cdot (g_{X_L|\theta_i} \oplus g_{i_1|\theta_i})^T &= \theta \\
&\vdots \\
\vec{\theta_i} \cdot (g_{X_L|\theta_i} \oplus g_{i_1|\theta_i} \oplus ... \oplus g_{i_{|G_X|}|\theta_i})^T &= \theta \\
&\forall 0 < i_1 < i_2 < ... < i_{|G_X|} \leq |G_X|.
\end{aligned}$$

Here $|0_{L|\theta_i}\rangle$ refers to the restriction to qubits which $\vec{\theta_i}$ acts nontrivially upon. Note that it is possible that $\theta = 0$ for some set of proportional irrational angles. As long as $\theta \neq 0$ for some set of proportional irrational angles, then the irrational part of $\vec{\theta}$ has contributed

nontrivially $Z_L(\theta')$. We will only consider the case when $\theta \neq 0$ as the other case is trivial (equivalent to applying the identity).

Now, we will try to find a set of rows of H_X and X_L which satisfy all these conditions. For the underlying code to be nontrivial we require that H_X has no zero columns. We assume that $a \neq 0$, otherwise the transversal operator is trivial ($Z_L(\theta') = I$).

If there is only one row h_1 then it must be all ones and

$$\begin{aligned} \vec{\theta} \cdot g_1^T &= 0 \\ \vec{\theta} \cdot g_{X_L}^T &\neq 0 \\ \vec{\theta} \cdot (g_1 \wedge g_{X_L})^T &= 0, \end{aligned}$$

but $\vec{\theta} \cdot (g_1 \wedge g_{X_L})^T = \vec{\theta} \cdot g_{X_L}^T = 0$ and we have a contradiction.

If H_X is nontrivial and has two rows, the columns of H_X are one of three types:

$$a = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, c = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{2.91}$$

We will refer to the combination of all columns of type a, b, c, by the matrix A, B, C, respectively.

If we have a logical operator X_L, then

$$\begin{aligned} \vec{\theta} \cdot (g_1 \wedge g_{X_L})^T &= \theta(\Delta w_A + \Delta w_C) = 0, \\ \vec{\theta} \cdot (g_2 \wedge g_{X_L})^T &= \theta(\Delta w_B + \Delta w_C) = 0, \\ \vec{\theta} \cdot (g_1 \wedge g_2 \wedge g_{X_L})^T &= \theta(\Delta w_C) = 0, \\ \vec{\theta} \cdot X_L &= \theta(\Delta w_A + \Delta w_B + \Delta w_C) \neq 0. \end{aligned}$$

Here, $\Delta w_A = w_A^+ - w_A^-$ and $w_A^+(w_A^-)$ is the overlap of A and X_L which has support on $H_X^+(H_X^-)$. Since $\theta \neq 0$ the first three constraints imply that $\Delta w_A, \Delta w_B, \Delta w_C = 0$ which imply $|X_L| = 0$ and hence a contradiction.

As we can see the proof proceeds in the same manner as in Sec. 2.4.2.1 with w_i replaced by Δw_i. We reach the same contradiction given any set of proportional irrational angles and have, therefore, proven that transversal gates with single qubit rotations by irrational angles have no effect and are equivalent to applying the identity. □

We now have the tools to proceed to the general proof for the classification of diagonal transversal gates for stabilizer codes.

Proposition 3. *A nontrivial stabilizer code (distance $d \geq 2$) can only have transversal Z rotations which are of the form $Z(a/2^k)$.*

In this case we have a transversal gate

$$Z_T(\theta) := Z(\theta_1) \otimes Z(\theta_2) \otimes ... \otimes Z(\theta_n). \tag{2.92}$$

As per Lemma 49, irrational angles must cancel and therefore cannot contribute to the logical gate. As such, we can assume that $Z(\theta_i)$ is rational. Therefore, we have a transversal gate of the form

$$Z_T(\theta) := Z(p_1/q_1) \otimes Z(p_2/q_2) \otimes ... \otimes Z(p_n/q_n). \tag{2.93}$$

We can find the least common denominator q of $q_1, ..., q_n$ and express this as

$$Z_T(\theta) := Z(p'_1/q) \otimes Z(p'_2/q) \otimes ... \otimes Z(p'_n/q). \tag{2.94}$$

We can also use that $Z(2 + p/q) = Z(p/q)$ to claim $p'_i/q_i \in [0, 2)$ and also assume that $Z(p/q) \neq I$, else we could ignore this operator and the qubit it acted upon as they would not affect the overlap conditions.

Now, we repeatedly apply the decompression lemma until we have an $[[\sum_i p_i, k, 2]]$ code with a transversal gate

$$Z_T(\theta) := Z(1/q) \otimes ... \otimes Z(1/q). \tag{2.95}$$

We have now reduced these more general gates to strongly transversal gates, and the proof follows as before.

2.4.2.4 Classification of all single qubit logical gates

Recall that Zeng *et al.* showed that all single-qubit logical transversal gates for a stabilizer code must have the form [142]:

$$U = L \left(\bigotimes_{j=1}^{n} \mathrm{diag}(1, e^{i\pi\theta_j}) \right) R^\dagger P_\pi, \tag{2.96}$$

where P_π is a permutation matrix of the physical qubits while R and L are transversal single-qubit Clifford operators. Given that the action of the transversal unitary is given

by the composition of a transversal diagonal gate with Clifford operations, by classifying all possible mappings between stabilizer codes given by transversal diagonal operations we can classify all possible transversal gates. We summarize this statement in the following two results.

Proposition 4. *Given two nontrivial (distance $d \geq 2$) n-qubit stabilizer codes $\mathcal{C}_\mathcal{S}$ and $\mathcal{C}_\mathcal{T}$ consisting of r logical qubits, transversal Z rotations of the form $D = \otimes_{j=1}^{r} Z(\theta_j)$ which map $\mathcal{C}_\mathcal{S} \longrightarrow \mathcal{C}_\mathcal{T}$ (and possibly apply a logical unitary in the process) must be of the form $D = \otimes_{j=1}^{r} Z(a_j/2^{k_j})$.*

Proof. Let $\{|\psi_m\rangle_{m=1}^{2^r}\}$ form a logical basis set for the stabilizer code and choose $\mathcal{C}_\mathcal{S}$ to be of the form outlined in Lemma 45. Then, given a transversal application of diagonal gates, the resulting set of states must also form a basis for a stabilizer code (in this case chosen to be $\mathcal{C}_\mathcal{T}$) such that each individual basis state will have the same expansion in terms of the computational basis states. However, the transversal application of diagonal gates may result in relative phases between the states; since the states must form a basis for the stabilizer code of the type given by Lemma 45, the relative phases must be powers of i. Therefore, the transformed states read:

$$|\psi_m\rangle = \sum_l i^{a_{m,l}} |m_l\rangle \xrightarrow{D} |\varphi_m\rangle = e^{i\pi\phi_m} \sum_l i^{a_{m,l}} i^{c_{m,l}} |m_l\rangle.$$

Therefore, repeating the action of the diagonal transversal gate 4 times must return the original set of basis states (with the possible introduction of a phase).

$$|\psi_m\rangle = \sum_l i^{a_{m,l}} |m_l\rangle \xrightarrow{D^4} e^{i4\pi\phi_m} |\psi_m\rangle = e^{i4\pi\phi_m} \sum_l i^{a_{m,l}} |m_l\rangle.$$

We are now back to the original case of classifying transversal diagonal gates for logical gates returning to the same codespace, which we have already classified to be rotations of the form $D = \otimes_{j=1}^{n} Z(a_j/2^{k_j})$. Therefore, we are similarly restricted for the case of logical mappings between stabilizer codes. □

Corollary 50. *Given an n-qubit stabilizer code $\mathcal{C}_\mathcal{S}$ and a transversal unitary U implementing a logical gate, then U must be of the following form:*

$$U = L \left(\bigotimes_{j=1}^{n} Z(a_j/2^{k_j}) \right) R^\dagger P_\pi, \tag{2.97}$$

and must implement a logical unitary in the Clifford hierarchy.

Proof. First note that by the characterization in Ref [142], a transversal unitary gate must have the form given in Equation 2.96. Given that P_π is a permutation unitary, it must necessarily map $\mathcal{C}_\mathcal{S}$ to another stabilizer code $\mathcal{C}_{\mathcal{S}_1}$ with the exact same code properties, namely code distance. Additionally, since $R^\dagger$ is a transversal Clifford operation, it will map any stabilizer code $\mathcal{C}_{\mathcal{S}_1}$ to another stabilizer code $\mathcal{C}_{\mathcal{S}_2}$ whose distance is preserved. The distance is preserved as suppose one is given a Pauli error E such that $\text{wt}(E) < d$, where d is the distance. Then for $|\psi_2\rangle \in \mathcal{C}_{\mathcal{S}_2}$, $R(E|\psi_2\rangle) = E'|\psi_1\rangle$, where $|\psi_1\rangle$ must be in $\mathcal{C}_{\mathcal{S}_1}$ and E' is a modified Pauli error of the same weight as E since R is a transversal Clifford. Therefore since E' is correctable by the distance property of $\mathcal{C}_{\mathcal{S}_1}$, the transformed error must remain correctable in $\mathcal{C}_{\mathcal{S}_2}$. Similarly, $L^\dagger$ must map $\mathcal{C}_\mathcal{S}$ to a stabilizer code $\mathcal{C}_{\mathcal{S}_3}$ with the same distance d. Therefore, since $R^\dagger P_\pi$ maps $\mathcal{C}_\mathcal{S}$ to $\mathcal{C}_{\mathcal{S}_2}$ and L maps $\mathcal{C}_{\mathcal{S}_3}$ to $\mathcal{C}_\mathcal{S}$, in order for U to be a logical operation that preserves the codespace, the operator $D = \otimes_{j=1}^n Z(\theta_j)$ must be a mapping between two nontrivial stabilizer codes $\mathcal{C}_{\mathcal{S}_2}$ and $\mathcal{C}_{\mathcal{S}_3}$. As such, the operator must be constrained by the result of Proposition 4 and the resulting gate is given by the form in Equation 2.97. Finally, since U is a result of the application of a transversal gate belonging to the Clifford hierarchy in composition with Clifford gates, and using the fact that Clifford gates must preserve the given level of the Clifford hierarchy, the resulting gate must belong to the Clifford hierarchy by the same argument as given in Corollary 44. □

2.4.3 Multi-block gates

In this subsection we consider the case of multi-qubit logical gates, where each logical qubit (or codeblock) is encoded in the same $[[n, k, d]]$ error correcting code Q. Zeng *et al.* classified the set of gates that can be transversal across these codeblocks. Namely, if U is a transversal gate on $Q^{\otimes r}$, then for each $j \in [n]$ either $U_j \in \mathcal{L}_r$ or $U_j = L_1 V L_2$, where $L_1, L_2 \in \mathcal{L}_1^{\otimes r}$ are local Clifford gates, and V keeps the linear span of the group elements of $\langle \pm Z_j^{(i)}, i \in [r] \rangle$. This work focuses on the gates V, which must be diagonal in order to preserve the span of the group of Z operators across qubits at a fixed i.

2.4.3.1 Strong transversality for two-qubit logical gates

Consider first the implementation of a logical diagonal gate in the case of two codeblocks, where the logical gate is implemented by using a strongly transversal gate. That is, consider the implementation of the diagonal two-qubit logical gate by applying a given two-qubit gate, $U = \sum_j e^{i\pi\theta_j}|j\rangle\langle j|$ transversally $U^{\otimes n}$ among the corresponding pair of qubits between

the codeblocks. The desired logical gate to be implemented has the form

$$U_L = \sum_j e^{i\pi\omega_j} |j\rangle\langle j|_L,$$

where the states $\{|j\rangle_L\}_j = \{|00\rangle_L, |01\rangle_L, |10\rangle_L, |11\rangle_L\}$ are two-qubit logical states spanning the two codeblocks.

As in the single block case, the desired action of the logical gate on the logical states will impose a restriction on the form of the two-qubit physical gates that can be implemented in a strongly transversal manner. In the case of a quantum CSS code, the above logical gate description will have the following form, similar to the construction in Theorem 43:

$$\begin{aligned}
U_L|00\rangle_L &= U^{\otimes n} \prod_{i=1}^{|G_X|} (I + G_{X_i})|0\rangle^{\otimes n} \prod_{j=1}^{|G_X|} (I + G_{X_j})|0\rangle^{\otimes n} \\
&= U^{\otimes n} \Big(|g_0\rangle + \sum_{i_1} |g_{i_1}\rangle + \sum_{i_1<i_2} |g_{i_1} \oplus g_{i_2}\rangle + \ldots + \big|i_1 \oplus \ldots \oplus i_{|G_X|}\big\rangle \Big) \\
&\qquad \otimes \Big(|g_0\rangle + \sum_{j_1} |g_{j_1}\rangle + \sum_{j_1<j_2} |g_{j_1} \oplus g_{j_2}\rangle + \ldots + \big|j_1 \oplus \ldots \oplus j_{|G_X|}\big\rangle \Big) \\
&= e^{i\pi\omega_{00}} |00\rangle_L.
\end{aligned}$$

Note that each of the $4^{|G_X|}$ states in the summation of the $|00\rangle_L$ state are computational basis states and will not change under the action of $U^{\otimes n}$ except for the possible addition of a phase. Therefore, in order to remain a codeword, all states in the expansion must have the same phase.

Without loss of generality, one can assume that the phase $\theta_{00} = 0$ (this maps to a global phase freedom in the logical gate U_L). Consider now the phases introduced on all $2^{|G_X|}$ states in the expansion of the first qubit, along with the state $|g_0\rangle$ in the expansion of the second qubit. For clarity, we will list the state in the expansion, along with the corresponding condition imposed on its phase.

$$\begin{aligned}
&|g_0\rangle|g_0\rangle : && n\theta_{00} = \omega_0 = 0 && \mod 2 \\
&|g_{i_1}\rangle|g_0\rangle : && |g_{i_1}|\theta_{10} + (n - |g_{i_1}|)\theta_{00} = |g_{i_1}|\theta_{10} = 0 && \mod 2 \\
&|g_{i_1} \oplus g_{i_2}\rangle|g_0\rangle : && |g_{i_1} \oplus g_{i_2}|\theta_{10} = 0 && \mod 2 \\
&\vdots \\
&\Big|g_{i_1} \oplus \ldots \oplus g_{i_{|G_X|}}\Big\rangle|g_0\rangle : && |g_{i_1} \oplus \ldots \oplus g_{i_{|G_X|}}|\theta_{10} = 0 && \mod 2
\end{aligned}$$

These conditions are equivalent to the conditions derived on a single codeblock and therefore θ_{10} is restricted to be an integer multiple of $1/2^c$ (when paired with the appropriate restrictions on the logical phase ω_{10} as given below). In a very similar manner, a set of constraints can be obtained for the angle θ_{01} due to the symmetry of the two codes:

$$\begin{aligned}
&|g_0\rangle|g_{j_1}\rangle : && |g_{j_1}|\theta_{01} + (n - |g_{j_1}|)\theta_{00} = |g_{i_1}|\theta_{01} = 0 && \mod 2\\
&|g_0\rangle|g_{j_1} \oplus g_{j_2}\rangle : && |g_{j_1} \oplus g_{j_2}|\theta_{01} = 0 && \mod 2\\
&\vdots\\
&|g_0\rangle\left|g_{j_1} \oplus \ldots \oplus g_{j_{|G_X|}}\right\rangle : && |g_{j_1} \oplus \ldots \oplus g_{j_{|G_X|}}|\theta_{01} = 0 && \mod 2
\end{aligned}$$

Therefore, the phase angle θ_{01} will also be restricted to be an integer multiple of $1/2^c$.

In order to obtain a restriction on the phase angle θ_{11}, higher order state vectors must be considered in both expansions of the logical $|0\rangle_L$ states. Consider the full summation over states in the expansion of the first block, along with state vectors in the second block of the form $|g_{j_1}\rangle$.

$$\begin{aligned}
&|g_{i_1}\rangle|g_{j_1}\rangle : && |g_{i_1} \wedge g_{j_1}|(\theta_{11} - \theta_{01} - \theta_{10}) + |g_{i_1}|\theta_{10} + |g_{j_1}|\theta_{01} = 0 && \mod 2\\
&|g_{i_1} \oplus g_{i_2}\rangle|g_{j_1}\rangle : && |(g_{i_1} \oplus g_{i_2}) \wedge g_{j_1}|(\theta_{11} - \theta_{01} - \theta_{10})\\
& && \quad + |g_{i_1} \oplus g_{i_2}|\theta_{10} + |g_{j_1}|\theta_{01} = 0 && \mod 2\\
&\vdots\\
&\left|g_{i_1} \oplus \ldots \oplus g_{i_{|G_X|}}\right\rangle|g_{j_1}\rangle : && |(g_{i_1} \oplus \ldots \oplus g_{i_{|G_X|}}) \wedge g_{j_1}|(\theta_{11} - \theta_{01} - \theta_{10})\\
& && \quad + |g_{i_1} \oplus \ldots \oplus g_{i_{|G_X|}}|\theta_{10} + |g_{j_1}|\theta_{01} = 0 && \mod 2
\end{aligned}$$

First notice that every term other than the first term in each of the conditions will be equal to zero (mod 2), as a result of the set of conditions imposed on the phase angles θ_{01} and θ_{10}. Therefore, what remains are conditions on the phase difference $\theta'_{11} = (\theta_{11} - \theta_{01} - \theta_{10})$, which is equivalent to a condition on θ_{11}. Moreover, consider the expansion of the direct sum as given by Equation 2.76:

$$\begin{aligned}
|(g_{i_1} \oplus g_{i_2}) \wedge g_{j_1}|\theta'_{11} &= |(g_{i_1} \wedge g_{j_1}) \oplus (g_{i_2} \wedge g_{j_1})|\theta'_{11}\\
&= \Big(|g_{i_1} \wedge g_{j_1}| + |g_{i_2} \wedge g_{j_1}| - 2|g_{i_1} \wedge g_{i_2} \wedge g_{j_1}|\Big)\theta'_{11} = 0\\
\Rightarrow 2|g_{i_1} \wedge g_{i_2} \wedge g_{j_1}|\theta'_{11} &= 0,
\end{aligned}$$

where the implication in the final line is due to $|(g_i \wedge g_{j_1})|\theta'_{11} = 0$, for all i from the first set of constraints of the state vector $|g_{i_1}\rangle|g_{j_1}\rangle$. Similarly,

$$\begin{aligned}
|(g_{i_1} \oplus g_{i_2} \oplus g_{i_3}) \wedge g_{j_1}|\theta'_{11} &= |(g_{i_1} \wedge g_{j_1}) \oplus (g_{i_2} \wedge g_{j_1}) \oplus (g_{i_3} \wedge g_{j_1})|\theta'_{11} \\
&= \Big(|g_{i_1} \wedge g_{j_1}| + |g_{i_2} \wedge g_{j_1}| + |g_{i_3} \wedge g_{j_1}| \\
&\quad - 2|g_{i_1} \wedge g_{i_2} \wedge g_{j_1}| - 2|g_{i_1} \wedge g_{i_3} \wedge g_{j_1}| - 2|g_{i_2} \wedge g_{i_3} \wedge g_{j_1}| \\
&\quad + 4|g_{i_1} \wedge g_{i_2} \wedge g_{i_3} \wedge g_{j_1}|\Big)\theta'_{11} = 0 \\
\Rightarrow 4|g_{i_1} \wedge g_{i_2} \wedge g_{i_3} \wedge g_{j_1}|\theta'_{11} &= 0.
\end{aligned}$$

The final implication is a consequence of the above condition on $2|g_{i_1} \wedge g_{i_2} \wedge g_{j_1}|$. The same procedure will follow for all states in the expansion, and conditions on θ'_{11} can thus be modified as:

$$\begin{aligned}
&|g_{i_1}\rangle|g_{j_1}\rangle : && |g_{i_1} \wedge g_{j_1}|\theta'_{11} = 0 && \mod 2 \\
&|g_{i_1} \oplus g_{i_2}\rangle|g_{j_1}\rangle : && 2|g_{i_1} \wedge g_{i_2} \wedge g_{j_1}|\theta'_{11} = 0 && \mod 2 \\
&|g_{i_1} \oplus g_{i_2} \oplus g_{i_3}\rangle|g_{j_1}\rangle : && 4|g_{i_1} \wedge g_{i_2} \wedge g_{i_3} \wedge g_{j_1}|\theta'_{11} = 0 && \mod 2 \\
&\vdots \\
&\Big|g_{i_1} \oplus \ldots \oplus g_{i_{|G_X|}}\Big\rangle|g_{j_1}\rangle : && 2^{|G_X|-1}|g_{i_1} \wedge \ldots \wedge g_{i_{|G_X|}} \wedge g_{j_1}|\theta'_{11} = 0 && \mod 2.
\end{aligned}$$

Given that the two codebocks are encoded in the same quantum error correcting code, these conditions are a modified version of the conditions derived in the single block case, where an extra factor of 2 is present in all of the constraints. This extra factor of 2 will have a consequence on the type of logical gates that can be implemented transversally and will limit the 2-qubit gates to reside in the same level of the Clifford hierarchy as the 1-qubit gates that can be implemented for a given code.

Finally, consider the action of the strongly transversal gate on the logical states $|01\rangle_L$, $|10\rangle_L$, and $|11\rangle$ when performing a logical X_L on the appropriate qubit(s). The resulting set of conditions impose a restriction on the logical phases ω_{01}, ω_{10}, and ω_{11}. For the logical

state $|01\rangle_L$ the conditions are:

$$
\begin{aligned}
&|g_0\rangle|g_L\rangle : && |g_L|\theta_{01} = \omega_{01} && \mod 2\\
&|g_0\rangle|g_{j_1} \oplus g_L\rangle : && |g_{j_1} \oplus g_L|\theta_{01} = \omega_{01} && \mod 2\\
&\vdots\\
&|g_0\rangle\left|g_{j_1} \oplus \ldots \oplus g_{j_{|G_X|}} \oplus g_L\right\rangle : && |g_{j_1} \oplus \ldots \oplus g_{j_{|G_X|}} \oplus g_L|\theta_{01} = \omega_{01} && \mod 2.
\end{aligned}
$$

Therefore these restrictions, along with the conditions for the phase θ_{01}, will impose the restriction of the form of phases that can be applied, as shown in the single block case. In the exact same manner, restriction on the phases θ_{10} and ω_{10} are obtained. Finally, in order to obtain restrictions on the phase ω_{11}, consider the following:

$$
\begin{aligned}
&|g_L\rangle|g_L\rangle : && |g_L|\theta_{11} = \omega_{11} && \mod 2\\
&|g_L\rangle|g_{j_1} \oplus g_L\rangle : && |g_{j_1} \oplus g_L|\theta_{11} = \omega_{11} && \mod 2\\
&\vdots\\
&|g_L\rangle\left|g_{j_1} \oplus \ldots \oplus g_{j_{|G_X|}} \oplus g_L\right\rangle : && |g_{j_1} \oplus \ldots \oplus g_{j_{|G_X|}} \oplus g_L|\theta_{11} = \omega_{11} && \mod 2,
\end{aligned}
$$

there conditions are in fact exactly the same as the overlap between the g_L string in both states has to be the same since the two codes are encoded into the same codeblock. Therefore, the exact same overlap conditions are obtained for the phases θ_{11} and ω_{11}.

These set of conditions result in the following Theorem for two-qubit transversal diagonal gates.

Theorem 51. *Suppose for a given quantum error correcting code the logical gate $Z_L(1/2^k)$ can be obtained by applying a transversal $Z(1/2^k)^{\otimes n}$ on the underlying physical qubits, yet the transversal application of the logical gate $Z_L(1/2^{k+1})$ is impossible due to code constraints. Then, the two-qubit logical gate U_L that can be applied transversally is in the same level of the Clifford hierarchy as the single-qubit logical gate that transversal single qubit diagonal gates can implement, i.e. $Z_L(1/2^k) \in \mathcal{C}_1^{(k+1)}$ and $U_L \in \mathcal{C}_2^{(k+1)}$. More specifically, the set of two-qubit diagonal gates $U = \sum_j e^{i\pi\theta_j}|j\rangle\langle j|$ that can implement a logical two-qubit operation by applying such gates transversally $U^{\otimes n}$ will be restricted to the angles (up to a global phase freedom),*

$$
\begin{aligned}
\theta_{00} &= 0,\\
\theta_{01} &= a/2^k,\\
\theta_{10} &= b/2^k,\\
\theta_{11} &= a/2^k + b/2^k + c/2^{k-1},
\end{aligned}
$$

where a, b and c are arbitrary integers. The resulting two-qubit logical gate will have the form (up to arbitrary Clifford gates),

$$U_L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\pi\alpha/2^{k-1}} & 0 & 0 \\ 0 & 0 & e^{i\pi\beta/2^k} & 0 \\ 0 & 0 & 0 & e^{i\pi\gamma/2^k} \end{pmatrix},$$

where $\beta = 0 \Leftrightarrow \gamma = 0$.

Proof. The first result of the proof is proved in the above subsection by the resulting constraints on the angles that can be implemented transversally. More specifically, since $Z_L(1/2^k)$ can be implemented transversally, we know that the generators of stabilizer of the code must satisfy,

$$\begin{aligned} |g_{i_1}| &= 0 \mod 2^k \\ 2|g_{i_1} \wedge g_{i_2}| &= 0 \mod 2^k \\ &\vdots \\ 2^{k-1}|g_{i_1} \wedge \ldots \wedge g_{i_k}| &= 0 \mod 2^k, \end{aligned}$$

for all choices of valid indices $\{i_1, \ldots i_k\}$. Moreover, since the code is constrained to not be able to implement $Z(1/2^{k+1})$ transversally, we know the following must be true for some choice of indices $\{\mu_1, \ldots, \mu_{k+1}\}$,

$$2^k|g_{\mu_1} \wedge \ldots \wedge g_{\mu_{k+1}}| \neq 0 \mod 2^{k+1}.$$

Therefore, given the resulting set of constraints on the angle difference $\theta'_{11} = \theta_{11} - \theta_{01} - \theta_{10}$,

$$\begin{aligned} |g_{i_1} \wedge g_j| &= 0 \mod q \\ 2|g_{i_1} \wedge g_{i_2} \wedge g_j| &= 0 \mod q \\ &\vdots \\ 2^{k-1}|g_{i_1} \wedge \ldots \wedge g_{i_k} \wedge g_j| &= 0 \mod q, \end{aligned}$$

where $1/q$ is the desired angular rotation. The above conditions will not be able to be satisfied for $q = 2^k$ as for the indices $\{\mu_1, \cdots, \mu_{k+1}\}$ the following would lead to a contradiction with the final condition:

$$2^k|g_{\mu_1} \wedge \ldots \wedge g_{\mu_{k+1}}| \neq 0 \mod 2^{k+1} \Longrightarrow 2^{k-1}|g_{\mu_1} \wedge \ldots \wedge g_{\mu_{k+1}}| \neq 0 \mod 2^k.$$

Conversely, we know that we can satisfy the above equations for $q = 2^{k-1}$ by using an implication from the single-qubit conditions that must be satisfied for all indices $\{i_1, \ldots i_k\}$,

$$2^l | g_{i_1} \wedge \ldots \wedge g_{i_{l+1}} | = 0 \mod 2^k \Longrightarrow 2^{l-1} | g_{i_1} \wedge \ldots \wedge g_{i_{l+1}} | = 0 \mod 2^{k-1}.$$

Therefore, combining the results for the single qubit gates on the same restrictions and the above observations we know the angle θ'_{11} is restricted to have the form $c/2^{k-1}$. Since the angles θ_{01} and θ_{10} satisfy the same restrictions as the single qubit block case, this completes the proof of the first claim of Theorem 51 regarding the allowable angles which a multi-qubit diagonal gate can implementing a logical multi-qubit gate via a transversal application of the chosen gates. In order to obtain a description of the two-qubit logical gate, we can consider the set of equations that provide the restrictions on the allowable angles in order to obtain an explicit description of the logical angle that can be applied:

$$\begin{aligned}
U^{\otimes n}|00\rangle_L &= e^{i\pi n\theta_{00}}|00\rangle_L = |00\rangle, \\
U^{\otimes n}|01\rangle_L &= e^{i\pi|g_i|\theta_{01}}|01\rangle_L = e^{i\pi a|g_i|/2^k}|01\rangle_L = e^{i\pi\alpha/2^k}|01\rangle_L, \\
U^{\otimes n}|10\rangle_L &= e^{i\pi|g_i|\theta_{10}}|10\rangle_L = e^{i\pi b|g_i|/2^k}|10\rangle_L = e^{i\pi\beta/2^k}|10\rangle_L, \\
U^{\otimes n}|11\rangle_L &= e^{i\pi|g_i\wedge g_j|\theta_{11}}|11\rangle_L = e^{i\pi|g_i|(\theta_{01}+\theta_{10}+\theta'_{11})}|11\rangle_L = e^{i\pi(a+b+2c)|g_i|/2^k}|11\rangle_L \\
&= e^{i\pi(\alpha+\beta+2\eta)/2^k}|11\rangle_L.
\end{aligned}$$

The above equations must hold for any choice of the weight of the individual (or pairs) of stabilizer generators and in the last equation we have chosen g_j to equal g_i as we have a freedom over which j we choose. We have also introduced the integers $\alpha = a|g_i|$, $\beta = b|g_i|$, and $\eta = c|g_i|$. Consider the angle that is applied by the logical operation to the state $|11\rangle_L$ in more detail, $(\alpha + \beta + 2\eta)/2^k$. If both α and β are odd, then the overall angle will be of the form $\gamma/2^{k-1}$. If either α or β are even (or zero), but not both, then the angle will have the form $\gamma/2^k$; however this would mean that the other angle could then be expressed in its most reduced form as $\alpha'/2^{k-1}$. Finally, if both are even, it follows that all these angles can be reduced and shown to be proportional to $1/2^{k-1}$. Therefore, up to a relabelling of logical basis states (which can be achieved using either a logical X_L or $CNOT$ gate), the two-qubit logical gate can be expressed in the form

$$U_L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\pi\alpha/2^k} & 0 & 0 \\ 0 & 0 & e^{i\pi\beta/2^k} & 0 \\ 0 & 0 & 0 & e^{i\pi\gamma/2^{k-1}} \end{pmatrix},$$

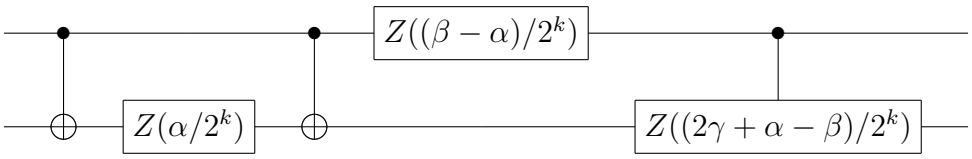
where $\alpha = 0 \Leftrightarrow \beta = 0$, such that either both phases are zero or the two phases are proportional to $1/2^k$ (in the case when only one is zero, then the gate will be a product of

a single logical qubit rotation proportional to $Z(1/2^k)$ and a controlled-$Z(1/2^{k-1})$ which are both in $\mathcal{C}_2^{(k)}$). To prove the final statement of Theorem 51 we must show that the above gate is contained within the Clifford hierarchy at the $(k+1)$-th level of the two-qubit Clifford hierarchy, $U_L \in \mathcal{C}_2^{(k+1)}$.

We shall prove this by induction. Begin with the case $k = 1$, and without loss of generality, assume $\alpha < \beta$, that both are not equal to zero, and that the angles are written in their most reduced form (if both $\alpha = \beta = 0$. Now the proof of the base case is trivial, as it becomes a controlled-Z gate which is clearly in $\mathcal{C}_2^{(2)}$, a Clifford gate). We can rewrite the logical gate as

$$U_L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\pi\alpha/2^k} & 0 & 0 \\ 0 & 0 & e^{i\pi\beta/2^k} & 0 \\ 0 & 0 & 0 & e^{i\pi\gamma/2^{k-1}} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\pi\alpha/2^k} & 0 & 0 \\ 0 & 0 & e^{i\pi\alpha/2^k} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{i\pi(\beta-\alpha)/2^k} & 0 \\ 0 & 0 & 0 & e^{i\pi\gamma/2^{k-1}} \end{pmatrix}$$
$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & e^{i\pi\alpha/2^k} & 0 & 0 \\ 0 & 0 & e^{i\pi\alpha/2^k} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{i\pi(\beta-\alpha)/2^k} & 0 \\ 0 & 0 & 0 & e^{i\pi(\beta-\alpha)/2^k} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\pi(2\gamma+\alpha-\beta)/2^k} \end{pmatrix}.$$

The above sequence of unitaries can be expressed as the following sequence of gates:



In this case, $k = 1$, and all of the single qubit gates are achieved by repeated action of the Clifford phase gate $S = \mathrm{diag}(1, i)$. The two-qubit coupling gate is actually the application of a controlled-Z gate since $\alpha \neq 0$, $\beta \neq 0$, and both are odd; therefore, their difference is even and the gate can be expressed in the form $Z(\zeta/2)$. Since all of these gates are in $\mathcal{C}_2^{(2)}$ and the first two levels of the Clifford hierarchy form a group, the resulting composition is an element of $\mathcal{C}_2^{(2)}$.

Assume the claim holds for $k-1$; we will now show that it holds for k. By definition, if $U_L \in \mathcal{C}_2^{(k)}$ it must map any two-qubit Pauli to an element in $\mathcal{C}_2^{(k-1)}$ when conjugating by U_L. We need only consider the action of U_L on the Pauli X elements, as the action on

Pauli-Z is trivial since diagonal gates commute. Consider the following:

$$U_L(X\otimes I)U_L^\dagger = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & e^{i\pi\alpha/2^k} & 0 & 0\\ 0 & 0 & e^{i\pi\beta/2^k} & 0\\ 0 & 0 & 0 & e^{i\pi\gamma/2^{k-1}}\end{pmatrix}\begin{pmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\end{pmatrix}\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & e^{-i\pi\alpha/2^k} & 0 & 0\\ 0 & 0 & e^{-i\pi\beta/2^k} & 0\\ 0 & 0 & 0 & e^{-i\pi\gamma/2^{k-1}}\end{pmatrix}$$
$$= \begin{pmatrix}0 & 0 & e^{-i\pi\beta/2^k} & 0\\ 0 & 0 & 0 & e^{i\pi(\alpha-2\gamma)/2^k}\\ e^{i\pi\beta/2^k} & 0 & 0 & 0\\ 0 & e^{-i\pi(\alpha-2\gamma)/2^k} & 0 & 0\end{pmatrix} = A.$$

Through the action of CNOT gates, we can map the above operator to the following:

$$e^{-i\pi\beta/2^k}\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & e^{i\pi(\alpha+\beta-2\gamma)/2^k} & 0 & 0\\ 0 & 0 & e^{i\pi 2\beta/2^k} & 0\\ 0 & 0 & 0 & e^{-i\pi(\alpha-\beta-2\gamma)/2^k}\end{pmatrix}.$$

Note that the left or right action of any Clifford gate will not change the level of an element in the Clifford hierarchy, as proven in Prop. 3 in Ref [141]. Therefore we can show that the above gate is in $\mathcal{C}_2^{(k-1)}$ which is equivalent to showing that $A\in\mathcal{C}_2^{(k-1)}$. We know we can write the integers α and β as $\alpha = 2k_\alpha+1$ and $2k_\beta+1$. Consider the following angular expressions:

$$\frac{\alpha+\beta-2\gamma}{2^k} = \frac{2(k_\alpha+k_\beta)-2\gamma+2}{2^k} = \frac{(k_\alpha+k_\beta)-(\gamma-1)}{2^{k-1}},$$
$$\frac{\alpha-\beta-2\gamma}{2^k} = \frac{2(k_\alpha-k_\beta)-2\gamma}{2^k} = \frac{(k_\alpha-k_\beta)-\gamma}{2^{k-1}}.$$

Since $(k_\alpha+k_\beta)$ is even if and only if $(k_\alpha-k_\beta)$ is even, one of the numerators in the final expression will be even, and as such, one of the above angles will necessarily be of the form $1/2^{k-2}$. Therefore, up to a logical Clifford operation (which preserves the level of the Clifford hierarchy), the gate A will have the form:

$$\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & e^{i\pi\alpha'/2^{k-1}} & 0 & 0\\ 0 & 0 & e^{i\pi\beta'/2^{k-1}} & 0\\ 0 & 0 & 0 & e^{i\pi\gamma'/2^{k-2}}\end{pmatrix},$$

which by the induction hypothesis is an element of $\mathcal{C}_2^{(k-1)}$. Finally, we must show the same property for the following mapping:

$$U_L(I\otimes X)U_L^\dagger = \begin{pmatrix}1&0&0&0\\0&e^{i\pi\alpha/2^k}&0&0\\0&0&e^{i\pi\beta/2^k}&0\\0&0&0&e^{i\pi\gamma/2^{k-1}}\end{pmatrix}\begin{pmatrix}0&1&0&0\\1&0&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}\begin{pmatrix}1&0&0&0\\0&e^{-i\pi\alpha/2^k}&0&0\\0&0&e^{-i\pi\beta/2^k}&0\\0&0&0&e^{-i\pi\gamma/2^{k-1}}\end{pmatrix}$$
$$= \begin{pmatrix}0&e^{-i\pi\alpha/2^k}&0&0\\e^{i\pi\alpha/2^k}&0&0&0\\0&0&0&e^{i\pi(\beta-2\gamma)/2^k}\\0&0&e^{-i\pi(\beta-2\gamma)/2^k}&0\end{pmatrix} = B.$$

Up to logical Clifford operations, the gate B has the following form:

$$\begin{pmatrix}1&0&0&0\\0&e^{i\pi 2\alpha/2^k}&0&0\\0&0&e^{i\pi(\alpha\beta-2\gamma)/2^k}&0\\0&0&0&e^{-i\pi(\beta-\alpha-2\gamma)/2^k}\end{pmatrix}.$$

We can see that this has the same form as the case above when the roles of α and β are exchanged; therefore $B\in\mathcal{C}_2^{(k-1)}$ by the induction hypothesis. Furthermore, since $U_L(Z\otimes I)U_L^\dagger = Z\otimes I$ and $U_L(I\otimes Z)U_L^\dagger = I\otimes Z$, we conclude that $U_L\in\mathcal{C}_2^{(k)}$, thus proving the induction hypothesis correct. □

It is fairly straightforward to note that the equivalent of Proposition 3 will also apply in the two-qubit case. That is, the gate restrictions will also apply to general transversal operations and not just to those that are strongly transversal by using the *Decompression Lemma*.

2.4.4 Summary & outlook

Zeng *et al.* classified the set of single-qubit logical transversal gates [142], showing that they must result from the application of single-qubit diagonal gates in addition to possible local Clifford operations and permutations (SWAP gates). In this work we have characterized the set of individual diagonal gates that can result in the application of a non-trivial logical gate, concluding that all of the entries must be of the form $e^{i\pi c/2^k}$. This severely limits the

set of logical gates that can be implemented in a transversal manner for qubit stabilizer codes. It also provides an important result for fault-tolerant quantum computing, as it rules out the possibility of finding transversal implementations for important gates in certain decomposition algorithms, such as the V gate. It also places restrictions on new fault-tolerant schemes which have thus far used a combination of codes to achieve fault-tolerant quantum computation.

Additionally, we have extended our analysis to two-qubit logical gates through the use of two-qubit physical diagonal gates, showing that a very similar restriction holds. In fact, in both the single and two-qubit case, the logical gates that can be implemented by transversal diagonal gate application must belong to the Clifford hierarchy, and moreover, both the single and two-qubit gates that can be implemented for a given code must reside at the same level of the hierarchy. We conjecture that this is true for all multi-qubit gates.

Open questions for future research would be to classify the set of physical diagonal gates that can implement a non-trivial logical gate for qudit systems. Additionally, it would be interesting to consider the set of logical gates that can be generated by coupling two codeblocks corresponding to different quantum error correcting codes, and determine if the same logical gate restrictions apply. Classifying the set of transversal gates for other types of codes is another interesting direction for future research which could provide insight into ways to circumvent the gate restrictions introduced in this work.

Additionally, many magic state distillation schemes use CSS codes to distill higher fidelity magic states. These schemes use stabilizer codes with strongly transversal gates directly related to the magic state which the scheme distills. Our results suggest that magic state distillation based on these methods, can only distill gates in the Clifford hierarchy. However, it is worth noting that the 5-to-1 distillation scheme proposed in Ref. [32] is not based on such a method and does allow for the distillation of a state that can implement $Z(\pi/6)$.

Chapter 3

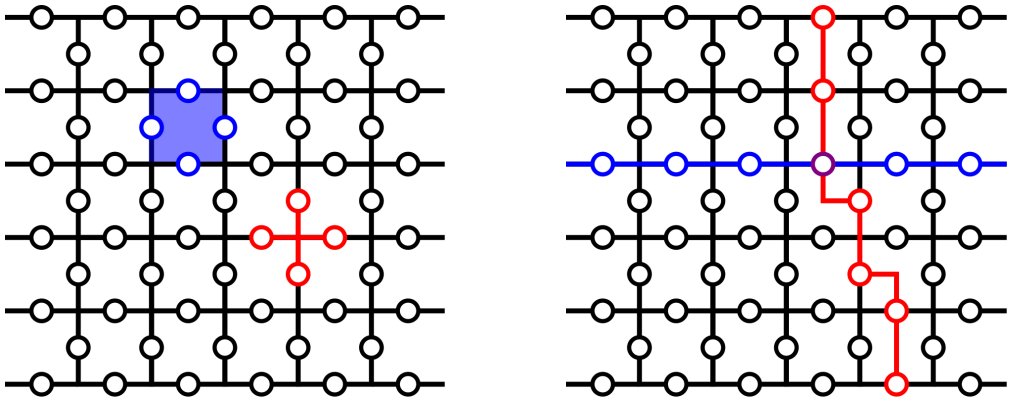
Topological methods for universality

3.1 Topological codes

In Chapter 2, many examples of codes were presented in the context of concatenated quantum error correction. While concatenated codes provide the simplest means to arriving at a fault-tolerance threshold, there are some drawbacks to their implementation. Firstly, the weight of the stabilizer generators naturally get very large as the Pauli operators that form the stabilizer generators at the first level now become encoded Pauli operators, of weight d, the distance of the code. Therefore, in order to measure the stabilizers of the code for the purposes of error correction, the joint Pauli operator of a high number of qubits need to be measured. This can be countered by using techniques such as Shor, Steane, or Knill error correction [126, 130, 87], however all of these techniques add extra ancillary overhead as well as require to post-select logical state preparation needed for their implementation. Another more subtle point, which at first may not seem overly impactful, is that the distance of the code takes large (exponential) jumps from one level to the next. Topological quantum codes avoid these complications by encoding information in error correcting codes where the stabilizers are finite-sized local observables and logical information is stored in non-local degrees of freedom.

3.1.1 The Toric and surface codes

Perhaps most famous among the class of topological codes is the Toric code. Originally defined with periodic boundary conditions, the Toric code encodes two logical qubits on



(a) (b)

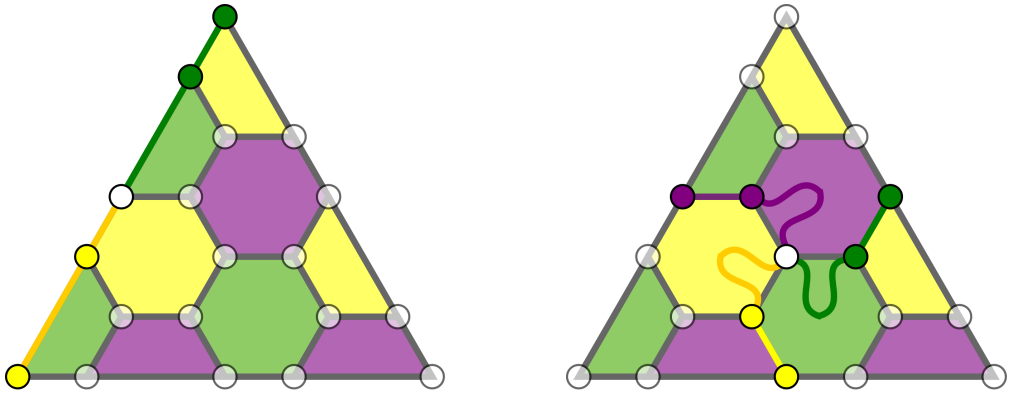
Figure 3.1: The surface code with distance $d = 6$. (a) Layout of the surface code with qubits on the edges. Z stabilizers are defined by square plaquette operators, shown in Blue. X stabilizers are defined by star operators, shown in Red. (b) Logical operators for the surface code, Z_L in blue and X_L in red. The Z_L logical operator connects rough boundaries on left and right while the X_L logical operators connect the smooth boundaries on top and bottom.

a toroidal surface with logical operations corresponding to non-contractable loops on the surface of the torus [85]. A planar version of the Toric code is obtained by cutting the torus, no longer imposing periodic boundary conditions, and thus negating one of the logical degrees of freedom. This version of the Toric code is given the popular name surface code.

The stabilizers of the Toric (and surface) code are weight-4 X and Z stabilizers, thus forming a CSS code. Qubits are placed on the edges of a lattice, with Z stabilizers being defined by plaquette operators, and X stabilizers being defined by star operators, see Fig. 3.1 for a pictoral representation. If we label the logical subspace as the ground state of the stabilizer Hamiltonian:

$$H = -\sum_{\mathcal{S}_X} G_i^X - \sum_{\mathcal{S}_Z} G_i^Z, \tag{3.1}$$

errors will cause excitations in the energy spectrum of the Hamiltonian, characterized by the anti-commutation of the error with the syndromes. Boundaries are formed by cutting the weight of the stabilizers to be 3 at the boundary, of a given type. The boundaries will be characterized as rough and smooth, depending on the type of excitation that can terminate at the boundary. X-type excitations terminate at the rough boundary, while Z-type excitations terminate at the smooth boundary. Logical operators are formed by



(a) (b)

Figure 3.2: 2D hexagonal color code of distance $d = 5$. Two instances of equivalent logical operators are presented. Logical operators are formed by connecting string-like excitations to all three colored boundaries, fusing within the bulk at the white qubit. In the case of subfigure (a), the logical operator lives on the purple boundary and is trivially connected to that boundary (fusion could be defined at any point in the chain). (b) Logical operators are formed by connecting the three boundary points in the bulk.

creating non-local excitations that connect two boundaries of the same type. As per their definition, logical operators will commute with all the stabilizers of the Hamiltonian, but will cause a change in the energy spectrum of the logical qubit. Since the logical X and Z operators need to cross the surface to connect their respective boundaries, they will meet at an odd number of qubits, therefore satisfying the necessary anti-commutation relationship. Note the parameters of the code, in general for a $d \times d$ lattice, the logical Pauli operators will be of weight-d, the length of one of the boundaries. Therefore, the surface code is a $[[cd^2, 1, d]]$ code, where c is a constant.

3.1.2 2D color code

The color codes are another family of topological error correcting codes, where logical qubits are encoded into systems with geometrically local stabilizers and non-local degrees of freedom corresponding to the logical information [25, 26]. Just as the case with the surface code, the color codes can be defined in a planar geometry.

The stabilizers of the 2D color code are again of X and Z type, resulting in a CSS code. However, unlike in the case of the surface code, all of the stabilizers are now plaquette

operators and each plaquette contains both an X and Z stabilizer. Due to this constraint, all of the plaquettes must meet at an even number of sites, with qubits now residing on the vertices of the lattice. The 2D color code is defined on a lattice where each vertex is of degree 3, and the plaquettes are 3-colorable. Boundaries of the color code are identified with one of the colors, where excitations of a given colored set of plaquettes can terminate at the boundary. Logical operators correspond to excitations that connect all three boundary types, fusing at a given vertex in the lattice.

The 2D color code offers some advantages over the surface code at the expense of slightly higher weights in the stabilizer generators. Since it is a CSS code, the CNOT gate is naturally transversal for the 2D color code. However, due to the symmetry in the X and Z stabilizers of the code, the logical Hadamard is transversal for this code. Moreover, even though the stabilizers are not always of weight 0 mod 4, the phase gate can be implemented transversally by applying $S^\dagger$ to a subset of the qubits [25, 93]. As for the 7-qubit Steane code, the 2D color code can implement all of the Clifford group logical operators transversally. In fact, the 7-qubit Steane code actually corresponds to the smallest distance non-trivial 2D color code. As in the case of the surface code, the parameters of the code will scale as $[[c'd^2, 1, d]]$, where c' is a constant. Roughly, c' will be half the size of the constant c for the surface code, that is an encoded color code qubit requires roughly half that of a surface code qubit [97].

3.1.3 3D color code

The 3D color code is an encoding of a logical qubit in a code whose stabilizers are of finite weight and local in a 3D geometry. Therefore, unlike the case of the 2D color code, the 3D code cannot be projected into a 2D geometry without introducing some form of non-local gate. Moreover, the lattice is defined to be 4-colorable, where stabilizers are defined on 3D cells, and 2D faces. There is some freedom in how one defines the stablizers, but typically the X stabilizers will be defined on 3D cells, and the Z stabilizers will be defined on the intersections of these 3D cells, forming small 2D faces. Therefore, while still forming a CSS code, the symmetry of the X and Z stabilizers is broken, and as such the code will no longer exhibit a transversal logical Hadamard operator.

However, given the 3-dimensional nature of the code, it is possible to find a combination of X stabilizers such that the codewords satisfy the triorthogonality conditions required for the ability to implement a transversal T gate [31]. A code is said to be triorthogonal if

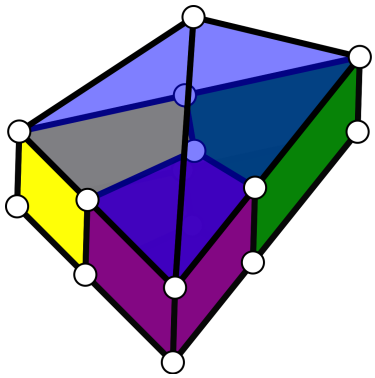


Figure 3.3: Instance of the $[[15, 1, 3]]$ Reed-Muller code as a 3D color code.

the following hold for the X generators of the code:

$$|G_i^X \cdot G_j^X| = 0 \mod 2, \ \forall \ i, j \tag{3.2}$$

$$|G_i^X \cdot G_j^X \cdot G_k^X| = 0 \mod 2, \ \forall \ i, j, k. \tag{3.3}$$

Moreover, an error correcting code has a transversal T gate if and only if the code is triorthogonal. Therefore just as the 2D color codes provide a generalization of the 7-qubit Steane code for varying distances, the 3D color codes provide a generalization of the 15-qubit Reed-Muller code at higher distances. There seems to be an interesting connection between the dimensionality of a topological code and the class of gates that can be implemented transversally, as eluded to by the notion that a 3rd dimension is required in order to generate a triorthogonal code. This notion was made concrete by Bravyi and König in the following Theorem.

Theorem 52 ([33]). *Given a D-dimensional topological stabilizer code encoding a logical qubit in a subspace. Any local, finite-depth quantum circuit implementing the logical gate G_L, implies that $G_L \in \mathcal{C}_n^{(D)}$, the D-th level of the Clifford hierarchy.*

The Bravyi-König result implies that in effect, the 2D color code is the best one could hope for in 2D in terms of logical gates as it can implement all Clifford gates. Furthermore, implementing 3D color codes, or other potential 3D architectures, would pose significant experimental challenges. Moreover, all universal fault-tolerant schemes that circumvent the use of magic states, based on code concatenation or gauge-fixing would require the use of a 3D architecture for the underlying code.

3.2 Code conversion for topological stabilizer codes

The research presented in this section is based on the manuscript from Ref. [75], copyrighted by the American Physical Society. Both authors contributed to the development of the framework of the project and the research ideas within. I did the majority of the writing of the manuscript with editing being done equally by both authors.

3.2.1 Overview

In this paper, we present a method for fault-tolerantly performing a universal set of quantum logic gates within a 2D architecture. Our method translates between error correcting codes—a 2D color code, and a special class of 3D color codes—to allow for the transversal application of different sets of logical gates. Specifically, we present a mapping from 2D color codes to a 3D code, which we call the *stacked code*, by pairing multiple copies of the 2D color code, generalizing the work of Ref. [6]. Multiple 2D color codes can be pairwise stacked to increase the overall distance of the newly created stacked code to equal the distance of the 2D color code. We show that the stacked code admits a transversal $\pi/8$ logic gate, and that the transformation from the 2D color code to the stacked code and back can be performed fault-tolerantly. Furthermore, by unfolding the stacked code and tiling the original 2D color codes in a 2D layout, this code maintains its properties. The transformation from 2D color code to stacked code in this 2D layout can be performed with a sequence of local gauge measurements in the bulk of the 2D color codes and Bell pairing measurements along the boundary of neighboring 2D codes. In order to not violate the Bravyi–König no-go theorem, the measurements pairing the different 2D color codes are necessarily nonlocal, but in a very limited way. Specifically, these measurements can be performed along one-dimensional (1D) strips forming the boundary between neighboring 2D codes in a 2D arrangement.

A recent result by Bravyi and Cross [30] presents a very similar construction to the one we present here. Specifically, they detail a fault-tolerant 2D construction for universal quantum computation that relies on the same type of pairing of 2D color codes (which they call doubled color codes) and measurements between the different layers of color codes, as we propose, to implement a gate outside the Clifford group. Importantly, the results of Ref. [30] go beyond our construction to show how to reduce the nonlocal joint logical Pauli operators that have to be measured into a sequence of local measurements by proposing a "subdivision gadget." They further supplement their findings by proposing a decoding method to address for the correlated noise that is introduced by the action of the non-

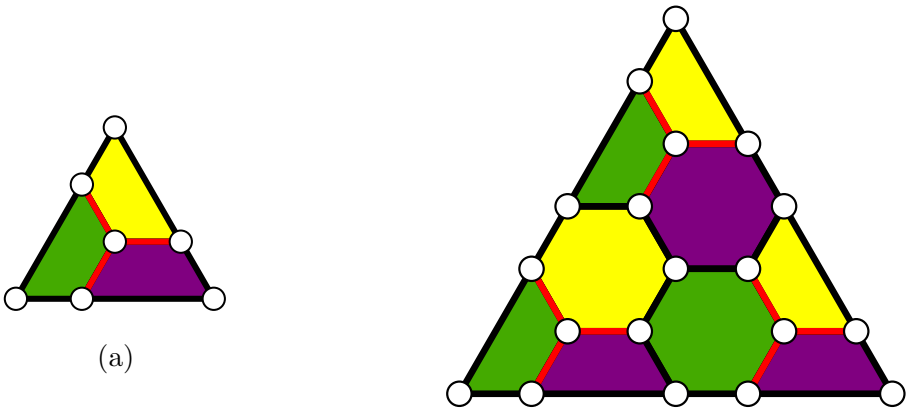
Clifford $\pi/8$ gate. In addition, Jones, Brooks, and Harrington recently proposed a method to implement a similar form of construction for the [4.8.8] color code [81], as opposed to the hexagonal color code studied here and in Ref. [30]. In their construction, they propose a method for measuring the set of nonlocal joint logical Pauli operators through a series of local measurements inspired by lattice surgery methods [72, 97]. Our results complement those of Refs. [30, 81] by providing an explicit presentation of the properties of this 2D structure as a type of 3D color code with stabilizers that can be inferred by measurements only of local 2D stabilizers and gauge operators, as well as weight-$O(d)$ 1D operators on the boundaries of 2D codes.

3.2.2 Transforming to the stacked code

In this section, we describe a transformation to map the logical qubit encoded in a 2D color code into a particular form of 3D color code, which we call a *stacked code*. This stacked code will allow for the transversal implementation of a logical $\pi/8$ gate (defined by $\mathrm{diag}[e^{-i\pi/8}, e^{i\pi/8}]$), which together with the transversal logical gates in the 2D color code form a universal gate set. We introduce this transformation by generalizing the technique of Anderson *et al.* [6], which mapped a seven-qubit Steane code (also a $d = 3$ 2D color code) to a 15-qubit quantum Reed-Muller code (also a $d = 3$ 3D color code). Our generalization applies to hexagonal color codes of any distance, and gives rise to a 3D color code of distance $d = 3$. We then show to further generalize this transformation to yield a stacked code with arbitrary distance d.

3.2.2.1 Transforming 2D color codes to 3D: distance 3 protection

Consider a $[[n, 1, d]]$ hexagonal color code family [26], with $n = (3d^2 + 1)/4$, defined by X and Z stabilizer generators expressed as plaquette operators $G_{P_i} = \otimes_{\nu \in P_i} X_\nu$ and $H_{P_i} = \otimes_{\nu \in P_i} Z_\nu$, where the tensor product is over vertices ν defining a hexagonal plaquette P_i, with appropriate modification at the boundaries. Our construction will use multiple copies of such codes with stabilizer generators $\{G_{P_i}^{(l)}\}$ and $\{H_{P_i}^{(l)}\}$, where l is a label for the particular copy of the 2D color code. For any such code, one can identify a set of weight-2 Z-type edge operators $\{H_{e_i}^{(l)}\}$, see Fig. 3.4, that will, along with the Z-type plaquette operators, generate any Z-type edge in the 2D lattice. We label these edges by e_i, as they can be identified in a one-to-one correspondence with plaquette operators labeled by P_i. Given such a generating set $\{H_{e_i}^{(l)}\}$, one can identify each X plaquette generator $G_{P_i}^{(l)}$ with a particular Z edge operator $H_{e_i}^{(l)}$ such that this pair of operators anti-commute, as they will intersect at only one site.



(a) (b)

Figure 3.4: Two instances of the 2D hexagonal color code of distance (a) $d = 3$ and (b) $d = 5$. In each case, a set of independent edges $\{H_{e_i}\}$, shown in red, can be chosen as the set that will form the Z gauge operators when paired with the identical edge from another code copy, thus forming weight-4 gauge operators $\{H_{e_i}^{(2k-1)} H_{e_i}^{(2k)}\}$.

Consider a logical qubit encoded in a 2D hexagonal color code labeled $l = 1$ of distance d, with stabilizer generators $\{G_{P_i}^{(1)}\}$ and $\{H_{P_i}^{(1)}\}$. We now consider a process by which we transform this 2D code into a 3D code, following the method of Anderson *et al.* [6]. Our transformation makes use of a second 2D color code of equivalent size to the first, with its encoded logical qubit entangled in a Bell state with a single ancilla qubit. That is, denoting the logical operators for the second color code labeled $l = 2$ by $X_L^{(2)}$ and $Z_L^{(2)}$, and the operators for a single ancilla qubit by X and Z, this Bell state is stabilized by $X_L^{(2)} \otimes X$ and $Z_L^{(2)} \otimes Z$ as well as the code stabilizers $\{G_{P_i}^{(2)}\}$ and $\{H_{P_i}^{(2)}\}$.

We induce the transformation through joint measurement of gauge operators of the two color codes. Specifically, we measure the Z-type gauge operators between the two codes corresponding to pairing up the generating Z-type edge operators of the two codes and jointly measuring the corresponding weight-4 operators $\{H_{e_i}^{(1)} \otimes H_{e_i}^{(2)}\}$. Because each of the original X-type plaquette operators of the two codes $G_{P_i}^{(l)}$ anti-commute with the measured gauge operator $H_{e_i}^{(1)} \otimes H_{e_i}^{(2)}$, they will no longer be stabilizers of the code. However, the joint volume operator $G_{P_i}^{(1)} \otimes G_{P_i}^{(2)}$ obtained by pairing corresponding plaquette operators between the two code copies will remain a stabilizer as it has even overlap with the gauge operator $H_{e_i}^{(1)} \otimes H_{e_i}^{(2)}$. As a result of these measurements, the evolution of the stabilizers for the entire system is given by:

2D code + ancilla Bell	3D code	
$\{G_{P_i}^{(1)}\} \otimes I^{\otimes n} \otimes I$	$\{H_{e_i}^{(1)} \otimes H_{e_i}^{(2)}\} \otimes I$	(3.4)
$\{H_{P_i}^{(1)}\} \otimes I^{\otimes n} \otimes I$	$\{H_{P_i}^{(1)}\} \otimes I^{\otimes n} \otimes I$	(3.5)
$\{G_{P_i}^{(1)} \otimes G_{P_i}^{(2)}\} \otimes I$	$\{G_{P_i}^{(1)} \otimes G_{P_i}^{(2)}\} \otimes I$	(3.6)
$\{H_{P_i}^{(1)} \otimes H_{P_i}^{(2)}\} \otimes I$	$\{H_{P_i}^{(1)} \otimes H_{P_i}^{(2)}\} \otimes I$	(3.7)
$I^{\otimes n} \otimes X_L^{(2)} \otimes X$	$I^{\otimes n} \otimes X_L^{(2)} \otimes X$	(3.8)
$I^{\otimes n} \otimes Z_L^{(2)} \otimes Z$	$I^{\otimes n} \otimes Z_L^{(2)} \otimes Z$	(3.9)

where the last two stabilizers represent those corresponding to the second code copy being prepared in a Bell pair with an ancilla qubit. We note that choosing the smallest nontrivial 2D color code, corresponding to $n = 7$ and $d = 3$ and equivalent to the seven-qubit Steane code, this mapping corresponds to that of Anderson *et al.* [6] in this case. Even though in general the 2D codes used in this construction are of distance d, the overall distance of transformed code is limited to be 3. Logical Z string operators are formed by matching pairs of qubits from the two copies of the 2D codes along with the single ancilla qubit, and take the form $Z_i^{(1)} Z_i^{(2)} Z$. A higher weight logical Z operator can be obtained by traversing the 2D color code layers and connecting error strings of different colors. We shall expand upon this point for the general case in Sec. 3.2.3.2.

This new code is a 3D color code, where the 3D code stabilizers of Eqs. (3.8)–(3.9) correspond to the stabilizers of the fourth color and the boundary of the new color corresponds to the original 2D code. We prove that it is a 3D color code, and determine its distance in the general case, in Sec. 3.2.3.2. The code possesses a transversal $\pi/8$ gate, as proven below, in a similar manner to the techniques proposed in Refs. [25, 31, 93], and will therefore form a universal fault-tolerant gate set along with the logical Clifford gates that can be applied transversally to the original 2D code.[1]

However, this code has a number of undesirable features from the perspective of topological stabilizer codes. First, we note that the stabilizers in Eqs. (3.8)-(3.9) are very high weight, having support on the entire set of qubits across a full 2D layer. We postpone discussion about how one might infer the values of these high-weight stabilizers using only lower-weight measurements to Sec. 3.2.3.3. Second, the distance of this 3D code is limited

[1]The transversal gates are not strictly transversal, that is all the same rotation, for the hexagonal color code. However, by applying the inverse rotation to the appropriate set of qubits the correct logical operator can be applied [23, 93].

by the width of the third dimension (two layers + one ancilla qubit). This limitation is in line with the intuition behind the no-go result of Bravyi and König [33], where it is shown that a topological stabilizer code must be at least dimension 3 or higher to possess a transversal gate operation that lies outside the Clifford group. One might suspect that the fault-tolerance protection that one should get from the distance of the code should be related to the depth of the third dimension of the code.

3.2.3 Proof of transversal logical $\pi/8$ gate for the stacked code

Consider a $[[n, 1, d]]$ qubit 2D color code whose X and Z generators are labeled by $\{G_i^{(1)}\}$ and $\{H_i^{(1)}\}$, respectively, that encodes a single logical qubit. Consider the following basis for the 2D code based off the CSS code construction (up to state normalization):

$$|0_{2D}\rangle = \prod_i (I + G_i^{(1)})|0\rangle^{\otimes n} = \sum_{\boldsymbol{g_x}} |\boldsymbol{g_x}\rangle,$$
$$|1_{2D}\rangle = X_L^{(1)} \prod_i (I + G_i^{(1)})|0\rangle^{\otimes n} = \sum_{\boldsymbol{g_x}} |\overline{\boldsymbol{g_x}}\rangle,$$

where $X_L^{(1)} = X^{\otimes n}$ is the X logical operator for the code and $\boldsymbol{g_x}$ is an n-bit binary vector that lies in the set of vectors generated by the operators $\{G_i\}$, and $\overline{\boldsymbol{g_x}} = \boldsymbol{g_x} \oplus (1, \ldots, 1)$.

Introduce a $(n+1)$-qubit ancillary system in the following state:

$$\frac{1}{\sqrt{2}}(|0_{2D}\rangle|0\rangle + |1_{2D}\rangle|1\rangle). \tag{3.10}$$

The stabilizer generators of the original encoded state and the ancillary state thus correspond to:

2D code + ancilla Bell state stabilizers	Equivalent stabilizers	
$\{G_{P_i}^{(1)}\} \otimes I^{\otimes n} \otimes I$	$\{G_{P_i}^{(1)}\} \otimes I^{\otimes n} \otimes I$	(3.11)
$\{H_{P_i}^{(1)}\} \otimes I^{\otimes n} \otimes I$	$\{H_{P_i}^{(1)}\} \otimes I^{\otimes n} \otimes I$	(3.12)
$I^{\otimes n} \otimes G_{P_i}^{(2)} \otimes I$	$\{G_{P_i}^{(1)} \otimes G_{P_i}^{(2)}\} \otimes I$	(3.13)
$I^{\otimes n} \otimes H_{P_i}^{(2)} \otimes I$	$\{H_{P_i}^{(1)} \otimes H_{P_i}^{(2)}\} \otimes I$	(3.14)
$I^{\otimes n} \otimes X_L^{(2)} \otimes X$	$I^{\otimes n} \otimes X_L^{(2)} \otimes X$	(3.15)
$I^{\otimes n} \otimes Z_L^{(2)} \otimes Z$	$I^{\otimes n} \otimes Z_L^{(2)} \otimes Z$	(3.16)

where the right stabilizer generators are equivalent to those on the left by multiplying lines 3 and 4 on the left by lines 1 and 2, respectively (using the notation $\{G_i^{(1)} \otimes G_i^{(2)}\}$ to signify that we are multiplying the corresponding i^{th} stabilizer of each code with one another). Then by following a procedure similar to that proposed by Anderson *et al.* [6], one can replace the X generators from the first line on the right by measuring appropriate gauge Z operators. to form a new $(2n+1)$-qubit code. The code remains a valid CSS code as the stabilizers all commute and satisfy the requirements of $C_2 \subset C_1$, where C_1 is the classical code whose parity check matrix is given by the X stabilizers $\{G_i^{(1)} \otimes G_i^{(2)}\}$ and C_2 is the classical code whose parity check matrix is obtained from the Z stabilizers.

We proceed to show we can implement a logical gate from $\mathcal{C}_3$ transversally. We define an individual Z-axis rotation as follows: $Z(\theta) = \text{diag}[1, e^{i\pi\theta}]$. Suppose that the 2D color code is chosen such that $U_T = \otimes_{i=1}^{n} Z(\theta_i) = Z(\boldsymbol{\theta})$ implements a logical phase gate $S_L = \text{diag}[1, i] \in \mathcal{C}_2$ (the vector $\boldsymbol{\theta}$ represents the individual rotations about the Z axis on the physical qubits forming the quantum code). Note the following observation:

$$\begin{aligned}
U_T|0_{2D}\rangle &= Z(\boldsymbol{\theta}) \sum_{\boldsymbol{g_x}} |\boldsymbol{g_x}\rangle = \sum_{\boldsymbol{g_x}} e^{i\pi\boldsymbol{\theta}\cdot\boldsymbol{g_x}} |\boldsymbol{g_x}\rangle = \sum_{\boldsymbol{g_x}} |\boldsymbol{g_x}\rangle \\
&\Longrightarrow e^{i\pi\boldsymbol{\theta}\cdot\boldsymbol{g_x}} = 1, \ \forall\ \boldsymbol{g_x} \\
&\Longrightarrow \boldsymbol{\theta}\cdot\boldsymbol{g_x} = 0 \mod 2, \ \forall\ \boldsymbol{g_x}, \qquad (3.17) \\
U_T|1_{2D}\rangle &= Z(\boldsymbol{\theta}) \sum_{\boldsymbol{g_x}} |\overline{\boldsymbol{g_x}}\rangle = \sum_{\boldsymbol{g_x}} e^{i\pi\boldsymbol{\theta}\cdot\overline{\boldsymbol{g_x}}} |\overline{\boldsymbol{g_x}}\rangle = e^{i\pi/2} \sum_{\boldsymbol{g_x}} |\overline{\boldsymbol{g_x}}\rangle \\
&\Longrightarrow e^{i\pi\boldsymbol{\theta}\cdot\overline{\boldsymbol{g_x}}} = e^{i\pi/2}, \ \forall\ \boldsymbol{g_x} \\
&\Longrightarrow \boldsymbol{\theta}\cdot\overline{\boldsymbol{g_x}} = \frac{1}{2} \mod 2, \ \forall\ \boldsymbol{g_x}. \qquad (3.18)
\end{aligned}$$

The assumption that the transversal gate U_T implements a logical phase gate translates into conditions on the individual physical rotations $\boldsymbol{\theta}$ coupled to the form of the binary vectors $\boldsymbol{g_x}$ related to the X generators of the 2D quantum code. Consider the CSS code proposed in Sec. 3.2.2, where the X generators are given by,

$$\begin{aligned}
&\{G_i^{(1)} \otimes G_i^{(2)}\} \otimes I \\
&I^{\otimes n} \otimes X_L^{(2)} \otimes X,
\end{aligned}$$

then a particular choice of code states can be obtained by the CSS code construction as

(upto state normalization):

$$|0_{3D}\rangle = (I^{\otimes(2n+1)} + I^{\otimes n} \otimes X_L^{(2)} \otimes X) \times \prod_i (I + G_i^{(1)} \otimes G_i^{(2)} \otimes I)|0\rangle^{\otimes(2n+1)} \tag{3.19}$$

$$= (I^{\otimes(2n+1)} + I^{\otimes n} \otimes X_L^{(2)} \otimes X) \sum_{\boldsymbol{g_x}} |\boldsymbol{g_x}\rangle|\boldsymbol{g_x}\rangle|0\rangle$$

$$= \sum_{\boldsymbol{g_x}} \Big(|\boldsymbol{g_x}\rangle|\boldsymbol{g_x}\rangle|0\rangle + |\boldsymbol{g_x}\rangle|\overline{\boldsymbol{g_x}}\rangle|1\rangle \Big), \tag{3.20}$$

$$|1_{3D}\rangle = (X_L^{(1)} \otimes X_L^{(2)} \otimes X)|0_{2D}\rangle$$

$$= \sum_{\boldsymbol{g_x}} \Big(|\overline{\boldsymbol{g_x}}\rangle|\boldsymbol{g_x}\rangle|0\rangle + |\overline{\boldsymbol{g_x}}\rangle|\overline{\boldsymbol{g_x}}\rangle|1\rangle \Big). \tag{3.21}$$

Claim 53. *The $(2n+1)$ qubit transversal gate $V_T = Z(\frac{\boldsymbol{\theta}}{2}) \otimes Z(\frac{\boldsymbol{\theta}}{2}) \otimes Z(\alpha)$, where α is chosen such that $\alpha \in \{1/4, 5/4\}$, implements a logical T or TZ gate in the logical computational basis $\{|0_{3D}\rangle, |1_{3D}\rangle\}$, where $T = \pi/8$ gate.*

Proof. For the purpose of this proof, we consider the case where the $\pi/8$ gate has the form $T = \mathrm{diag}[1, e^{i\pi/4}]$, which is equivalent to $\mathrm{diag}[e^{-i\pi/8}, e^{i\pi/8}]$ up to a global phase. Consider first the action of V_T upon the state $|0_{3D}\rangle$ which should return $|0_{3D}\rangle$ without a phase.

$$V_T|0_{3D}\rangle = \sum_{\boldsymbol{g_x}} \Big(e^{i\pi\frac{\boldsymbol{\theta}}{2}\cdot\boldsymbol{g_x}} e^{i\pi\frac{\boldsymbol{\theta}}{2}\cdot\boldsymbol{g_x}} |\boldsymbol{g_x}\rangle|\boldsymbol{g_x}\rangle|0\rangle + e^{i\pi\frac{\boldsymbol{\theta}}{2}\cdot\boldsymbol{g_x}} e^{i\pi\frac{\boldsymbol{\theta}}{2}\cdot\overline{\boldsymbol{g_x}}} e^{-i\pi\alpha} |\boldsymbol{g_x}\rangle|\overline{\boldsymbol{g_x}}\rangle|1\rangle \Big) \tag{3.22}$$

$$= \sum_{\boldsymbol{g_x}} \Big(e^{i\pi\boldsymbol{\theta}\cdot\boldsymbol{g_x}} |\boldsymbol{g_x}\rangle|\boldsymbol{g_x}\rangle|0\rangle + e^{i\pi\frac{\boldsymbol{\theta}}{2}\cdot\boldsymbol{g_x}} e^{i\pi\frac{\boldsymbol{\theta}}{2}\cdot\overline{\boldsymbol{g_x}}} e^{-i\pi\alpha} |\boldsymbol{g_x}\rangle|\overline{\boldsymbol{g_x}}\rangle|1\rangle \Big) \tag{3.23}$$

$$= \sum_{\boldsymbol{g_x}} \Big(|\boldsymbol{g_x}\rangle|\boldsymbol{g_x}\rangle|0\rangle + |\boldsymbol{g_x}\rangle|\overline{\boldsymbol{g_x}}\rangle|1\rangle \Big), \tag{3.24}$$

where the first coefficient in (3.23) is equal to 1 by the identity in Eq. 3.17, and the second coefficient is equal to 1 by the following observation. Define the phase a to be the phase $e^{i\pi a} = e^{i\pi\frac{\boldsymbol{\theta}}{2}\cdot\boldsymbol{g_x}} e^{i\pi\frac{\boldsymbol{\theta}}{2}\cdot\overline{\boldsymbol{g_x}}}$. Due to the symmetries of color codes, the value of a in the following

is independent of $\boldsymbol{g_x}$:

$$\begin{aligned}
&\frac{\boldsymbol{\theta}}{2}\cdot\boldsymbol{g_x}+\frac{\boldsymbol{\theta}}{2}\cdot\overline{\boldsymbol{g_x}}=a \mod 2\\
\implies &\boldsymbol{\theta}\cdot\boldsymbol{g_x}+\boldsymbol{\theta}\cdot\overline{\boldsymbol{g_x}}=2a \mod 2\\
\implies &0+\frac{1}{2}=2a \mod 2\\
\implies &a=\{\frac{1}{4},\frac{5}{4}\} \mod 2,
\end{aligned} \tag{3.25}$$

therefore α is chosen in order to set the coefficient equal to 1. Consider now the action of V_T, with the appropriate choice of α for the state $|1_{3D}\rangle$, which should return the state $\pm e^{i\pi/4}|1_{3D}\rangle$.

$$\begin{aligned}
V_T|1_{3D}\rangle=\sum_{\boldsymbol{g_x}}\Big(&e^{i\pi\frac{\boldsymbol{\theta}}{2}\cdot\overline{\boldsymbol{g_x}}}e^{i\pi\frac{\boldsymbol{\theta}}{2}\cdot\boldsymbol{g_x}}|\overline{\boldsymbol{g_x}}\rangle|\boldsymbol{g_x}\rangle|0\rangle\\
&+e^{i\pi\frac{\boldsymbol{\theta}}{2}\cdot\overline{\boldsymbol{g_x}}}e^{i\pi\frac{\boldsymbol{\theta}}{2}\cdot\overline{\boldsymbol{g_x}}}e^{-i\pi\alpha}|\overline{\boldsymbol{g_x}}\rangle|\overline{\boldsymbol{g_x}}\rangle|1\rangle\Big),
\end{aligned} \tag{3.26}$$

which given a choice of α gives the following:

$$\begin{aligned}
V_T|1_{3D}\rangle&=\sum_{\boldsymbol{g_x}}\Big(e^{i\pi\alpha}|\overline{\boldsymbol{g_x}}\rangle|\boldsymbol{g_x}\rangle|0\rangle+e^{i\pi(\frac{1}{2}-\alpha)}|\overline{\boldsymbol{g_x}}\rangle|\overline{\boldsymbol{g_x}}\rangle|1\rangle\Big),\\
&=e^{i\pi\alpha}|1_{3D}\rangle=\pm e^{i\pi/4}|1_{3D}\rangle,
\end{aligned} \tag{3.27}$$

since $\alpha=\dfrac{1}{2}-\alpha \mod 2$. □

Therefore we can apply a transversal $\pi/8$ gate to the code construction given above by applying a transversal logical Z gate at the completion of our gate (the action of T or TZ is fixed by the code and is not probabilistic).

Corollary 54. *The stacked code has a transversal logical $\pi/8$ gate.*

Proof. The only assumption the proof of Claim 53 makes about the ancilla state is that the rotation $Z(\alpha)$ induces a phase of $e^{i\pi\alpha}$ on the $|1\rangle$ state and leaves the $|0\rangle$ state invariant. Therefore, we replace the single physical qubit by a logical qubit $\{|0_{3D}\rangle,|1_{3D}\rangle\}$ prepared in a 3D state according to the construction laid out in this appendix. Replacing the single Z of angle α by a transversal rotation as given by the construction of the previous claim, we can recursively build the stacked code to implement an overall transversal rotation of the $\pi/8$ gate for the stacked code. □

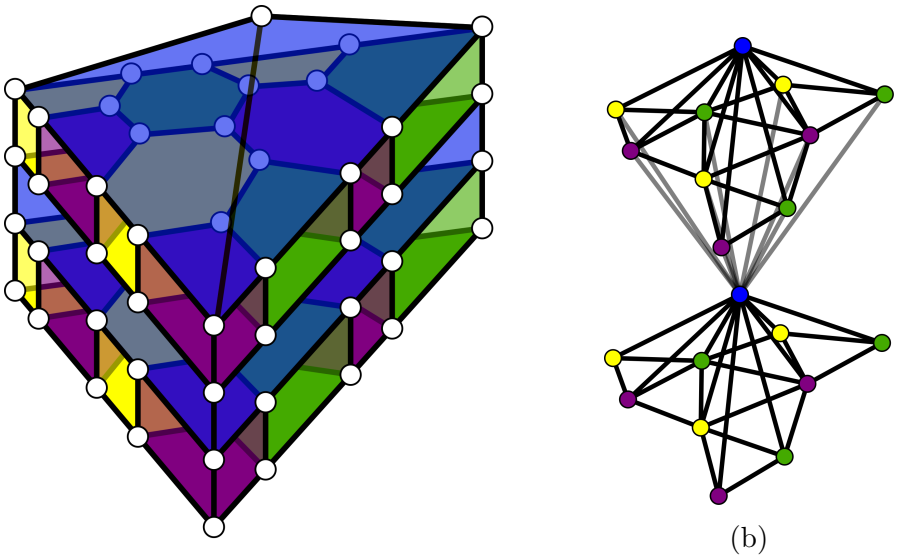
3.2.3.1 Transforming 2D color codes to 3D: distance d protection

To increase the distance of our newly formed code, we must increase the width of its third dimension. A natural method to provide such added protection would be to encode the weakest part of the code, the bare ancilla qubit, into a 3D code of its own using the exact same technique. We can continue this process recursively, by performing the joint stabilizer measurements in (3.8)–(3.9) as joint logical X and Z measurements. The encoded ancilla state will be prepared offline using 2D color codes arranged as layers in a stack, coupled into logical Bell pairs by performing joint logical X and Z measurements, henceforth referred to as *Bell stabilizers*. This bulk ancilla state will allow us to transform our 2D color code into a 3D color code with large distance. In addition, as the individual components forming the bulk ancilla state are restricted to pairs of 2D layers, this will allow us to show in Sec. 3.2.4 that such a process can be made fault-tolerant on a 2D lattice.

Specifically, our recursive transformation from a 2D color code on layer $k = 1$ to a d-layer stack is defined by the following evolution of stabilizers:

2D code + ancilla Bell	3D code	
$\{G_{P_i}^{(2k-1)}\}$	$\{H_{e_i}^{(2k-1)} H_{e_i}^{(2k)}\}$	(3.28)
$\{H_{P_i}^{(2k-1)}\}$	$\{H_{P_i}^{(2k-1)}\}$	(3.29)
$\{G_{P_i}^{(2k-1)} G_{P_i}^{(2k)}\}$	$\{G_{P_i}^{(2k-1)} G_{P_i}^{(2k)}\}$	(3.30)
$\{H_{P_i}^{(2k-1)} H_{P_i}^{(2k)}\}$	$\{H_{P_i}^{(2k-1)} H_{P_i}^{(2k)}\}$	(3.31)
$X_L^{(2k)} X_L^{(2k+1)}$	$X_L^{(2k)} X_L^{(2k+1)}$	(3.32)
$Z_L^{(2k)} Z_L^{(2k+1)}$	$Z_L^{(2k)} Z_L^{(2k+1)}$	(3.33)

where $k \in \{1, \cdots, \frac{d-1}{2}\}$. As the final layer is a single qubit, we have $X_L^{(d)} = X$, and $Z_L^{(d)} = Z$. The logical qubit is initially stored in the first 2D color code layer, stabilized by the operators $\{G_{P_i}^{(1)}\}$ and $\{H_{P_i}^{(1)}\}$. The additional layers are prepared in joint Bell pairs, as indicated by the Bell stabilizers $X_L^{(2k)} X_L^{(2k+1)}$ and $Z_L^{(2k)} Z_L^{(2k+1)}$. The pairs of copies of the 2D sheets are then coupled together by measuring the gauge operators $\{H_{e_i}^{(2k-1)} H_{e_i}^{(2k)}\}$ between one sheet and another sheet from a different pair. This is logically equivalent to stacking the different pairs to form one large stack of height distance d, where each layer is a copy of a 2D color code also with distance d, as shown in Fig. 3.5. We call the resulting 3D code the $(d-1)+1$ *stacked code*. At this point, the Bell stabilizers in Eqns. (3.32)–(3.33) have a cell-like structure connecting the two 2D color code sheets with which they



(a) (b)

Figure 3.5: (a) Graphical representation of the primal lattice of the $(d-1)+1$ stacked code formed by stacking different copies of 2D color codes, shown here for $d = 5$. The copies of the 2D code are coupled either by measuring gauge operators or logical operator pairs (shown in blue) between the different layers. (b) Dual lattice for the 3D stacked code ($d = 5$). Vertices represent cell stabilizers in the primal lattice and edges represent faces shared by connected stabilizers.

are associated. These correspond to the Blue stabilizers in Fig. 3.5 and will have particular features when viewing this code as a 3D color code, as we explore in the next section, as well as several properties needed to make our 2D arrangement of this code in Sec. 3.2.4.1.

3.2.3.2 Properties of the stacked code

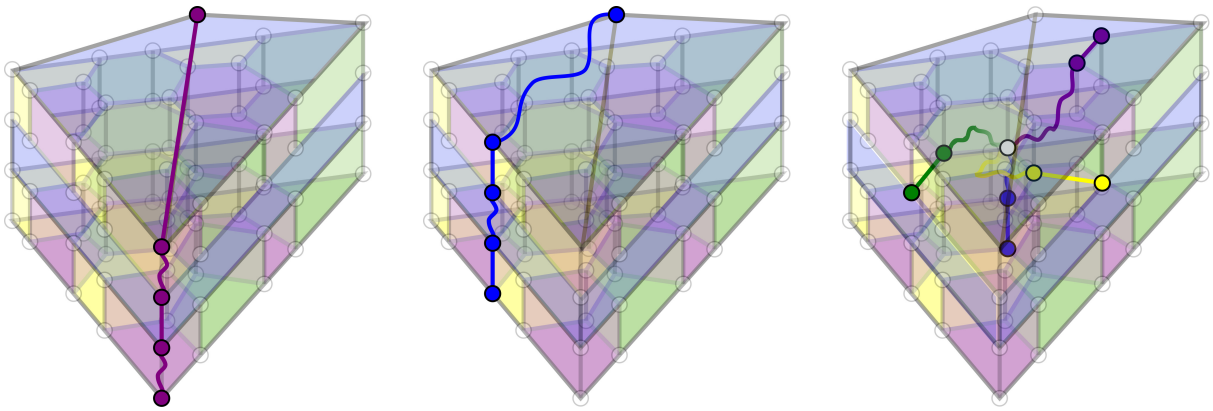
The $(d-1)+1$ stacked code is also a 3D color code. This can most easily be seen using its dual lattice, as follows. Take the dual lattice of the 2D color code, connect each of the vertices of the dual lattice (consisting of 3 colors) to a single vertex of a different color. We shall denote the colors of the original 2D code as green (g), purple (p), and yellow (y) and the color of the newly formed stabilizers in 3D by blue (b). Connect this single vertex to another set of vertices forming a 2D code, and repeat this process $(d-1)/2$ times. Each of the vertices in the dual lattice form a 3D stabilizer cell in the primal lattice, where edges

between the vertices in the dual lattice are equivalent to faces at the intersection of cells in the primal lattice, see Fig. 3.5 for an example of the dual lattice. It is straightforward to see that this construction is equivalent to the construction outlined for the stacked code, and moreover, because the dual lattice is four-colorable and composed of tetrahedra, it is a valid 3D color code [25, 93].

We now proceed to determine the distance of the $(d-1)+1$ stacked code, making use of the well-studied properties of the 3D color code. The edges in the primal lattice of a color code can be identified with one of the colors of the code [23, 93]. In the case of a 3D color code, the faces at the intersection of two tetrahedra in the dual lattice correspond to edges in the primal lattice, where the color of the edge in the primal lattice is given by the complementary color to the vertices forming the face in the dual lattice. A boundary of a given color is the set of points at which edges of that given color terminate without a stabilizer of the given color being present. In the case of the stacked code, the three original colors of the 2D lattice form boundaries along the three sides of the stack extending upwards from their original 1D boundary given by the 2D color code. The fourth boundary, for the newly introduced color in three dimensions (blue), is located along the bottom boundary of the 3D lattice, as none of these qubits touch a blue stabilizer.

The Bell stabilizers given in Eqs. (3.32)–(3.33) correspond to the blue stabilizers in Fig. 3.5, and are equivalent to measuring the joint logical X and Z operators of the two 2D color codes forming the top and bottom faces of the blue stabilizer. As opposed to traditional constructions of 3D color codes, the Blue stabilizers are not of low weight, but rather act on $\mathcal{O}(d^2)$ qubits. This is a particular feature of the stacked code structure, as the Blue stabilizers measure joint logical operators across pairs of 2D sheets and thus must contain all qubits across those faces. However, as we show in Sec. 3.2.3.3, these high weight stabilizers across the full 2D sheets need not be measured in practice.

Logical operators in any color code are given by string operators that connect the boundaries of different colors [25]. A c-colored string operator is given by a set of qubits formed of connected edges of color c (two edges are connected if they share a stabilizer of color c). A c-colored string operator either has endpoints at the boundary of color c, in which case the final edge of this string connects the endpoint to the boundary, or in the bulk where the endpoint is located at a particular c-colored stabilizer, thus causing an excitation. If all of the colored strings meet at a given qubit, then the strings can "fuse" and the bulk excitation formed by this endpoint will be negated [25, 26]. Therefore, in order to obtain a logical string operator, all colored string operators must connect their respective boundary to a shared fusion point, leading to a nontrivial string connecting boundaries of all colors without excitations. These properties now allow us to prove the distance of the stacked code.



(a) (b) (c)

Figure 3.6: Examples of the different representations of equivalent logical error strings that exist in the 3D stacked code. The color of the logical strings are chosen according to the color of the edges they follow. The curved lines represent joining of edges through a stabilizer of the same color. In (a), because the string lies on the green–yellow boundary, it can be chosen to be either of the complementary colors, blue or purple. In (b), the error string connects the bottom blue boundary to the joint boundary of the other three colors at the ancilla qubit, following blue edges. Example (c) shows how multiple colored boundaries can fuse in the bulk, thus negating the excitation that would otherwise be present.

Lemma 55. *A $(d-1)+1$ stacked code is a 3D color code whose distance is d.*

Proof. The stacked code comprises pairs of 2D layers separated by large blue (b) stabilizers. We shall consider two different representations of logical Z operators, one where the logical operator is composed of qubits that are only in a single pair of these 2D layers, and one where the logical operators span multiple pairs of 2D layers. In the first case, the only way for such a logical operator to connect to the b boundary would be for it to be in the bottom-most pair of 2D layers, as they themselves are trivially connected to the b boundary. However, because we are focusing on a single pair of 2D layers, we can map the problem of finding a logical operator to that of finding one in a single 2D layer, where connecting edges correspond to one of the pair of edges connecting two stabilizer cells of the same color (these edges correspond to the original edges of the individual 2D codes). If the same edge in both color code copies is part of the error chain, then these two edges cancel out as the resulting face corresponds to a gauge Z operator. Therefore, we can refer back to individual edges connecting stabilizers in the 2D color code. As such, because the 2D code is a code of distance d, the smallest-weight logical string that connects the different colored boundaries must be weight d, and therefore any such logical operator will be of distance d.

Suppose we are given a set of Z errors forming a string operator of one of the colors of the original 2D code. Without loss of generality, let this string be of color g. Now, given that a string operator formed by a set of edges of color g, the only way for a string operator of color g to connect qubits from different pairings of 2D layers (that is, traverse a blue stabilizer) is by using g-colored edges at the corner of a given layer. These points lie at the joint boundary of p and y by definition. There are then two methods for such an error string to connect to the g boundary, either by traversing through a given 2D pair to the g boundary of the other side, or by connecting up to the single ancilla qubit that is at the intersection of the g, p, and y boundaries. In the case of the former, in order for a logical string to connect across a given pair of 2D layers to the boundary on the other side, the minimal weight will be governed by the distance of the individual 2D codes, as we previously saw. Therefore, the minimal weight of such a logical operator will be d. In the case when the error string connects to the single ancilla qubit, then in order to form a logical operator it must also connect to the b boundary, as shown in Fig. 3.6a. The single ancilla qubit is as far away from the b boundary as it can be, and in order to create a logical string that connects to the bottom boundary through g edges, there will have to be at least a single qubit per 2D layer connecting to the ancilla qubit. Therefore, the minimal distance of such an operator will also be d. Finally, we must consider the case where the logical string is not composed of strings of colors $\{g, p, y\}$ (the original colors of the 2D code). In such a case, the string operator must terminate at the joint boundary of the three colors,

again given by the single ancilla qubit, and as in the previous case must connect the single qubit to the bottom b boundary through a b-colored chain, as shown in Fig. 3.6b. Such an operator will be of weight at least d as argued above. As such, the minimum weight non-trivial Z operator is of weight d. Since logical X operators are formed by connecting 2D membranes in the 3D code [25], the X logical distance will be greater than that of the Z logical operators, and as such the distance of the code is d. □

We note a potential efficiency that may be gained in the number of qubits in the stacked code. Because the distance to the blue boundary (the bottom layer) of each pair of 2D code sheets increases by 2 for each separation by a blue stabilizer, as shown in Fig. 3.5, we can in principle use pairs of 2D color codes of decreasing distance according to how far away they are from the blue boundary, i.e., decreasing with k. Although we do not prove this result here, the intuition behind this idea is as follows. Because a logical error must connect to the blue boundary, there is extra protection for any logical error that wants to span a given pair of 2D sheets as the error string will have to traverse all layers below the pair of layers. The stacked code prepared in such a way would resemble more of a pyramid than a prism. This method of stacked code construction leads to an analogous code as presented by Bravyi and Cross, based on differing sizes of doubled color codes [30].

3.2.3.3 Fault-tolerant implementations of a universal gate set

Consider a qubit encoded into a 2D hexagonal color code. By the properties of this code, logical Hadamard H and Phase S are transversal, and a logical CNOT between two such codes is also transversal. These are all logical Clifford gates, and so we require an additional gate such as the logical $\pi/8$ gate to complete a universal set. As we now show, transforming to the 3D stacked code can be used as a means to complete a universal gate set, just as gauge fixing provides a means for dimensional jumps in gauge color codes [22, 23].

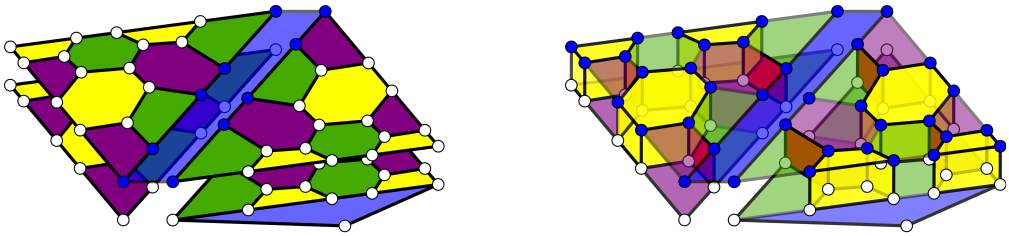
The initial ancillary 2D layers can be prepared in their appropriate Bell pairs offline. Because these states are stabilizer states, they can be prepared fault-tolerantly. In order to preserve the fault-tolerance property of the high-weight Bell stabilizers measurements, a cat state of the same number of qubits as the weight can be prepared fault-tolerantly offline [126, 51]. The measurement of these high-weight stabilizers is repeated in order to ensure fault-tolerance [2]. Note that this preparation process can be combined with the final measurement process outlined below, and therefore will not contribute to the overall runtime to complete the operation.

With the ancilla layers prepared in the appropriate state, the transformation from the $k = 1$ 2D color code to the stacked code can be induced by measuring the gauge operators in

a fault-tolerant manner similar to that of surface code, such that any errors do not spread between data qubits. At this point, the logical qubit is stored throughout the different stacks in the $(d-1)+1$ stacked code. We emphasize that the high-weight stabilizers of the stacked code are *not* measured at this stage. Rather, the logical transversal $\pi/8$ gate is performed, and we then immediately transform back to the 2D code (without any active error correction being performed on the stacked code). The transformation back to the 2D color code is induced by measuring the *original* stabilizers of the 2D code, and the ancillary 2D stacks and their Bell stabilizers. Because the measurements can be performed fault-tolerantly without spreading errors, the code is protected by a distance d code at all times, and any error that occurred throughout the process can be inferred from the final measurements, as explained below.

Having returned to the original 2D code, the computation can continue with the application of transversal Clifford gates before potentially doing the same process for another $\pi/8$ gate at a different point in the computation. It is worth noting that the ancilla state is required to be measured fault-tolerantly through repeated measurements in order to correctly infer the errors on the final 2D color code after completion of the gate. Therefore, this ancilla remains "ready" at this stage for future non-Clifford computation and does not have to be re-prepared.

What remains to be shown is how an error that occurs while the information is encoded in the stacked code can be inferred from the final 2D code plus ancilla measurements. Suppose an error of weight less than d occurred while the state is encoded in the stacked code. Because the logical $\pi/8$ gate is transversal, errors may transform but will not increase in weight as a result of the logical gate. Therefore, such an error will remain of weight less than d. As such, if one were to measure the stabilizers of the stacked code, one would see a change in the sign of one of the cell stabilizers. Suppose the error anticommutes with cell $G_{P_i}^{(2k-1)}G_{P_i}^{(2k)}$ (this corresponds to an Z error, a similar argument follows for X errors). The presence of the error can be inferred from the measurement of the original stabilizers of the 2D planes, because the product of the individual outcomes of measurements $G_{P_i}^{(2k-1)}$ and $G_{P_i}^{(2k)}$ will be equivalent in sign to the measurement of the cell of the stacked code. It should be noted that the sign of the individual measurements will not necessarily be preserved, because the individual stabilizers of the 2D sheets anticommute with the gauge operators. However, the effect of these sign changes will simply be to set the stabilizer reference frame for subsequent measurements. Finally, if the error anticommutes with a blue stabilizer of the form $X_L^{(2k)}X_L^{(2k+1)}$, one can still infer the error from the measurement of the individual operators on the sheets and the joint logical measurements along the shared boundary of the sheets. We return to this last point in Sec. 3.2.4.



(a) (b)

Figure 3.7: A 2D layout for the implementation of the stacked code ($d = 5$ shown). Pairs of copies of the 2D hexagonal color code are layered on top of one another in a single 2D layer, in such a way as to keep the gauge operators geometrically local. (a) Initial layout of the stacked code transformation in 2D. The 2D layers $(2k)$ and $(2k + 1)$ are coupled by measuring joint logical X and Z operators (Bell stabilizers), with supporting qubits shown in blue. Although Bell stabilizers for the stacked code are high-weight, involving all blue qubits, the only required measurements are those associated with local 2D stabilizer/gauge operators together with one-dimensional operators of weight $\mathcal{O}(d)$ (shaded blue). The only 2D plane that is not initially coupled to another layer (or ancilla qubit) is the bottom $k = 1$ layer, which stores the encoded qubit. (b) Measurement of the weight-4 Z-type gauge operators, shown in Red. X-type stabilizers from individual layers are combined to form cell-like stabilizers by stabilizer evolution. Original joint logical X measurements, given by Blue shaded region, are mapped to all Blue qubits.

3.2.4 Unfolding the stacked code: A 2D implementation

Our stacked code provides a mechanism for performing a fault-tolerant logical $\pi/8$ gate on a qubit encoded in a 2D color code by switching to a third dimension. It requires the measurement of high-weight Bell stabilizers that couple pairs of 2D color codes—a requirement that is *not* necessary if one used the related approach of dimensional jumps in gauge color codes [22, 23], wherein the 3D color codes have low-weight, geometrically local stabilizers in three dimensions.

In this section, we show that our stacked code has a key advantage over more standard 3D color codes possessing geometrically local stabilizers, in that it can be arranged in a *two-dimensional* geometry. For the transformation to and from the stacked code in 2D, we require only geometrically local (in 2D) gauge measurements in the bulk, together with Bell stabilizers measurements along one-dimensional boundaries in this 2D layout.

3.2.4.1 Arranging the stacked code in two dimensions

Consider the 2D layout of different copies of the 2D hexagonal color code presented in Fig. 3.7, where layers $(2k-1)$ and $(2k)$ are combined into a single 2D plane and neighboring pairs of layers are arranged next to each other within this 2D plane, equivalent to the doubled color codes of Ref. [30]. The geometric arrangement can be viewed as unfolding the pairs of copies of the 2D code separated by the Bell stabilizers and tiling the pairs in a 2D plane. We shall refer to this arrangement as the *unfolded stacked code*. While it is visually useful to place layers $(2k-1)$ and $(2k)$ separated vertically as in Fig. 3.7, the qubits in these layers can be arranged in a single 2D plane; see Fig. 3.8.

The key feature of this geometric arrangement, which we show in the next section, is that the Bell stabilizers between layers $(2k)$ and $(2k+1)$ can be measured along the shared boundary. Although not geometrically local, this is a very desirable type of measurement from the perspective of physical implementations as the measurement is along a single 1D strip defined by the boundary of the two layers, and may be performed by coupling to a common mode or bus. One way to ensure fault-tolerance for such a measurement would be to prepare an ancillary state for readout, such as a cat state [126, 51], and repeat the measurement $\mathcal{O}(d)$ times [2]. The qubits composing the cat state could be arranged along the boundary, and because they will have to be measured to infer the logical measurement, they will be reset and available for the next round of measurement. We note that the scheme is not limited to performing this measurement using a cat state. Any fault-tolerant readout scheme for these high-weight operators may be applied here, assuming it can conform to the architectural constraints. We leave this for future work. A nonlocal operation, such as the one described here, is a necessary feature in order to circumvent the Bravyi–König no-go theorem for constant-depth logical gates outside the Clifford group in topological stabilizer codes in two dimensions [33]. The resulting code is equivalent to the stacked code, as the joint logical measurement operators along the boundary are mapped to 2D sheets due to the modification of the stabilizers by the gauge measurements.

3.2.4.2 Transformation of the joint boundary Bell stabilizers

In order to understand the transformation of the Bell stabilizer operators along the boundary, we consider the transformation of stabilizer operators under measurement of anti-commuting Pauli operators. The Z-type Bell stabilizer measurement is straightforward, because the gauge measurements are of type Z and thus a Z-type Bell stabilizer along the boundary remains of that form. This statement is equivalent to the fact that the volume

operator of weight $\mathcal{O}(d^2)$ can be mapped to a boundary plaquette operator due to the gauge Z measurements.

Next, we consider the transformation of the joint X logical boundary operators. Consider an instance of two pairs of 2D codes that are connected by joint logical string operators $X_{L,s}^{(2k)} X_{L,s}^{(2k+1)}$ and $Z_{L,s}^{(2k)} Z_{L,s}^{(2k+1)}$, initially shown in Fig. 3.7a. Let $\{H_{c_i}^{(2k-1)} H_{c_i}^{(2k)}\}$ denote the set of gauge operators that touch the joint logical boundary for 2D layers $(2k-1)$ and $(2k)$ of color c, indexed by the label c_i. Because these Z operators only intersect with $X_{L,s}^{(2k)} X_{L,s}^{(2k+1)}$ at a single qubit, these operators anti-commute. Additionally, $\{H_{c_i}^{(2k-1)} H_{c_i}^{(2k)}\}$ anti-commutes with the individual $G_{P_{c_i}}$ plaquette operators of matching color from the individual 2D codes $(2k-1)$ and $(2k)$. The stabilizers of the code are thus modified as follows: $\{H_{c_i}^{(2k-1)} H_{c_i}^{(2k)}\}$ becomes a new stabilizer of the code, replacing $G_{P_{c_i}}^{(2k)}$. Then, $G_{P_{c_i}}^{(2k-1)}$ is modified by being multiplied by the replaced stabilizer, thus becoming the cell operator $G_{P_{c_i}}^{(2k-1)} G_{P_{c_i}}^{(2k)}$. Finally, the joint logical operator is also modified by being multiplied by all replaced plaquettes of color c, that is, it becomes

$$\Big(\prod_{c_i} G_{P_{c_i}}^{(2k)}\Big) X_{L,s}^{(2k)} X_{L,s}^{(2k+1)} \,. \tag{3.34}$$

Because similar joint gauge Z measurements are performed between layers $(2k{+}1)$ and $(2k{+}2)$, the original joint boundary operator is mapped to the operator:

$$\Big(\prod_{c_i} G_{P_{c_i}}^{(2k)}\Big)\Big(\prod_{c'_j} G_{P_{c'_j}}^{(2k+1)}\Big) X_{L,s}^{(2k)} X_{L,s}^{(2k+1)} \,, \tag{3.35}$$

which corresponds to all qubits on layers $(2k)$ and $(2k+1)$. An example of the modified joint logical operator is shown in Fig. 3.7b. The joint logical X operator is spread over the full 2D lattice, as governed by the transformation of stabilizer operators, and thus becomes one of the blue cells shown in Fig. 3.5.

3.2.4.3 Implementation of a fault-tolerant $\pi/8$ gate in two dimensions

We now describe how to perform a fault-tolerant $\pi/8$ gate using this stacked code arranged in two dimensions. We initialize with the information encoded into a 2D color code and pairs of 2D codes laid out edge-to-edge in a 2D arrangement. Bell stabilizers are measured along 1D boundaries between two single sheets from different pairs, before finally measuring out the gauge operators in a local manner between pairs of 2D sheets. Having completed

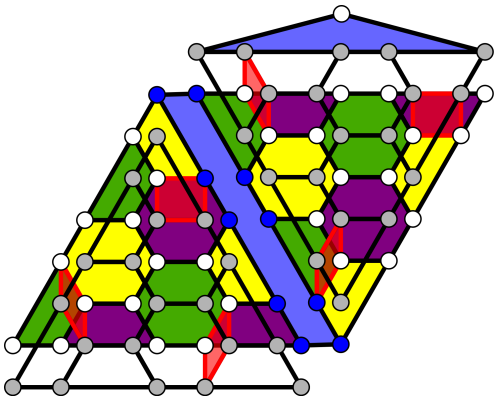


Figure 3.8: A two-dimensional layout of the construction presented in Fig. 3.7. The two originally superimposed lattices have respective grey and white lattice qubits. Only one of the color code stabilizers (per pair) have been colored, for clarity. Gauge measurement operators are given by red faces. Here, we have identified three individual gauge measurements per pair of codes for clarity, there are actually $3(d^2-1)/8$ such gauge measurements for each pair of distance d codes.

this process, the original information of the 2D code is now stored in a stacked code, and the non-Clifford $\pi/8$ gate can be executed transversally. After completion of the gate, the process is reversed by measuring the original stabilizers of the 2D code and ancilla qubits. The information is mapped back into the 2D color code, where transversal Clifford gates are available for further logical computation.

We emphasize that the expanded joint logical operators are never measured in practice, as the transformation from the 2D color code to the stacked code only serves for the application of the logical $\pi/8$ gate. Because the code has distance d throughout the process without coupling qubits during the measurements, the procedure remains fault-tolerant. If an error of weight less than d were to occur while the state was encoded in the stacked code, such an error will anticommute with one of the stabilizer cells of the stacked code. We covered the case when it anticommutes with one of the cells of the original 2D code color in Sec. 3.2.3.3. Thus, consider the case where the error anti-commutes with $X_{L,2D}^{(2k-1)} X_{L,2D}^{(2k)}$, where this joint logical operator is across the full 2D surface of the sheets. However, note the following:

$$X_{L,2D}^{(2k-1)} X_{L,2D}^{(2k)} = \prod_i G_{P_{c_i}}^{(2k-1)} \prod_j G_{P_{c_j}}^{(2k)} \left(X_{L,b}^{(2k-1)} X_{L,b}^{(2k)} \right),$$

where $X_{L,b}^{(2k-1)}X_{L,b}^{(2k)}$ is the joint boundary operator of color c that is shared by both 2D sheets, and $G_{C_i}^{(l)}$ are the individual X stabilizers of color c of the two sheets. Therefore, the product of the outcome of all these individual measurements will have to be preserved, that is by taking their product one can infer the measurement outcome of the joint logical operator across the full 2D sheets, as given by the blue qubits in Fig. 3.7b. As such, this large weight operator does not actually have to be measured to ensure fault-tolerance and rather it is sufficient to measure the individual 2D operators and joint-logical operators along their boundary after the completion of the transversal $\pi/8$ gate.

This construction results in a fault-tolerant application of a logical $\pi/8$ gate, yet the growing size of the joint boundary operators leave open the question of whether a rigorous fault-tolerance threshold exists. We note that, although the subdivision gadget of Ref. [30] establishes a method to reduce the overall weight of the individual operators that have to be measured, it bears similarities to weight reduction techniques proposed in subsystem codes [10] which exhibit a decreasing pseudothreshold for each distance rather than a threshold.

3.2.4.4 Comparison to Bravyi–Cross result

We briefly compare our construction with that of the very recent parallel result by Bravyi and Cross [30]. In that paper, the authors present a construction of a code for the application of a transversal $\pi/8$ gate through the construction of a triply even code from multiple copies of doubly even codes. They use a construction that mirrors the construction presented here, where 2D color code lattices are chosen with two qubits per site, denoted "doubled color codes." Each 2D lattice interacts with another 2D lattice through a joint logical measurement at their boundary (the Bell stabilizers presented in our work). A key insight in Ref. [30] is the proposal of a method to measure the Bell stabilizers using only local gauge measurements by applying a "subdivision gadget." Jones, Brooks, and Harrington recently proposed a similar method for breaking down the measurement of the Bell stabilizers in the construction of triply even codes based on the 2D [4.8.8] color code [81]. Their construction is inspired by lattice surgery methods for the implementation of joint logical measurements between two copies of 2D color codes [72, 97].

Another key contribution of Ref. [30] is the development of an online decoder to handle the transformation of Pauli errors to non-Pauli errors due to the action of the non-Clifford $\pi/8$ gates. Because this gate transforms X errors into a form of correlated X and Z errors, this can cause difficulties in the decoding of such errors. The authors introduce a Pauli twirling map after the application of the transversal $\pi/8$ in order to map the original

X error to a probabilistic application of Z errors in combination with the original X error. This twirling map allows for the construction of a maximum likelihood decoder for error correction. Techniques developed for the purpose of this decoder could potentially be applied to our construction as well.

3.2.5 Mapping from a larger distance 2D color code to the stacked code

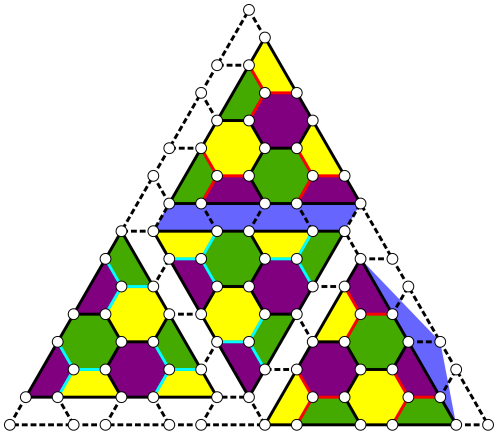


Figure 3.9: Initial coupling of split regions of a 2D color code. The original code is split into multiple color code copies by turning off and modifying certain stabilizer measurements. Different patches are coupled to form a Bell pair by measuring joint logical X and Z stabilizers between them, shown in blue. The patch that is not coupled in this way retains the quantum information that was originally stored in the code. The different patches are then further joined together by measuring gauge operators by matching up weight-2 edges from the different patches (forming weight-4 gauge operators), shown by red and cyan edges.

In this section, we describe a procedure to construct the stacked code as a reduction of a single higher distance 2D color code. This analysis is provided not as a direct means to implement the stacked code in 2D, as we believe the scheme outlined in Sec. 3.2.4.1 is a more practical approach. Rather, we introduce this scheme in order to analyze the scaling of the distance of the stacked code architecture when constructed from a larger 2D color code. The motivation of this analysis is to characterize the degree of nonlocality that is

required for stacked codes as a function of the larger 2D distance d_2 in order to implement a non-Clifford transversal gate.

To convert between the 2D architecture and the stacked code architecture, consider initializing a qubit encoded in a higher distance 2D color code, with distance $d_2 \geq d\sqrt{d-1}+1$, where d is the target distance of the stacked architecture. The initial 2D code is then converted to multiple copies of smaller color codes by "turning off" certain stabilizers and changing the weighting of others, while simultaneously measuring joint logical X and Z operators between neighboring pairs of these newly formed smaller regions, as shown in Fig. 3.9. The logical qubit that was encoded in string operators spanning the full distance of the 2D code is mapped by this process to only a single patch in the 2D layout—the patch that is not paired with another. This process corresponds to initializing the different layers of the 3D stacked code before the measurement of the gauge operators. The errors that occur can be tracked by recording the statistics of the measurement of the stabilizers before and after their modification, mirroring the technique for various logical gates in the 2D surface code [55]. The disadvantage of this architecture is that the gauge operators have to be measured by pairing qubits at different spatial locations in the 2D code, by matching individual edges in each code to form weight-4 operators. A particular set of edges that could be used for gauge measurements in the case of $d = 5$ is identified in Fig. 3.9. However, the pairings remain relatively local with respect to the 2D code distance as their separation is $\mathcal{O}(d) = \mathcal{O}(d_2^{2/3})$, which is the same order of nonlocality as the required joint logical measurements. Therefore, by modifying stabilizer measurements and performing joint measurements whose spatial nonlocality is of order $\mathcal{O}(d_2^{2/3})$ one can logically map a 2D color code to a 3D color code, thus providing the framework to perform a transversal $\pi/8$ gate and enabling a universal set of fault-tolerant gates.

The distance penalty one pays for such a process is a reduction from d_2 to $d_2^{2/3}$, however note that for two color codes with the same number of physical qubits n, the distance of the 2D color code has scaling $d_2 = \mathcal{O}(\sqrt{n})$ while the 3D color code has scaling $d = \mathcal{O}(n^{1/3}) = \mathcal{O}(d_2^{2/3})$. Such a distance penalty is to be expected, as the no-go result of Bravyi and König states that any circuit of depth h whose individual gates have geometric nonlocal range r that satisfies $hr \ll d^{1/2}$, for a 2D topological stabilizer code, can only implement a gate from the Clifford group. Therefore, it should be expected that if one can map to code that can implement a transversal $\pi/8$ gate the degree of geometric nonlocality must be at least of order $\mathcal{O}(d_2^{1/2})$, which our scheme clearly satisfies (yet does not saturate). Whether there exists methods to implement a fault-tolerant non-Clifford gate in 2D using a reduced degree of nonlocality is an interesting open problem.

3.2.6 The existence of a threshold?

In this section we present arguments for why the stacked code construction, while being fault-tolerant, will not have a threshold. First recall that a threshold is an error rate p below which the noise rate can be made arbitrarily small as the code distance d grows. We denote the pseudothreshold of the code for a given distance d, as the error rate p_d below which the logical error rate at that distance $p_{L,d}$ is smaller than the physical error rate. That is, $p_{L,d} < p,\ \forall\ p < p_d$. We have already seen examples of the difference between the threshold and pseudothreshold in Section 2.3 for the concatenated setting.

Consider the stacked code of distance d. While the stabilizers of the stacked code are not measured directly, its stabilizers are inferred from the 2D measurements once mapping back into 2D in order to guarantee error correction. As such, any non-trivial error that commutes with the stabilizers of the stacked code will result in a logical error. Consider a potental Z error connecting the bottom of the stacked code to the top, resulting in a logical error. If on two consecutive layers that are not separated by a Blue joint stabilizer, there are Z errors on the same qubit locations, such errors will go undetected by the stabilizers between these two layers. Given that there are $n = cd^2$ qubits per layer, where c is a constant, there will be that many possible ways for such errors to occur. Again, the same argument will hold for the next two layers, and moreover these errors will now go undetected by the Blue stabilizers as there will be an error on the two layers above and below, therefore of even parity. This argument continues until the top of the stack.

Roughly, the probability of such an error occurring will be on the order of Np^d, where p is the error probability, and N is the total number of such length d errors. Since there are cd^2 possible pairs of errors for each pair of layers, $N = ((cd^2))^{d/2}$ and therefore the logical error rate will be bounded as:

$$\begin{aligned} p_{L,d} &\leq Np^d && (3.36)\\ &= (cd^2)^{d/2}p^d && (3.37)\\ &= (\sqrt{c}dp)^d. && (3.38) \end{aligned}$$

As such, in order for the logical error rate to be below the physical error rate, we know that at the very least $\sqrt{c}dp < 1$ which implies $p_d < 1/(\sqrt{c}d)$, that is the pseudothreshold must scale inverse proportionally with d. Since the psudothreshold has distance dependance, no threshold can exist. However, it should be emphasized that in all likelihood the distance that would be required to implement interesting quantum algorithms would not necessarily be too high, and as such the constant prefactors in determining the pseudothreshold will likely dominate over the distance dependance.

3.2.7 Summary & outlook

Here, we have introduced stacked codes: a class of 3D color codes composed of individual 2D color code layers. We present a method to implement a universal set of logical gates transversally based on a 2D topological stabilizer code. We show that by layering pairs of 2D hexagonal color codes and connecting individual copies of the color code from different pairs through the measurement of nonlocal Bell-like stabilizers, we can then use gauge measurements as proposed in previous works [114, 22, 6] to map the logical information initially stored in a 2D color code into a stacked code. This fault-tolerant transformation allows for the application of a transversal gate outside the Clifford group in a 2D layout without having to resort to magic state distillation. Our proposal circumvents the Bravyi–König no-go result for transversal non-Clifford gates in 2D stabilizer codes by relying on a realistic form of nonlocal measurements along 1D boundaries in the 2D lattice.

Due to the growing size of the joint boundary operators, it is argued the proposed scheme for fault-tolerant universal computation does not exhibit a threshold in contrast to traditional 3D gauge color codes [34]. However, it remains of interest to establish the value of the pseudothreshold for low distance realizations of this scheme for the purposes of near-future experiments as well as potential multilayered quantum error correcting architectures, as in Ref. [47]. Moreover, the stacked codes merit further investigation into their stabilizer measurement properties, because 3D gauge color codes have the capacity for single-shot measurement [24]. Further research into the development of schemes for nonlocal operations to map a 2D stabilizer code to a 3D code, such as the recent proposal in Ref. [30], could lead to great reductions in architectural complexity and qubit overhead for the implementation of universal fault-tolerant quantum logic.

3.3 Other methods to circumvent magic state distillation

The last two chapters have presented methods for implementing a universal gate set without the need for magic state distillation. Given the expensive overhead associated with magic state distillation, this has been a very active area of research and it should be noted that alternative methods have also been proposed for completing the universal fault-tolerant gate set.

3.3.1 Stabilizer state preparation methods

Before the discovery of magic state distillation, alternative ancillary state distillation techniques were proposed. The first was a method developed by Shor to implement the Toffoli gate in a fault-tolerant manner by preparing an ancillary cat state offline and further extended to general stabilizer codes by Gottesman [126, 60]. Knill, Laflamme, and Zurek developed a scheme to implement the Hadamard gate fault-tolerantly on the 15-qubit code using a method that was equivalent to gate teleportation for the H gate [89]. In order to implement this scheme, the logical $|+\rangle$ state had to be prepared fault-tolerantly which required some overhead, yet would be required in any fault-tolerant implementation.

3.3.2 Gauge fixing

The 15-qubit code has the ability to perform transversal T and $CNOT$ gates, however lacks the ability to perform a transversal H (or any other gate that would allow for universality). However, consider the action of the transversal H gate on the stabilizers of the code, originally presented in Eq. 2.51. The stabilizer generators of the code will map to:

$$\begin{array}{cc} IIIIIIIXXXXXXXX & IIIIIIIZZZZZZZZ \\ IIIXXXXIIIIXXXX & IIIZZZZIIIIZZZZ \\ IXXIIXXIIXXIIXX & IZZIIZZIIZZIIZZ \\ XIXIXIXIXIXIXIX & ZIZIZIZIZIZIZIZ \\ \hline IIIIIIIIIIIXXXX & IIIIIXXIIIIIIXX \\ IIIIIIIIXXIIXX & IIIIXIXIIIIIXIX \\ IIIIIIIIXIXIXIX & IIXIIIXIIIXIIIX, \end{array} \tag{3.39}$$

where all of the stabilizer generators above the horizontal line are also present in the original formulation of the code, while those below have changed from being Z-type to X-type. Additionally, the logical X_L will be mapped to Z_L, and vice versa as per the action of the logical H. However, note as well that the X and Z stabilizers above the line are sufficient to correct an arbitrary weight-1 error of either type. In fact, using only these stabilizer generators will restrict the code to only being of distance 3 in both Z and X errors, as opposed to originally being distance 7 in error of X type. What Paetznick and Reichardt showed was that by remeasuring the missing Z stabilizer generators, one could forgo the information in the new X stabilizers and map back to the original code [114]. Moreover,

this process could be done in a fault-tolerant manner, thus allowing for the application of transversal H without the need for magic state distillation. The method is called gauge fixing, as the additional stabilizers can be interpreted as fixing a particular gauge, and while the action of the transversal H will change the gauge, the original gauge can be recovered without disrupting the logical state and still allowing for error correction.

This idea was further generalized by Bombín to the family of 3D color codes [23], which allows for the application of the transversal T gate for a family of codes with growing distance, as presented in Sec. 3.1.3. In the 3D code, the X stabilizers correspond to cells in a 3D structure, while the Z stabilizers correspond to faces between the cells. However, the Z faces can be multiplied in order to infer Z stabilizers on the same cells as the X stabilizers. Then, the remaining Z stabilizers will form the gauge syndromes, that will be remeasured after the action of the transversal H.

3.3.3 Dimensional jumps

We have already seen in Sec. 3.2.2 the idea of mapping between the 7-qubit an 15-qubit code in a fault-tolerant manner using gauge measurements [6]. To recap, suppose an arbitrary state is encoded in the 7-qubit code, allowing for the application of transversal Clifford operators, then an ancillary stabilizer state is prepared offline fault-tolerantly, and then can be fused to the original code through gauge measurements, a subset of the same gauge measurements that were present in the gauge fixing method. In order to guarantee fault-tolerant properties of the code, the X and Z distance of the scheme will be the same. The logical state will then be encoded in the 15-qubit code and the transversal T gate can be applied to complete the universal gate set. The code can be mapped back to the 7-qubit codespace by remeasuring the original stabilizer of the code.

Bombín generalized this method to logical systems of larger distances [24]. Namely, if the logical state is now instead encoded in a 2D color code of arbitrary distance, an ancillary state can be prepared offline that will allow for an equivalent mapping. The ancillary state that is prepared offline is a 3D bulk stabilizer state, that is the eigenstate of local stabilizers in 3D. Then, the bulk fixed state can be fused to the encoded state by performing a set of gauge measurements between the boundary of the bulk ancillary state and the 2D plane. Using this method, the original 2D state is mapped to a 3D state, allowing for the application of the transversal T gate. Again, the logical state can then be mapped back to the original 2D color code by remeasuring the original stabilizers of the code.

3.3.4 Pieceable fault-tolerance

A recent addition to the set of schemes for universal fault-tolerant logic is the idea of Pieceable Fault-Tolerance [140]. In this method, rather than attempting to complete the universal gate set in a transversal manner by mapping between code, the gate implemented will necessarily couple different qubits within the code, similar to the scheme presented in Chapter 2. However, unlike the previously presented method, protection is not guaranteed by using a different code, but rather by breaking the non-transversal operation into different pieces. Between each segment, a subset of the stabilizers are remeasured to allow for the correction of errors. Therefore, while if all of the gates were strung together the operation would not be fault-tolerant, by breaking it into pieces, errors can be managed and fault-tolerance restored. In this scheme, rather than complete the universal gate set with the T gate, typically the Toffoli or Controlled-controlled-Z gate are applied in a transversal manner, and can be generalized to high distance codes.

Chapter 4

Private quantum channels

The ideas presented in this chapter encompass the research that was developed in Refs. [76, 77], both copyrighted by the American Physical Society. All authors contributed to the development of the research program and ideas, with David Kribs providing the initial catalyst for the development of this project due to his contributions to the connection of privacy and quantum error correction [90]. Sarah Plosker and myself did the majority of the writing of the manuscript, with all four authors contributing in the editing of the manuscript.

4.1 Privacy vs. quantum error correction

Quantum error correction serves as a means to protect quantum information from external errors. However, as was discussed in Chapter 1.3.4, any quantum channel can be thought of as a unitary channel on a higher dimensional Hilbert space, where external degrees of freedom are traced out due to the observer not having access to the extension of the channel. This allows for the establishment of a complementary channel which characterizes the flow of information to the external environment. As such, the protection of the quantum information by using an error correcting code, and allowing for the recovery of information, should prevent any information about the state of leaking to the external environment. This can be thought of as a consequence of the no-cloning theorem since there exists no quantum operation that should be able to clone arbitrary quantum states. If the quantum error correcting code allows for full recovery of the logical state, then if any element of the state has leaked to the environment and as such could be recovered, then that element of

the arbitrary state must have been copied which should not be allowed. Therefore, the external system has no knowledge of the original logical state and the channel introduces a notion of privacy. More generally, we can define an operator private quantum channel in the following sense:

Definition 56. *A subsystem B is an **operator private subsystem** for Φ if for all $\sigma_A \in \mathcal{L}(A)$ there is a $\rho_0 = \rho_0(\sigma_A) \in \mathcal{L}(S)$ such that*

$$\Phi(\sigma_A \otimes \sigma_B) = \rho_0, \quad \forall\ \sigma_B \in \mathcal{L}(B). \tag{4.1}$$

We will arrive at more general notions of privacy later in this chapter, however we first introduce the following notion due to its connections to a class of quantum error correcting codes called operator quantum error correcting codes.

Definition 57. *Let $S = (A \otimes B) \oplus (A \otimes B)^{\perp}$ and let $\mathcal{E}$ be a channel acting on $\mathcal{L}(S)$. Then B is an **operator quantum error correcting code (OQECC)** for $\mathcal{E}$ if there exists a quantum channel $\mathcal{R}$ such that for all σ_A, for all σ_B, there exists some fixed state $\tau_A = \tau_A(\sigma_A)$ (dependent on σ_A) such that*

$$\mathcal{R} \circ \mathcal{E}(\sigma_A \otimes \sigma_B) = \tau_A \otimes \sigma_B. \tag{4.2}$$

Equipped with these two notions, we can now more formally state the duality between a channel that is private and one that is error correctable.

Theorem 58 ([90])**.** *Let $S = (A \otimes B) \oplus (A \otimes B)^{\perp}$ and let Φ be a channel acting on $\mathcal{L}(S)$. Then B is an operator private subsystem for Φ if and only if it is an operator quantum error correcting code for the the complementary channel $\Phi^{\sharp}$.*

Therefore, by developing a stronger sense of the theory of private quantum channels, one can hope to gain some understanding in potential error correction models through this complementary link. This chapter aims to further characterize the set of channels that can be private and uncover the underlying mathematical structure behind them.

4.2 Private quantum subsystems

4.2.1 Introduction & background

The most essential primitive for private communication between two parties, Alice and Bob, in classical computation is the one-time pad. In such a scheme, the two parties

share a secret key that is unknown to an external observer Eve; this key enables reliable communication by the parties as the message appears to be a random mixture of input bits from Eve's viewpoint without the key.

Private quantum codes were initially introduced as the quantum analogue of the classical one-time pad. The basic setting for a "private quantum channel" [5, 27] is as follows: Alice and Bob share a private classical key that Alice uses to inform Bob which of a set of unitary operators $\{U_i\}$ she has used to encode her quantum state: $\rho \mapsto U_i \rho U_i^\dagger$. With this information in hand, Bob can decode and recover the state ρ without disturbing it. The set of unitaries $\{U_i\}$ and the probability distribution $\{p_i\}$ that makes up the random key which determines the encoding unitary are shared publicly. Thus, without further information, Eve's description of the system is given by the random unitary channel $\Phi(\rho) = \sum_i p_i U_i \rho U_i^\dagger$. By selecting certain sets of unitary operators with appropriate coefficients, the random unitary channel will provide Eve with no information about the input state.

The body of work on private quantum codes now includes a variety of other applications, with realizations both as subspaces and subsystems of Hilbert space. Private shared reference frames exploit private subspaces and subsystems that also arise from the ignorance associated with an eavesdropper's description of a system [14, 13]. The notion of using mixed state ancilla qubits to encode information, which can be viewed as subsystem encodings, has also been studied in the context of quantum secret sharing [43, 45]. There, the goal is to encode information into a globally mixed state of n qubits such that to recover the quantum information one would need access to k qubits of the global state, where any fewer would yield no information regarding the initial state. Using mixed states allows for the increase of k for a fixed n, thus solidifying the idea that mixed state encodings increase privacy. There are also bridges between these works and quantum error correction, formalized by the complementarity results of [90]. Connections between the study of private quantum codes and the theory of operator algebras have recently been found as well [41].

In this work we consider the most general notion of a private quantum code [5, 27, 14], which involves the encoding of quantum bits into subsystems. Private quantum channels, private subspaces, and what we refer to as "operator" private subsystems—are captured as special cases of this general phenomena. We consider a class of phase damping channels throughout the presentation that highlights the main differences between mappings on subsystems and subspaces. Most surprisingly, we show that certain classes of channels can only be private in the subtle subsystem sense; thus establishing that private subsystems can exist in the absence of private subspaces.

We also make the first significant move toward a structure theory for private quantum codes; specifically we set out algebraic conditions that characterize privacy of a code in

terms of the Kraus operators for a given quantum channel. This can be viewed as an analogue of the set of Knill-Laflamme conditions [88] from quantum error correction to this setting, and indeed we discuss further connections with error correction. In particular we show that complementarity of private and error-correcting codes fails at the most general level, and we point out a potentially new type of quantum error-correcting code.

We now describe our notation and nomenclature. Given a quantum system S, with (finite-dimensional and complex) Hilbert space also denoted by S, we will use customary notation such as ρ, σ for density operators. The set of linear operators on S is denoted by $\mathcal{L}(S)$. Linear maps on $\mathcal{L}(S)$ can be viewed as operators acting on the operator space $\mathcal{L}(S)$. We use the term *(quantum) channel* to refer to a completely positive and trace-preserving linear map $\Phi : \mathcal{L}(S) \to \mathcal{L}(S)$. Such maps describe (discrete) time evolution of open quantum systems in the Schrödinger picture, and can always be written in the Choi-Kraus operator-sum form $\Phi(\rho) = \sum_i V_i \rho V_i^\dagger$ for some operators V_i in $\mathcal{L}(S)$ satisfying $\sum_i V_i^\dagger V_i = I$. The composition of two maps will be denoted by $\Phi \circ \Psi(\sigma) = \Phi(\Psi(\sigma))$.

A (linearly closed) subspace C of S is said to be a *private subspace* for Φ if there is a density operator ρ_0 on S such that $\Phi(|\psi\rangle\langle\psi|) = \rho_0$ for all pure states $|\psi\rangle$ in C. By linearity, $\Phi(\rho) = \rho_0$ for all ρ in $\mathcal{L}(C)$. We could also consider a collection of private states not associated with a subspace of the Hilbert space, but, as in quantum error correction, we wish to allow for arbitrary superpositions of our code states and this demands the set of states considered are linearly closed.

Definition 59. *A subsystem B is a* private subsystem *for Φ if there is a $\rho_0 \in \mathcal{L}(S)$ and $\sigma_A \in \mathcal{L}(A)$ such that*

$$\Phi(\sigma_A \otimes \sigma_B) = \rho_0 \quad \forall \sigma_B \in \mathcal{L}(B). \tag{4.3}$$

The case of random unitary channels Φ in Eq. (4.3) was first considered in [5, 27] where the terminology *private quantum channels* was used, and the case of general channels Φ was formalized in [14] where private subsystems were given the extra prefix "completely" that we have dropped. If Eq. (4.3) holds for all σ_A, as opposed to a single state σ_A, then we shall refer to B as an *operator private subsystem* as defined in Definition 56 (since these are precisely the private subsystems that are complementary to operator quantum error-correcting subsystems discussed below).

4.2.2 Private subsystems in the absence of private subspaces

An operator private subsystem is one in which the private channel splits into a product of maps on the individual subsystems A and B when the channel is restricted to the

combined product subspace $A \otimes B$. Such private subsystems cannot exist without the existence of private subspaces; indeed, if Eq. (4.3) holds for all states on A, it follows that every sub*space* $|\psi\rangle \otimes B$ is private for Φ for any fixed pure state $|\psi\rangle$ on A.

Even though the definition given by Eq. (4.3) allows for the possibility of examples of private subsystems that do not extend to private subspaces, the private subsystems exhibited in the literature [5, 27, 14, 13] thus far have either been of operator type, or are already subspaces. Here we present the first examples of private subsystems for which there are no private subspaces that exist; in particular these are private subsystems that are not of operator type. Our motivating class of channels is built upon a simple phase damping model. We begin the discussion by recalling the most basic private quantum channel and asking some basic questions on quantum privacy.

The completely depolarizing channel ($\Phi(\rho) = \frac{1}{\dim S} I$ for all ρ) is an easy to describe example of a quantum channel that is private. In this case the entire Hilbert space acts as a private code for the channel, and so in order to implement such a private channel a full set of Pauli rotations must be available. This leads to a very basic question in the study of private quantum codes: Do there exist channels with fewer physical operations such that we can still encode qubits for privacy?

Perhaps the simplest class of channels one could imagine would be the family of phase damping channels that can be applied to any qubit of a larger Hilbert space $\mathcal{S}$ of n qubits,

$$\Lambda_i(\rho) = \frac{1}{2}(\rho + Z_i \rho Z_i), \qquad \forall \rho \in \mathcal{L}(S). \tag{4.4}$$

A single qubit phase damping channel is not private. Yet we can ask: can composing the phase damping channel on multiple qubits yield a private subspace $C \subseteq \mathcal{S}$? Such a question is analogous to the sort of questions that have been asked in quantum error correction for some time; for example, given a set of errors that are uncorrectable on a single qubit, does there exist a larger Hilbert space such that the action of the error on the encoded Hilbert space is correctable? The answer to such a question in quantum error correction is yes, as demonstrated by the five-qubit code which corrects for arbitrary single-qubit errors, an error that would be uncorrectable if one did not have access to a larger Hilbert space to encode the quantum information into a quantum code.

We shall define the map Λ as the composition of the maps Λ_i on each of the n qubits of the state $\rho \in S$,

$$\Lambda(\rho) \quad = \Lambda_n \circ \Lambda_{n-1} \circ \cdots \circ \Lambda_1(\rho). \tag{4.5}$$

Equivalently one could consider the n-product map $\Lambda_1^{\otimes n}$ of the single qubit channel Λ_1. For any input state ρ, this channel will decohere all off-diagonal terms in the computational basis; as such, the resulting output density matrix will be diagonal.

Consider the case when $n = 2$. Every output state of Λ has the form

$$\rho_0 = \frac{1}{4}\Big(II + \alpha IZ + \beta ZI + \gamma ZZ\Big), \tag{4.6}$$

where I and Z are the one-qubit identity and Pauli Z matrices. The goal is to find a subspace C of dimension 2 and a state $\rho_0 \in \mathcal{L}(S)$ such that $\Lambda(\rho) = \rho_0 \ \forall \rho \in \mathcal{L}(C)$. This would show that Λ has a private qubit subspace, defined by a pair of orthogonal logical states $|0_L\rangle$, $|1_L\rangle$ in C.

However, one can show that such a subspace *does not* exist. In fact we can prove the following more general result, which applies to the channels Λ directly, and can be extended to channels with commuting normal Kraus operators as well.

Theorem 60. *Let Φ be a random unitary channel with mutually commuting Kraus operators. Then Φ has no private subspaces.*

Proof. Let Φ be a random unitary channel with mutually commuting Kraus operators described by

$$\Phi(\rho) = \sum_i p_i U_i \rho U_i^\dagger \quad \forall \rho. \tag{4.7}$$

Since the unitaries U_i are mutually commuting, there exists a common eigenbasis $\{|e_j\rangle\}_{j=1}^d$ for all of the unitaries such that,

$$U_i|e_j\rangle = \alpha_{ij}|e_j\rangle \text{ with } \quad |\alpha_{ij}| = 1. \tag{4.8}$$

Suppose a non-trivial private subspace C exists. Then there must exist at least two pure states $|0_L\rangle$, $|1_L\rangle$ such that $\Lambda(|0_L\rangle\langle 0_L|) = \Lambda(|1_L\rangle\langle 1_L|) = \rho_0$, where ρ_0 is some fixed density matrix. Then for some scalars $\beta_j, \gamma_j \in \mathbb{C}$, we can write

$$|0_L\rangle = \sum_{j=1}^d \beta_j |e_j\rangle, \quad |1_L\rangle = \sum_{j=1}^d \gamma_j |e_j\rangle. \tag{4.9}$$

Consider the action of the channel on these states:

$$\begin{aligned}
\Phi(|0_L\rangle\langle 0_L|) \quad &= \textstyle\sum_i p_i U_i \Big(\sum_{j,k=1}^d \beta_j \beta_k^* |e_j\rangle\langle e_k| \Big) U_i^\dagger \\
&= \textstyle\sum_i p_i \sum_{j,k=1}^d \alpha_{ij}\alpha_{ik}^* \beta_j \beta_k^* |e_j\rangle\langle e_k| \\
&= \textstyle\sum_{j,k=1}^d \Big(\sum_i p_i \alpha_{ij}\alpha_{ik}^* \Big) \beta_j \beta_k^* |e_j\rangle\langle e_k|
\end{aligned} \tag{4.10}$$

Similarly,

$$\Phi(|1_L\rangle\langle 1_L|) = \sum_{j,k=1}^{d} \Big(\sum_i p_i \alpha_{ij} \alpha_{ik}^* \Big) \gamma_j \gamma_k^* |e_j\rangle\langle e_k|. \tag{4.11}$$

Comparing the diagonal terms, where $j = k$, the inside sum over i is always equal to 1 since the modulus of the eigenvalues is 1, thus the respective coefficients are $|\beta_j|^2$ and $|\gamma_j|^2$. Therefore, if the output of the channel is the same in both cases, we must have $|\beta_j| = |\gamma_j|$, $\forall j$. (Observe that this is independent of the orthogonality of $|0_L\rangle$ and $|1_L\rangle$; any two basis states mapped by Φ to the same state would satisfy this coefficient condition.) However, we prove below that no such $|0_L\rangle$ and $|1_L\rangle$ can form a subspace. Indeed, we can write

$$|0_L\rangle = \textstyle\sum_{j=1}^{d} \beta_j |e_j\rangle = |\beta_1||e_1\rangle + \sum_{j=2}^{d} |\beta_j| e^{i\theta_j} |e_j\rangle \tag{4.12}$$

$$|1_L\rangle = \textstyle\sum_{j=1}^{d} \gamma_j |e_j\rangle = |\beta_1||e_1\rangle + \sum_{j=2}^{d} |\beta_j| e^{i\phi_j} |e_j\rangle, \tag{4.13}$$

where we have, without loss of generality, performed a global phase shift on the two vectors so that the coefficient of $|e_1\rangle$ is real *for both vectors* (under a global phase shift, the vectors remain orthogonal). We have relabelled the coefficients to reflect the fact that $|\beta_j| = |\gamma_j|$.

Any linear combination of the basis states must additionally be in C by the closure of the subspace under scalar addition. With this in mind, consider the normalized state,

$$\frac{|0_L\rangle + |1_L\rangle}{\sqrt{2}} = \frac{2|\beta_1||e_1\rangle + \sum_{j=2}^{d} |\beta_j|(e^{i\theta_j} + e^{i\phi_j})|e_j\rangle}{\sqrt{2}}. \tag{4.14}$$

Since such a state must be an element of C, it must satisfy the conditions on the moduli of its coefficients; namely, the jth coefficient must be equal in modulus to $|\beta_j|$. However, one can clearly see that the modulus of the coefficient of the $|e_1\rangle$ term is equal to $\sqrt{2}|\beta_1|$ which is not equal to $|\beta_1|$ unless $|\beta_1| = 0$. Therefore, we have reduced the basis states to have the form,

$$|0_L\rangle = |\beta_2||e_2\rangle + \textstyle\sum_{j=3}^{d} |\beta_j| e^{i\theta'_j} |e_j\rangle \tag{4.15}$$

$$|1_L\rangle = |\beta_2||e_2\rangle + \textstyle\sum_{j=3}^{d} |\beta_j| e^{i\phi'_j} |e_j\rangle, \tag{4.16}$$

where we have performed a global phase shift on both states and redefined the phase on the components $|e_j\rangle, 3 \leq j \leq d$. By the same argument as above, we can show that all coefficients must be equal to zero in order for the channel Λ to be private while C remains a subspace. As such, there does not exist two orthonormal basis vectors satisfying the requirements for the channel to be private, implying that no non-trivial subspace $C \subset S$ exists. □

Corollary 61. *Let S be n-qubit Hilbert space. Then there exists no subspace $C \subset S$ where $dim(C) \geq 2$ such that C is private for the channel $\Lambda = \Lambda_n \circ \Lambda_{n-1} \circ \cdots \circ \Lambda_1$.*

Proof. All Kraus operators are tensor products of I_2 and Z, and thus Theorem 60 applies. □

Is this the end of the story? This result is intuitive—at first glance it certainly does not "feel" as though we should be able to find private codes for channels such as the phase damping maps $\Lambda = \Lambda_n \circ \cdots \circ \Lambda_2 \circ \Lambda_1$ due to the preservation of information stored in the diagonal elements of the initial density matrix. Moreover, the experience with operator private subsystems, which demand the existence of private subspaces, also suggests we can go no further with these channels. Somewhat surprisingly, we do find private subsystems for these channels, and necessarily they are not of the type exhibited before.

Consider the following logically encoded qubits in two-qubit Hilbert space:

$$\rho_L = \frac{1}{4}(II + \alpha XX + \beta YI + \gamma ZX). \tag{4.17}$$

This describes a single qubit encoding, as equation (4.17) describes the coordinates for a logical Bloch sphere in two-qubit Hilbert space with logical Pauli operators given by $X_L = XX, Y_L = YI, Z_L = ZX$. Now, observe that the dephasing map $\Lambda = \Lambda_2 \circ \Lambda_1$ acting on each density operator ρ_L produces an output state that is maximally mixed; that is, $\Lambda(\rho_L) = \frac{1}{4} II$ for all ρ_L. Thus, we see that equation (4.17) yields a private single-qubit code for the dephasing map Λ.

We claim that this private code can be viewed as a single qubit subsystem embedded inside two qubit space, where the ancilla operator σ_A, from equation (4.3), in this case is the single qubit identity operator I_2; that is, up to a unitary equivalence the set of operators ρ_L can be seen to generate the operator algebra $I_2 \otimes \mathbb{M}_2$. To see this, it is enough to show that all two-qubit states ρ of the form $\rho = \frac{1}{4}(II + \alpha XX + \beta YI + \gamma ZX)$ can be sent through appropriate unitary gates to obtain ρ' of the form $\rho' = I_2 \otimes \frac{1}{4}(I_2 + \alpha' X + \beta' Y + \gamma' Z)$. Since I, X, Y, Z form a basis for $\mathbb{M}_2$, the claim will follow.

We find that an application of the inverse of the K-gate,

$$K = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ i & -i \end{pmatrix} = \frac{1}{\sqrt{2}}(|0\rangle(\langle 0| + \langle 1|) + i|1\rangle(\langle 0| - \langle 1|)),$$

on the first qubit, and applications of $CNOT_{2,1}$ and $CNOT_{1,2}$, yields the desired transformation. Indeed, the composition $CNOT_{1,2}CNOT_{2,1}\big((K^\dagger \otimes I_2)(\cdot)(K \otimes I_2)\big)CNOT_{2,1}CNOT_{1,2}$

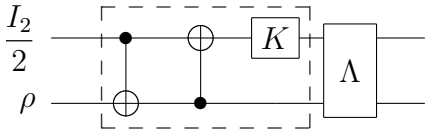


Figure 4.1: The gates in the box region implement the encoding of an arbitrary two-qubit state belonging to the $I \otimes \mathbb{C}^{2\times 2}$ algebra into encoded states of the form of Eq. (4.17). The encoded two-qubit state subjected to the two-qubit phase damping channel $\Lambda = \Lambda_2 \circ \Lambda_1$ results in an output of the completely mixed state on two qubits, $\rho_0 = \frac{1}{4} II$.

acts as:

$$\begin{aligned} XX &\longmapsto YX \longmapsto ZY \longmapsto IY \\ YI &\longmapsto ZI \longmapsto ZZ \longmapsto IZ \\ ZX &\longmapsto XX \longmapsto IX \longmapsto IX. \end{aligned}$$

Thus, we obtain $\rho' = \frac{1}{4}(I_4 + \gamma IX + \alpha IY + \beta IZ)$. In particular, by defining the unitary

$$U = CNOT_{1,2} \circ CNOT_{2,1} \circ (K^\dagger \otimes I_2) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & -i & 0 \\ 0 & 1 & 0 & i \\ 0 & 1 & 0 & -i \\ 1 & 0 & i & 0 \end{pmatrix},$$

we see the set of operators $U\rho_L U^\dagger$ generate the algebra $I_2 \otimes \mathbb{M}_2$.

Thus, this subsystem encoding fits into the framework of the definition of private quantum subsystem; that is, the subsystem defined by the set of operators $U\rho_L U^\dagger$ is a private subsystem for the channel Λ. In fact, it is a private subsystem that is not operator private. This follows from Theorem 60 together with the complementarity theorem of [90]: If the subsystem was operator private, it would complement an operator quantum error correcting code (discussed in the next subsection), which would imply the complementary channel also has a correctable subspace code of the same size, and then incorrectly imply that the original channel has a private subspace again by complementarity. For completeness, we will show directly below that this private subsystem is not operator private.

We have shown explicitly that $\mathcal{L}(C)$ is isomorphic to $I_2 \otimes \mathbb{M}_2$, where C is the set of all private states for Λ, via $U\mathcal{L}(C)U^\dagger = I_2 \otimes \mathbb{M}_2$. Alternatively, we can consider the modified channel $\Lambda'(\cdot) := U\Lambda(\cdot)U'$. Then the second qubit in the standard computational basis decomposition $A \otimes B$, $A = \mathbb{C}^2 = B$, is private for Λ' with $\sigma_A = \frac{1}{2} I_2$. That is, rather than applying the unitary transformation $\rho_L \mapsto U\rho_L U^\dagger$ and sending this resulting state through

the channel Λ, we can modify the Kraus operators of Λ by the same unitary U so that Λ' is private for $\frac{1}{2}I_2 \otimes \sigma_B$ for any $\sigma_B \in \mathcal{L}(B)$. In this manner, our example directly fits the definition of private subsystem.

For any

$$\sigma_A = \begin{pmatrix} a_A & b_A \\ c_A & d_A \end{pmatrix} \in \mathcal{L}(A), \quad \sigma_B = \begin{pmatrix} a_B & b_B \\ c_B & d_B \end{pmatrix} \in \mathcal{L}(B),$$

we compute

$$\Lambda'(\sigma_A \otimes \sigma_B) = \frac{1}{2} \begin{pmatrix} \gamma_{AB} & 0 & 0 & \eta_{AB} \\ 0 & \gamma_{AB} & \zeta_{AB} & 0 \\ 0 & \zeta_{AB} & \gamma_{AB} & 0 \\ \eta_{AB} & 0 & 0 & \gamma_{AB} \end{pmatrix}. \tag{4.18}$$

where $\gamma_{AB} = a_A a_B + d_A d_B$, $\eta_{AB} = b_A b_B + c_A c_B$, and $\zeta_{AB} = b_A c_B + c_A b_B$. Note that this output is symmetric in the subsystems A and B. In particular, $\Lambda'(\sigma_A \otimes \frac{1}{2}I_2) = \frac{1}{4}\text{diag}(a_A + d_A, a_A + d_A, a_A + d_A, a_A + d_A)$ and $\Lambda'(\frac{1}{2}I_2 \otimes \sigma_B) = \frac{1}{4}\text{diag}(a_B + d_B, a_B + d_B, a_B + d_B, a_A + d_B)$. Thus for density matrices σ_A, σ_B we have $\Lambda'(\sigma_A \otimes \frac{1}{2}I_2) = \Lambda'(\frac{1}{2}I_2 \otimes \sigma_B) = \frac{1}{4}I_4$, and so both the first and second computational basis subsystems are private for Λ'.

If we were looking at an operator private subsystem here, the channel Λ' would split up into two distinct channels acting on systems A and B respectively. Thus, we would have density matrices τ_A, τ_B such that $\Lambda'(\sigma_A \otimes \sigma_B) = \tau_A \otimes \tau_B$. Equating this equation with equation (4.18), we find the system has no solution for general σ_A, σ_B (equating components forces τ_B to be the zero matrix, which then forces $\Lambda'(\sigma_A \otimes \sigma_B)$ to be the zero matrix). Hence, this subsystem is private for Λ', but not operator private. More generally, a logical qubit encoding into a subsystem of a n-qubit Hilbert space can be constructed to privatize the n-qubit phase damping channel $\Lambda = \Lambda_n \circ \cdots \circ \Lambda_2 \circ \Lambda_1$, which by Theorem 60 cannot have a private subspace.

Theorem 62. *For any n-qubit Hilbert space $\mathcal{H}$, there exist quantum channels Φ for which a private quantum subsystem B of $\mathcal{H}$ can be constructed in the absence of the existence of any private quantum subspace $C \subseteq \mathcal{H}$.*

4.2.3 Testable conditions for private quantum codes

If we are given a quantum channel $\Phi(\rho) = \sum_i V_i \rho V_i^\dagger$ and a subsystem B, we can ask if it is possible to decide whether B is private for Φ; and more to the point, we can ask if this can

be answered in terms of the Kraus operators V_i for the channel. The analogous question in quantum error correction is answered by the fundamental Knill-Laflamme conditions [88], which provide an explicit set of algebraic constraints in terms of the Kraus operators and the code, and allow one to test whether a given code is correctable for a channel. The generalization of these conditions to the case of operator error-correcting subsystems was established in [91, 92, 109].

The following result answers this question for private quantum subsystems. In addition to Kraus operators, we would expect the algebra to include the fixed A state σ_A and output state ρ_0—observe that this information is indeed included in the conditions.

Theorem 63. *A subsystem B is private for a channel $\Phi(\rho) = \sum_i V_i \rho V_i^\dagger$ with fixed A state σ_A and output state ρ_0 if and only if there are complex scalars λ_{ijkl} forming an isometry matrix $\lambda = (\lambda_{ijkl})$ such that*

$$\sum_m \sqrt{p_k} V_j |\psi_{A,k}\rangle |\psi_{B,m}\rangle\langle\psi_{B,m}| = \sum_{i,l} \lambda_{ijkl} \sqrt{q_l} |\phi_l\rangle\langle\psi_{B,i}|, \tag{4.19}$$

where $|\psi_{A,k}\rangle$ (p_k) and $|\phi_l\rangle$ (q_l) are eigenstates (eigenvalues) of σ_A and ρ_0 respectively, $|\psi_{B,i}\rangle$ is an orthonormal basis for B.

The key observation in establishing this result is that the left and right hand sides of Eq. (4.3) each define channels from B to S which are in fact the same. One can then use basic results from the theory of completely positive maps to obtain the equations spelled out in the theorem.

Proof. Consider first the left-hand side of the equation (4.3) of the definition of private quantum subsystem. Let $\Phi : \mathcal{L}(S) \to \mathcal{L}(S)$ be a quantum channel satisfying this definition. Let $\{V_j\}$ be the Kraus operators of Φ. Consider a spectral decomposition $\sigma_A = \sum_k p_k |\psi_{A,k}\rangle\langle\psi_{A,k}|$, where $|\psi_{A,k}\rangle$ and p_k are the eigenstates and eigenvalues, respectively, of σ_A. We can consider the action of Φ on $\mathcal{L}(S)$ as the composition of maps $\Phi \circ \Psi(\sigma_B)$, where, for fixed σ_A, $\Psi : \mathcal{L}(B) \to \mathcal{L}(S)$ is the map $\sigma_B \mapsto \sigma_A \otimes \sigma_B$. The Kraus operators of Ψ are $\{\sum_m \sqrt{p_k} |\psi_{A,k}\rangle |\psi_{B,m}\rangle\langle\psi_{B,m}|\}_k$ (the $\sum_m |\psi_{B,m}\rangle\langle\psi_{B,m}|$ acts trivially on σ_B, but is necessary to obtain the correct dimension when later acted on by V_j). It follows that the Kraus operators of the composition $\Phi \circ \Psi(\sigma_B)$ are $\{\sum_m \sqrt{p_k} V_j |\psi_{A,k}\rangle |\psi_{B,m}\rangle\langle\psi_{B,m}|\}_{j,k}$.

On the other hand, the right-hand side of equation (4.3) can be viewed as a quantum channel

$$\sigma_B \mapsto \mathrm{Tr}(\sigma_B) \sum_l q_l |\phi_l\rangle\langle\phi_l| = \sum_{i,l} q_l |\phi_l\rangle\langle\psi_{B,i}|\sigma_B|\psi_{B,i}\rangle\langle\phi_l|,$$

where $\{|\psi_{B,i}\rangle\}$ is an orthonormal basis for the subsystem B, and we have used the fact that $|\phi_l\rangle$ and q_l form a spectral decomposition for ρ_0. The Kraus operators of this map are $\{\sqrt{q_l}|\phi_l\rangle\langle\psi_{B,i}|\}_{i,l}$.

However, the quantum channels described by the left- and right-hand sides of equation (4.3) are equal in that, given an arbitrary input σ_B, their outputs are equal. Thus we may use a well-known fact regarding equal CP maps with Kraus operators $\{X_i\}_{i=1}^m$ and $\{Y_j\}_{j=1}^n$, respectively, with $m \leq n$; that is, they are related via $X_i = \sum_j \lambda_{ij} Y_j$ for some isometry matrix $\lambda = (\lambda_{ij})$. When $m = n$, λ is unitary. It follows immediately from this that for all j, k, we have $\sum_m \sqrt{p_k} V_j |\psi_{A,k}\rangle|\psi_{B,m}\rangle\langle\psi_{B,m}| = \sum_{i,l} \lambda_{ijkl}\sqrt{q_l}|\phi_l\rangle\langle\psi_{B,i}|$, for some isometry (or, appropriately, unitary) λ, as desired.

The converse implication follows by reversing the above steps, or by direct calculation, to show that Eq. 4.19 implies Eq. 4.3 is satisfied. □

The conditions of Theorem 63 are somewhat intricate in the most general case, so it is worthwhile to give further context and discuss some special cases. We note that this result is new even for the special cases of operator private codes and private subspaces, and, via complementarity, the result can thus be viewed as the quantum privacy analogue of the Knill-Laflamme theorem for quantum error-correcting (subspace) codes [88] and its operator quantum error correction generalization [91, 92]. However, the most general case covered by Theorem 63 may have no analogue in quantum error correction. The next two subsections discuss this topic in more detail.

As one would expect, the algebraic conditions can be further simplified in the case of private subspaces; which is captured in the formalism when A is one-dimensional and B is a subspace. In this case, the Theorem statement becomes $V_j P_B = \sum_{i,l} \lambda_{ijl}\sqrt{q_l}|\phi_l\rangle\langle\psi_{B,i}|$. By taking the inner product of this equation with its complex conjugate, one arrives at the statement $P_B V_{j_1}^\dagger V_{j_2} P_B = \sum_{i_1,i_2,l} q_l \overline{\lambda_{ij_1l}}\lambda_{ij_2l}|\psi_{B,i_1}\rangle\langle\psi_{B,i_2}|$ for all j_1, j_2, where P_B is the projector onto the B subspace. Here we have a more noticeable connection with the Knill-Laflamme conditions for quantum error correction: $P_B V_{j_1}^\dagger V_{j_2} P_B = \alpha_{j_1,j_2} P_B$, where the V_j's are the Kraus operators of the *error map* and P_B is the projection onto the *correctable* B subspace.

The algebraic conditions of the theorem can also be simplified in the case that ρ_0 is a scalar multiple of a projection, as we now state.

Corollary 64. *Suppose the output state ρ_0 of a private quantum channel $\Phi = \{V_i\}$ is proportional to a projection: $\rho_0 \propto Q = \sum_k |\psi_k\rangle\langle\psi_k|$, and $P = \sum_l |\phi_l\rangle\langle\phi_l|$. It follows that*

there are scalars u_{ikl} such that for all i

$$V_i P = \sum_{k,l} u_{ikl} A_{kl} \quad \text{where} \quad A_{kl} = \frac{1}{\sqrt{rank(Q)}} |\psi_k\rangle\langle\phi_l|.$$

Here A_{kl} are the Kraus operators of the channel $X \mapsto \lambda_X P$.

Thus far in our investigations, most of the physical examples of private codes that we have come across do indeed have a projector output as in this Corollary. Of course, the simplest general class of channels satisfying this condition is the n-qubit complete depolarizing channel. In that case, both P and Q are the maximally mixed state, and the result indicates that any family of Kraus operators for the map will arise as linear combinations, where the scalars are precisely defined with the right balance to induce privacy, of the matrix units $|i\rangle\langle j|$. Another simple (non-unital) example is provided by the spontaneous emission channel. In the single qubit case, the extremal channel from this class is given by $\Phi(\rho) = |0\rangle\langle 0|$ for all single qubit ρ. Here P is the maximally mixed state and $Q = |0\rangle\langle 0|$, and the result simply states that any Kraus operators for Φ must be balanced multiples of $|0\rangle\langle 0|$ and $|0\rangle\langle 1|$.

As a more intricate example in the most general (non-subspace, non-operator) case of a private code, we point out how the 2-qubit phase damping channel Λ can be viewed from the perspective of this result. The eigenstates of $\rho_0 = \frac{1}{4} I_4$ are $|00\rangle, |01\rangle, |10\rangle, |11\rangle$, each having eigenvalue $\frac{1}{4}$. For simplicity, we will use the standard orthonormal basis on the subsystem B: $\{\langle\psi_{B,i}|\} = \{\langle 0|, \langle 1|\}$. In our example, $\sigma_A = \frac{1}{2} I_2$, hence its eigenstates are $\{|\psi_{A,k}\rangle\} = \{|0\rangle, |1\rangle\}$, with corresponding eigenvalues $\frac{1}{2}$.

Using the Kraus operators $\{V_j\} = \{\frac{1}{2} II, \frac{1}{2} XX, \frac{1}{2} ZZ, -\frac{1}{2} YY\}$ of Λ', we compute $V_j|\psi_{A,k}\rangle$

as follows:

$$
\begin{aligned}
V_1|\psi_{A,1}\rangle &= \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} = \frac{1}{2}\left(|00\rangle\langle 0| + |01\rangle\langle 1|\right) \\
V_1|\psi_{A,2}\rangle &= \frac{1}{2}\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} = \frac{1}{2}\left(|10\rangle\langle 0| + |11\rangle\langle 1|\right) \\
V_2|\psi_{A,1}\rangle &= \frac{1}{2}\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} = \frac{1}{2}\left(|10\rangle\langle 1| + |11\rangle\langle 0|\right) \\
V_2|\psi_{A,2}\rangle &= \frac{1}{2}\begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} = \frac{1}{2}\left(|00\rangle\langle 1| + |01\rangle\langle 0|\right) \\
V_3|\psi_{A,1}\rangle &= \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & -1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} = \frac{1}{2}\left(|00\rangle\langle 0| - |01\rangle\langle 1|\right) \\
V_3|\psi_{A,2}\rangle &= \frac{1}{2}\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ -1 & 0 \\ 0 & 1 \end{pmatrix} = \frac{1}{2}\left(-|10\rangle\langle 0| + |11\rangle\langle 1|\right) \\
V_4|\psi_{A,1}\rangle &= \frac{1}{2}\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & -1 \\ 1 & 0 \end{pmatrix} = \frac{1}{2}\left(-|10\rangle\langle 1| + |11\rangle\langle 0|\right) \\
V_4|\psi_{A,2}\rangle &= \frac{1}{2}\begin{pmatrix} 0 & 1 \\ -1 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} = \frac{1}{2}\left(|00\rangle\langle 1| + |01\rangle\langle 0|\right).
\end{aligned}
$$

Note that the V_j are 4×2 matrices formed with 2×2 Pauli operators and zero blocks. Recall that we can consider both the left-hand and right-hand side of equation (4.19) as

quantum channels. Moreover, the Kraus operators $\{X_i\}, \{Y_j\}$ of equal quantum channels are related via $X_i = \sum_j \lambda_{ij} Y_j$ for some isometry $\lambda = (\lambda_{ij})$. When the number of Kraus operators X_i is equal to the number of Kraus operators Y_i, λ is unitary. In this case, $\sqrt{p_k} = \frac{1}{\sqrt{2}}$ for all k, and each V_j has a factor of $\frac{1}{2}$, so the coefficient of the left-side of this equation is always $\frac{1}{2\sqrt{2}}$. The coefficient of $\lambda_{ijkl}\sqrt{q_l}|\phi_l\rangle\langle\psi_{B,i}|$ is $\lambda_{ijkl}\sqrt{q_l} = \frac{1}{\sqrt{2}} \cdot \frac{1}{2}$ for all i, j, k, l.

Thus in our example, we find that λ is the following matrix:

$$\lambda = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

The scalar matrix $\lambda = (\lambda_{ijkl})$ is indeed an isometry. Furthermore, because the number of operators $\sum_m \sqrt{p_k} V_j |\psi_{A,k}\rangle|\psi_{B,m}\rangle\langle\psi_{B,m}|$ agrees with the number of operators $|\phi_l\rangle\langle\psi_{B,i}|$ (namely, 8), the matrix λ is in fact unitary.

4.2.3.1 Complementarity and quantum error correction

Several links have been made between quantum error correction and quantum privacy. In the case of operator private subsystems and operator error-correcting subsystems, the complementarity theorem of [90] discussed below established an algebraic bridge between the two subjects. This firmly links the operator quantum error correction theory to that of operator private subsystems—results in one field can immediately be exported to the other. Thus, it is natural to ask whether such a result holds in this more general setting. To answer this we need the concept of complementary channels.

As a consequence of the Stinespring dilation theorem, every channel Φ may be seen to arise from an environment Hilbert space E, a pure state $|\psi\rangle$ on the environment, and a unitary operator U on the composite $S \otimes E$ in the following sense: $\Phi(\rho) = \mathrm{Tr}_E\big(U(\rho \otimes |\psi\rangle\langle\psi|)U^\dagger\big)$. Tracing out the system instead yields a complementary channel: $\Phi^\sharp(\rho) = \mathrm{Tr}_S\big(U(\rho \otimes |\psi\rangle\langle\psi|)U^\dagger\big)$. The uniqueness built into the theorem yields a certain uniqueness for such a pair of channels, so that we talk of "the" complementary channel $\Phi^\sharp$ for a given channel Φ [70, 84].

We have already discussed operator private subsystems—the essential difference being that instead of a single state on A, it is demanded that Eq. (4.3) holds for all states on A. Similarly, an operator quantum error-correcting subsystem B for a channel $\mathcal{E}$ [91, 92] requires the existence of a correction operation $\mathcal{R}$ such that: $\forall \sigma_A \ \forall \sigma_B$, $\exists \tau_A$ for which $\mathcal{R} \circ \mathcal{E}(\sigma_A \otimes \sigma_B) = \tau_A \otimes \sigma_B$. The main result of [90] shows that B is private for Φ if and only if it is error-correcting for $\Phi^\sharp$.

Does this result extend to the more general setting? The Kraus operators of the complementary map $\Lambda^\sharp$ are four orthogonal rank-one projectors in two-qubit Hilbert space, and in particular the map determines a von Neumann measurement. No error-correcting subsystem can be extracted in such a setting; moreover, when the input space is restricted to be that of our example, the complementary map is private, as we saw in Section 4.2.2. Thus, not only does the complementarity result fail, it fails dramatically.

This discussion motivates the following observation: The notion of an operator quantum error-correcting subsystem can be expanded to mimic the general definition of a private quantum subsystem. Indeed, a revised definition analogous to that of Eq. (4.3) could be proposed as follows: B is *correctable* for $\mathcal{E}$ if there exists an operation $\mathcal{R}$ such that for all σ_B and some *fixed* states σ_A, τ_A, we have $\mathcal{R} \circ \mathcal{E}(\sigma_A \otimes \sigma_B) = \tau_A \otimes \sigma_B$. This is a potentially new notion of quantum error-correcting code. We show this is not the case in the following discussion, which focusses on the class of phase damping examples considered above. However, in the general discussion that follows, we show how a modified view of the associated dilations recaptures the complementarity result.

For our phase damping channel Λ, we can compute the Kraus operators of the complementary channel $\Lambda^\sharp$ by "stacking" the j-th column of each of the eight Kraus operators V_i of Λ one below the next, to obtain the j-th Kraus operator of $\Lambda^\sharp$:

$$A_1 = \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \quad A_2 = \frac{1}{2}\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}$$

$$A_3 = \frac{1}{2}\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 \end{pmatrix} \quad A_4 = \frac{1}{2}\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

We now ask what is the behaviour of the complementary channel on the subsystem B paired with the fixed state σ_A; that is, we compute how $\Phi^\sharp$ acts on operators $\sigma_A \otimes \sigma_B$ for all σ_B. Again, we must be careful: this pairing, which in this case we can identify with the algebra

$I_2 \otimes \mathbb{M}_2$, is private for Λ', and so we wish to test $I_2 \otimes \mathbb{M}_2$ on $(\Lambda^\sharp)'$, where we obtain $(\Lambda^\sharp)'$ by applying the unitary transformation $U(\cdot)U^\dagger$, with $U = CNOT_{1,2}CNOT_{2,1}(K^\dagger \otimes I_2)$, as before.

We compute the Kraus operators of $(\Lambda^\sharp)'$ to be $\{B_i = UA_iU^\dagger\}$, where

$$\begin{aligned}
B_1 &= \frac{1}{4}\begin{pmatrix} 1-i & 0 & 0 & 1-i \\ 1+i & 0 & 0 & 1+i \\ 1-i & 0 & 0 & 1-i \\ 1+i & 0 & 0 & 1+i \end{pmatrix} \\
B_2 &= \frac{1}{4}\begin{pmatrix} 0 & 1-i & 1-i & 0 \\ 0 & -1-i & -1-i & 0 \\ 0 & -1+i & -1+i & 0 \\ 0 & 1+i & 1+i & 0 \end{pmatrix} \\
B_3 &= \frac{1}{4}\begin{pmatrix} 1-i & 0 & 0 & -1+i \\ -1-i & 0 & 0 & 1+i \\ 1-i & 0 & 0 & -1+i \\ -1-i & 0 & 0 & 1+i \end{pmatrix} \\
B_4 &= \frac{1}{4}\begin{pmatrix} 0 & -1+i & 1-i & 0 \\ 0 & -1-i & 1+i & 0 \\ 0 & 1-i & -1+i & 0 \\ 0 & 1+i & -1-i & 0 \end{pmatrix}.
\end{aligned}$$

Now, for any $\sigma_B = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathcal{L}(B)$, we find

$$(\Lambda^\sharp)'\left(\frac{1}{2}I_A \otimes \sigma_B\right) = \sum_i B_i \left(\frac{1}{2}I_A \otimes \sigma_B\right) B_i^\dagger = \frac{1}{4}I_4.$$

Far from being correctable on the algebra $I_2 \otimes \mathbb{M}_2$, the complementary channel $(\Lambda^\sharp)'$ (with the proper unitary transformation) is completely depolarizing. All information is lost, so there is no possibility of the channel being correctable in any sense. In fact, note in this case that the Kraus operators of the complementary map $\Lambda^\sharp$ are four orthogonal rank-one projectors in two-qubit Hilbert space, and in particular the map determines a von Neumann measurement.

However, one can rightly ask if the dephasing map Λ and its complementary map are both private, where does the quantum information go? Figure 4.2 illustrates the isometric

extension of the dephasing channel, along with the encoding of the information from the algebra $I_2 \otimes \mathbb{M}_2$ to a state of the form of equation (4.17).

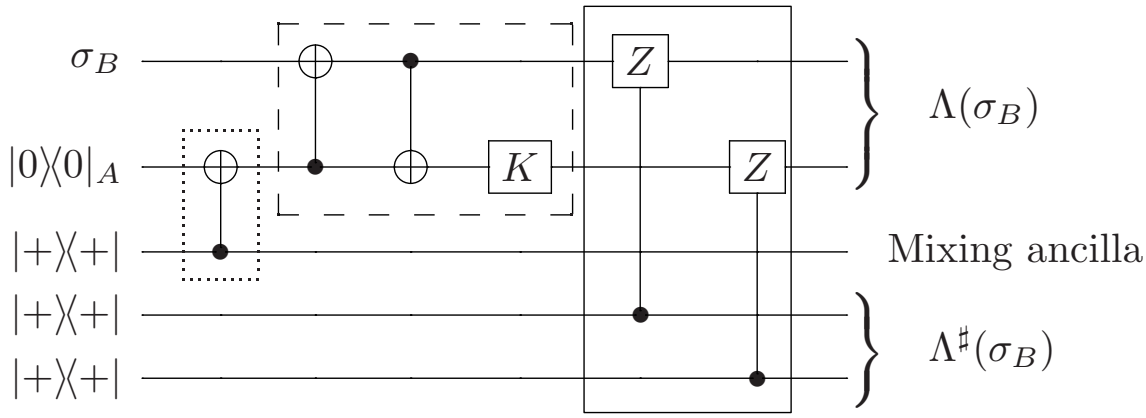


Figure 4.2: Isometric extension of the two-qubit dephasing channel $\Lambda = \Lambda_2 \circ \Lambda_1$. The extension of the channel to be a unitary transformation is given in the solid box (—) by introducing ancilla qubits 4 and 5. The dashed box (– –) contains the encoding of ρ of the form in equation (4.17) into the algebra $I_2 \otimes \mathbb{M}_2$. The dotted box $(\cdots)$ gives a particular preparation of the mixed state for the subsystem encoding, using a "mixing ancilla" that is traced out for both the channel Λ and its complementary channel $\Lambda^\sharp$.

As Figure 4.2 shows, the isometric extension of the channel to a larger Hilbert space, where the state evolution is described by unitary evolution, can be achieved using two extra qubits. Moreover, in order to purify the mixed state used in the subsystem encoding, one could use an additional "mixing ancilla". Such an ancilla would be traced out both for the dephasing channel Λ and its complementary channel $\Lambda^\sharp$. By definition of the unitary extension of the channel, the channel mapping $\Lambda(\rho)$ can be obtained by tracing out the final two qubits as well as the mixing ancilla. The complementary channel is obtained by tracing out qubits 1 and 2 as well as the mixing ancilla. As shown above, both of these outputs are private.

However, what if one had access to the information stored in the mixing ancilla? The role of this state is to twirl the second qubit to obtain a mixed state, however, if one now had access to this state the overall evolution of the channel is no longer on a subsystem encoding, but rather it would be on a subspace encoding that included the mixing ancilla

itself. Define U_Λ as the full unitary evolution described in Figure 4.2, and let $\tilde{\rho} = \rho \otimes |0\rangle\langle 0| \otimes |+\rangle\langle +| \otimes |0\rangle\langle 0| \otimes |0\rangle\langle 0|$ be the input state into the unitary evolution. The dephasing channel is given by $\mathrm{Tr}_{345}(U_\Lambda \tilde{\rho} U_\Lambda^\dagger)$, where we trace out qubits 3 (the mixing ancilla), 4, and 5. If we now look at the output on registers 345, the channel $\mathrm{Tr}_{12}(U_\Lambda \tilde{\rho} U_\Lambda^\dagger)$ must describe an operator quantum error-correcting code as it is the complement to a channel that is an operator private quantum channel. It is clear that this is not the complementary channel in the sense of the isometric extension, as we are adding on an extra level of operations, namely the mixing of the qubit $|0\rangle$ using a mixing ancilla (and then considering the mixing ancilla as qubit 1 of a larger Hilbert space). However, in this modified notion of the complementary channel we find the quantum information that was lost. We can thus conclude that the mixing ancilla plays an important role in the perseverance of global quantum information, and that the information must be stored in the correlations between this space and one of the two output spaces.

More generally, given a private subsystem $\mathcal{L}(B)$ for a channel Φ (with fixed mixed state $\sigma_A \in \mathcal{L}(A)$), one can formalize the notion of a correctable complementary channel in a similar fashion. Let $\sigma_A = \sum_{i=1}^N p_i |\psi_i\rangle\langle\psi_i|_A$ and define a mixing ancillary Hilbert space $\mathcal{L}(M)$ containing N basis states. The mixing ancillary space $\mathcal{L}(M)$, as in the example, is used to apply a controlled unitary operation U_i based on the state $|\varphi\rangle_A$ in $\mathcal{L}(A)$, and the unitaries U_i are chosen such that $U_i(|i\rangle_M |\varphi\rangle_A) = |i\rangle_M |\psi_i\rangle_A$. That is,

$$\begin{aligned}\sum_i p_i U_i |i\rangle_M |\varphi\rangle\langle\varphi|_A \langle i|_M U_i^\dagger &= \sum_i p_i |i\rangle_M |\psi_i\rangle\langle\psi_i|_A \langle i|_M \\ &= \sum_i p_i |i\rangle\langle i|_M \otimes |\psi_i\rangle\langle\psi_i|_A.\end{aligned}$$

Since $\sigma_A = \mathrm{Tr}_M\big(\sum_i p_i |i\rangle\langle i|_M \otimes |\psi_i\rangle\langle\psi_i|_A\big)$, we find

$$\begin{aligned}\sigma_A &= \mathrm{Tr}_M\big(\sum_i p_i U_i |i\rangle_M |\varphi\rangle\langle\varphi|_A \langle i|_M U_i^\dagger\big) \\ &= \mathrm{Tr}_M\big(U_{MA} |\Theta\rangle\langle\Theta|_M \otimes |\varphi\rangle\langle\varphi|_A U_{MA}^\dagger\big),\end{aligned}$$

where $|\Theta\rangle_M = \sum_i \sqrt{p_i} |i\rangle_M$ is a chosen pure state for the mixing ancilla such that the U_{MA} performs the appropriate unitary transformation $U_{MA} = \sum_i |i\rangle\langle i|_M \otimes |\psi_i\rangle\langle\varphi|_A$. The private

quantum subsystem channel can then be expressed as follows:

$$\begin{aligned}\Phi(\sigma_A \otimes \sigma_B) &= \sum_j A_j(\sigma_A \otimes \sigma_B)A_j^\dagger \\ &= \mathrm{Tr}_K\Big(U_\Phi(\sigma_A \otimes \sigma_B \otimes |\zeta\rangle\langle\zeta|_K)U_\Phi^\dagger\Big) \\ &= \mathrm{Tr}_{MK}\Big(U_\Phi U_{MA}(|\Theta\rangle\langle\Theta|_M \otimes |\varphi\rangle\langle\varphi|_A \\ &\qquad \otimes \sigma_B \otimes |\zeta\rangle\langle\zeta|_K)U_{MA}^\dagger U_\Phi^\dagger\Big). \end{aligned} \tag{4.20}$$

The transformation within the parenthesis is a unitary transformation, as U_Φ is a unitary defined by the isometric extension of the channel Φ (that is, by Stinespring's dilation theorem), where we have introduced the ancillary system K to form the isometric extension with $|\zeta\rangle_K$ being a fixed pure state. The unitary U_{MA} corresponds to the transformation in order to prepare a mixed state σ_A, after tracing out over the mixing ancillary space M. Since the transformation within the brackets is a unitary transformation, if the output state of the channel contains no information about the input state σ_B, the quantum information must be completely contained in the traced out subsystem: the MK subsystem. That is, if one traced out the output space, and we were left with the MK subsystem, such an output would necessarily be correctable since all quantum information is contained in that system. That is to say, the generalized conjugate channel,

$$\begin{aligned}\tilde{\Phi}(\sigma_A \otimes \sigma_B) &= \mathrm{Tr}_{A\otimes B}\Big(U_\Phi(\sigma_A \otimes \sigma_B \otimes |\zeta\rangle\langle\zeta|_K)U_\Phi^\dagger\Big) \\ &= \mathrm{Tr}_{AB}\Big(U_\Phi U_{MA}(|\Theta\rangle\langle\Theta|_M \otimes |\varphi\rangle\langle\varphi|_A \\ &\qquad \otimes \sigma_B \otimes |\zeta\rangle\langle\zeta|)U_{MA}^\dagger U_\Phi^\dagger\Big), \end{aligned} \tag{4.21}$$

has the feature that B is error-correctable for it. The generalized form of a private quantum subsystem can thus be summarized as a unitary transformation on an extended Hilbert space by the circuit in Figure 4.3.

If we consider the action of the private subsystem channel Φ via the isometric extension U_Φ, then the Kraus operators of the original channel can be expressed as follows (without the extension to the mixing ancilla space M):

$$\begin{aligned}\Phi(\sigma_B) &= \mathrm{Tr}_K\big(U_\Phi(\sigma_A \otimes \sigma_B \otimes |\zeta\rangle\langle\zeta|_K)U_\Phi^\dagger\big) \\ &= \sum_i \langle i|_K U_\Phi(\sigma_A \otimes \sigma_B \otimes |\zeta\rangle\langle\zeta|_K)U_\Phi^\dagger|i\rangle_K \\ &= \sum_i P_i(\sigma_A \otimes \sigma_B \otimes |\zeta\rangle\langle\zeta|_K)P_i^\dagger, \end{aligned}$$

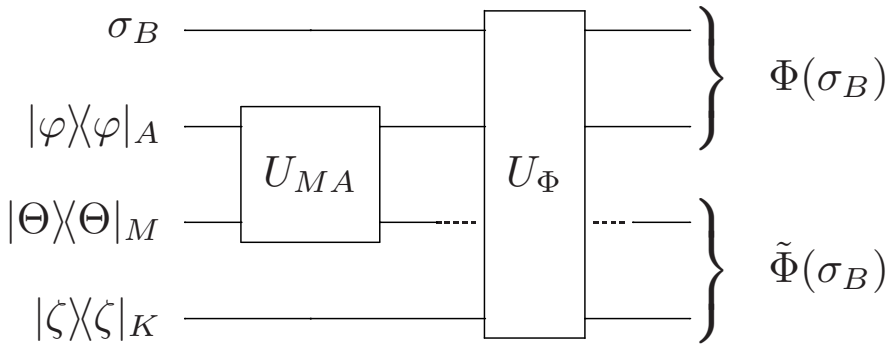


Figure 4.3: Generalized form of extending a private quantum subsystem channel to a unitary transformation via Stinespring's dilation theorem. The subsystem $\sigma_A \otimes \sigma_B$ that encodes the arbitrary state of quantum information σ_B is prepared by entangling an ancillary pure state mixing ancilla $|\Theta\rangle\langle\Theta|_M$ with a chosen pure state $|\varphi\rangle\langle\varphi|_A$ via the unitary U_{MA} and tracing out over the mixing ancilla space M. This operation corresponds to the dotted box in Figure 4.2. The action of the private quantum channel Φ can also be extended to a unitary transformation over a larger Hilbert space, as described by the action of U_Φ on systems ABK by introducing the ancillary state $|\zeta\rangle\langle\zeta|_K$, as described in Eq. 4.20. The unitary transformation U_Φ corresponds to the dashed and solid boxes in Figure 4.2. The complementary channel is defined on the output space of the extension of the channel Φ, and therefore corresponds to the final state on system K, yet in general will not be quantum error correctable for an arbitrary subsystem channel. However, a generalized conjugate channel $\tilde{\Phi}$ can be constructed on the Hilbert space MK, as described in Eq. 4.21, which will necessarily be a quantum error correctable channel since the overall extension is a subspace channel.

where the Kraus operators of the channel Φ are given by $\{P_i = \langle i|_K U_\Phi\}_i$. In a similar manner, the Kraus operators of the complementary channel $\Phi^\sharp$ are given as follows:

$$\begin{aligned}\Phi^\sharp(\sigma_B) &= \mathrm{Tr}_{AB}\big(U_\Phi(\sigma_A \otimes \sigma_B \otimes |\zeta\rangle\langle\zeta|_K)U_\Phi^\dagger\big) \\ &= \sum_j Q_j(\sigma_A \otimes \sigma_B \otimes |\zeta\rangle\langle\zeta|_K)Q_j^\dagger,\end{aligned}$$

where the Kraus operators are given by $\{Q_j = \langle j|_{AB} U_\Phi\}_j$. Finally, in order to extend the input space to be a subspace, rather than a subsystem, the ancillary mixing state is introduced. Defining the generalized complementary channel $\tilde{\Phi}$ as above, the Kraus operators of this channel can be defined on the extended Hilbert space as follows:

$$\begin{aligned}\tilde{\Phi}(\sigma_B) &= \mathrm{Tr}_{AB}\big(U_\Phi U_{MA}(|\Theta\rangle\langle\Theta|_M \otimes |\varphi\rangle\langle\varphi|_A \\ &\qquad\qquad \otimes \sigma_B \otimes |\zeta\rangle\langle\zeta|_K)U_{MA}^\dagger U_\Phi^\dagger\big) \\ &= \sum_k \langle k|_{AB} U_\Phi U_{MA}(|\Theta\rangle\langle\Theta|_M \otimes |\varphi\rangle\langle\varphi|_A \\ &\qquad\qquad \otimes \sigma_B \otimes |\zeta\rangle\langle\zeta|_K)U_{MA}^\dagger U_\Phi^\dagger |k\rangle_{AB} \\ &= \sum_k (I_M \otimes Q_k)U_{MA}(|\Theta\rangle\langle\Theta|_M \otimes |\varphi\rangle\langle\varphi|_A \\ &\qquad\qquad \otimes \sigma_B \otimes |\zeta\rangle\langle\zeta|_K)U_{MA}^\dagger (I_M \otimes Q_k)^\dagger \\ &= \sum_k R_k(|\Theta\rangle\langle\Theta|_M \otimes |\varphi\rangle\langle\varphi|_A \\ &\qquad\qquad \otimes \sigma_B \otimes |\zeta\rangle\langle\zeta|_K)R_k^\dagger,\end{aligned}$$

where the Kraus operators of the generalized complementary channel $\tilde{\Phi}$ mapping to the MK Hilbert space are related to the Kraus operators of the complementary channel $\Phi^\sharp$ via the relationship $\{R_k = (I_M \otimes Q_k)U_{MA}\}_k$. As outlined above, this channel must be quantum error correcting, and as such must satisfy the Knill-Laflamme conditions [88]:

$$\begin{aligned}\langle i|_B R_p^\dagger R_q |j\rangle_B &= \langle i|_B U_{MA}^\dagger (I_K \otimes Q_p^\dagger)(I_K \otimes Q_q)U_{MA}|j\rangle_B \\ &= \langle i_{U_{MA}}|_{MA}(I_K \otimes Q_p^\dagger)(I_K \otimes Q_q)|j_{U_{MA}}\rangle_{MA} \\ &= \delta_{ij} c_{pq}.\end{aligned}$$

For any generalized private subsystem channel there must be the existence of a higher dimensional Hilbert space such that the above Knill-Laflamme conditions for quantum error correcting hold for a set of Kraus operators related to the Kraus operators of the complementary channel of the original private subsystem channel.

4.2.3.2 Generalized channels on subspace and subsystem encodings

A common theme throughout this work has been that encoding into a subsystem, rather than a subspace, generates an increased freedom in the types of channels that can be used to privatize quantum information. In this subsection we explore this notion further, explicitly showing that the set of unitaries that can be used to privatize quantum information in a subsystem code are inherently richer than those for subspace codes. We shall focus on the case of encoding a single qubit of information into either a two-qubit subspace or a two-qubit subsystem.

Consider an arbitrary encoding of a single qubit into a two-qubit subspace:

$$(\alpha|0\rangle + \beta|1\rangle)|0\rangle \rightarrow \alpha|0_L\rangle_{12} + \beta|1_L\rangle_{12},$$

where $|0_L\rangle$ and $|1_L\rangle$ represent the logically encoded states in a higher dimensional Hilbert space. An arbitrary CPTP map can be described as a transformation of the encoded basis states to a larger dimensional Hilbert space, after which a trace is taken in the environment. Any arbitrary transformation can be described as follows:

$$|0_L\rangle_{12} \rightarrow \sum_{ij} |ij\rangle_{12}\big|E^0_{ij}\big\rangle_E$$
$$|1_L\rangle_{12} \rightarrow \sum_{ij} |ij\rangle_{12}\big|E^1_{ij}\big\rangle_E,$$

where the states $\big|E^m_{ij}\big\rangle$ are arbitrary environment states, that are not necessarily normalized or orthogonal. The environment states are the states on the ancillary qubits when expressing the final state in the computational basis of the first two qubits. Tracing out over the environment states, the resulting entries of the two-qubit mixed states have a particular form:

$$\begin{aligned} &|ij\rangle\langle kl|_{12}\mathrm{Tr}_E\big(\alpha\big|E^0_{ij}\big\rangle + \beta\big|E^1_{ij}\big\rangle\big)(\alpha^*\big\langle E^0_{kl}\big| + \beta^*\big\langle E^1_{kl}\big|\big) \\ =&|ij\rangle\langle kl|_{12}\mathrm{Tr}_E\big(|\alpha|^2|E^0_{ij}\rangle\langle E^0_{kl}| + \alpha\beta^*|E^0_{ij}\rangle\langle E^1_{kl}| \\ &\qquad + \alpha^*\beta|E^1_{ij}\rangle\langle E^0_{kl}| + |\beta|^2|E^1_{ij}\rangle\langle E^1_{kl}|\big), \end{aligned} \tag{4.22}$$

this imposes a set of conditions on the environmental states in order for the output on the first two qubits to be private, namely the terms after tracing out can yield no information about the input state as described by α and β.

Consider now the same isometric extension mapping along with the inclusion of a third qubit that will serve as a mixing ancilla. The encoding operation is now generalized to a

three qubit encoding, which upon tracing out the mixing ancilla will return the subsystem encoding on qubits 1 and 2. The generalized mapping is modified to include the third qubit.

$$|0_L\rangle_{123} \to \sum_{ij} |ijk\rangle_{123} |E^0_{ijk}\rangle_E$$
$$|1_L\rangle_{123} \to \sum_{ij} |ijk\rangle_{123} |E^1_{ijk}\rangle_E.$$

The generalized form of the mixed state entries on the first two qubits thus have the form

$$\begin{aligned}
&|ij\rangle\langle kl|_{12} \mathrm{Tr}_{3E}\Big(\big(\sum_p \alpha |p\rangle_3 |E^0_{ijp}\rangle + \beta |p\rangle_3 |E^1_{ijp}\rangle \big) \\
&\qquad \big(\sum_q \alpha^* \langle q|_3 \langle E^0_{klq}| + \beta^* \langle q|_3 \langle E^1_{klq}| \big) \Big) \\
=&|ij\rangle\langle kl|_{12} \mathrm{Tr}_E \Big(|\alpha|^2 (|E^0_{ij0}\rangle\langle E^0_{kl0}| + |E^0_{ij1}\rangle\langle E^0_{kl1}|) \\
&\qquad + \alpha\beta^* (|E^0_{ij0}\rangle\langle E^1_{kl0}| + |E^0_{ij1}\rangle\langle E^1_{kl1}|) \\
&\qquad + \alpha^*\beta (|E^1_{ij0}\rangle\langle E^0_{kl0}| + |E^1_{ij1}\rangle\langle E^0_{kl1}|) \\
&\qquad + |\beta|^2 (|E^1_{ij0}\rangle\langle E^1_{kl0}| + |E^1_{ij1}\rangle\langle E^1_{kl1}|) \Big),
\end{aligned} \tag{4.23}$$

therefore, by comparing equations 4.22 and 4.23, we find that in the case where a mixing ancilla has been introduced the set of conditions upon privatizing the output on the first two qubits is looser in terms of the environment states. Namely there is a freedom in choosing the environment states such that certain terms can cancel out to yield no information; this freedom does not exist in the case of a pure state encoding. We explore these set of conditions in more detail in Appendix A.

4.2.4 Quantum error correction revisited

In this subsection, we revisit the notion of an operator quantum error correctable subsystem, as given in Definition 57, and its parallels to private quantum subsystems. The discussion of private subsystem versus operator private subsystem in this work motivates the following observation: The notion of an operator quantum error-correcting subsystem can be expanded to mimic the general definition of a private quantum subsystem. We propose the following definition, which can be seen as the QEC analogue of equation (4.3):

Definition 65. *Let $S = (A \otimes B) \oplus (A \otimes B)^{\perp}$ and let $\mathcal{E}$ be a channel acting on $\mathcal{L}(S)$. Then B is a* generalized operator quantum error correcting code (GenOQECC) *for $\mathcal{E}$ if there exists a quantum channel $\mathcal{R}$ for which there exists a fixed state σ_A and a state $\tau_A = \tau_A(\sigma_A)$ (dependent on σ_A) such that for all σ_B, we have*

$$\mathcal{R} \circ \mathcal{E}(\sigma_A \otimes \sigma_B) = \tau_A \otimes \sigma_B. \tag{4.24}$$

Clearly no generality is lost in this definition by setting $\tau_A = \sigma_A$.

The key difference in this generalized definition as opposed to Definition 57 is that the state σ_A is fixed, as opposed to having to hold for all σ_A.

Example 66. Consider the following example of a generalized operator quantum error correcting code. Let

$$\sigma_A = (1 - 4p)|0000\rangle\langle 0000| + p \sum_{\mathrm{wt}(x)=1} |x\rangle\langle x|$$

be the fixed ancilla state, a mixed 4-qubit state, where the states $|x\rangle$ are the set of (four) computational basis states with Hamming weight 1. The weighting p can be thought of as a probability of failure of preparing a desired ground state $|0\rangle$ for the purpose of error correction, where we have omitted higher order p terms. Let σ_B be any single qubit state. The encoding of the subsystem code is a controlled operation from qubit B which targets all qubits of the state σ_A with a controlled-X, we shall call such an encoding operation U_{AB}. The error map will be the probabilistic application of an X error on any of the 5 encoded qubits given by the set of Kraus operators $\{\sqrt{\epsilon_i} X_i\}_{i=0}^{5}$, where ϵ_i is the probability of the error X_i occurring (X_0 denoting the identity operation). The application of such an error map will produce the following mapping on the encoded state for an arbitrary B state

$|\psi\rangle_B = \alpha|0\rangle + \beta|1\rangle$,

$$
\begin{aligned}
&\mathcal{E}\big(U_{AB}(\sigma_A \otimes |\psi\rangle\langle\psi|_B)U_{AB}^\dagger\big) \\
&= \mathcal{E}\Big(|\alpha|^2\big((1-4p)|00000\rangle\langle 00000| + p\sum_{\mathrm{wt}(x)=1}|x0\rangle\langle x0|\big) \\
&\quad + \alpha\beta^*\big((1-4p)|00000\rangle\langle 11111| + p\sum_{\mathrm{wt}(x)=1}|x0\rangle\langle \overline{x}1|\big) \\
&\quad + \beta\alpha^*\big((1-4p)|11111\rangle\langle 00000| + p\sum_{\mathrm{wt}(x)=1}|\overline{x}1\rangle\langle x0|\big) \\
&\quad + |\beta|^2\big((1-4p)|11111\rangle\langle 11111| + p\sum_{\mathrm{wt}(x)=1}|\overline{x}1\rangle\langle \overline{x}1|\big)\Big) \\
&= \sum_{i=0}^{5}\Big(|\alpha|^2\big((1-4p)X_i|0\rangle\langle 0|^{\otimes 5}X_i \\
&\quad + p\sum_{\mathrm{wt}(x)=1}X_i|x0\rangle\langle x0|X_i\big) \\
&\quad + \alpha\beta^*\big((1-4p)X_i|0\rangle\langle 1|^{\otimes 5}X_i + p\sum_{\mathrm{wt}(x)=1}X_i|0\rangle\langle \overline{x}1|X_i\big) \\
&\quad + \beta\alpha^*\big((1-4p)X_i|1\rangle\langle 0|^{\otimes 5}X_i + p\sum_{\mathrm{wt}(x)=1}X_i|\overline{x}1\rangle\langle x0|X_i\big) \\
&\quad + |\beta|^2\big((1-4p)X_i|1\rangle\langle 1|^{\otimes 5}X_i + p\sum_{\mathrm{wt}(x)=1}X_i|\overline{x}1\rangle\langle \overline{x}1|X_i\big)\Big),
\end{aligned}
$$

where we have defined $|\overline{x}\rangle = X^{\otimes 5}|x\rangle$. One can notice that the error map will flip at most one bit. This is important as the encoded $|0\rangle$ terms have weight 0 or 1 for all terms, while the encoded $|1\rangle$ have weight 4 or 5. This means that after the application of the error map, the encoded $|0\rangle$ will have a weight between 0 and 2, while the encoded $|1\rangle$ will have weight between 3 and 5. The recovery operation will then perform a weight check using measurement in the computational basis, associating all states with weight ≤ 2 to an encoded $|0\rangle$ and all states with weight ≥ 3 to an encoded $|1\rangle$ state. As such all X_i errors are corrected. Since this error correction procedure works for an arbitrary pure state encoding of σ_B, it will necessarily work for the full set of states in $\mathcal{L}(B)$. That is, B is a generalized operator quantum error correcting code for $\mathcal{E}$.

It is worth noting that the error correction procedure does not work if we chose the ancillary mixed state to be outside the set of states of weight 0 or 1. Consider a particular

example of a 4-qubit state of weight 2, given by $\sigma_A = |1100\rangle\langle 1100|_A$. We shall show that encoding using such an ancillary state will not correct the error map for a particular choice of $|\psi\rangle\langle\psi|_B = |0\rangle\langle 0|_B$. The action of the error map is as follows:

$$\begin{aligned}
&\mathcal{R} \circ \mathcal{E}\big(U_{AB}|11000\rangle\langle 11000|U_{AB}^\dagger\big) \\
&= \mathcal{R} \circ \mathcal{E}\big(|11000\rangle\langle 11000|\big) \\
&= \mathcal{R}\big(\sum_{i=0}^{5} \epsilon_i X_i |11000\rangle\langle 11000| X_i\big) \\
&= (\epsilon_0 + \epsilon_1 + \epsilon_2)\tau_{A,0} \otimes |0\rangle\langle 0|_B + (\epsilon_3 + \epsilon_4 + \epsilon_5)\tau_{A,1} \otimes |1\rangle\langle 1|.
\end{aligned}$$

The recovery operator maps all states that correspond to either no error or errors on the first 2 qubits to the correct state $|0\rangle\langle 0|$, this is since the action of the error map returns a state of weight 1 or 2. However, for an error that occurs on qubits 3 through 5, the state state before the action of the recovery operator is now of weight 3, which will then be mapped to the state $|1\rangle\langle 1|$ by the definition of the action of recovery operator. Thus, as long as there is a non-zero probability of an error on the last 3 qubits, the action of the error map will result in the recovery of an incorrect state, as the state does not have the form $\tau_A \otimes \sigma_B$. Similarly, for any choice of ancillary state σ_A of weight greater or equal to 2 there will exist a state $|\psi\rangle\langle\psi|$ that will result in faulty error correction.

Therefore, we know that the GenOQECC corrects for the error map $\{\sqrt{\epsilon_i}X_i\}_{i=0}^5$ for the given fixed state, and will not be error correcting for ancillary states with weight greater or equal to 2. However, it is worth noting that as long as only one of the qubits has a preparation error (therefore weight 1), the value of p does not matter. Thus, the channel $\mathcal{E}$ is correctable for all σ_B and for all σ_A of weight 0 or 1; but, it is not correctable for any arbitrary $\sigma_A \in \mathcal{L}(A)$ (that is any 4-qubit state in the ancillary space). Hence, if we consider the full 4-qubit ancillary Hilbert space $\mathcal{L}(A)$, we will not have an operator quantum error correcting code (OQECC) on such a space.

This example shows that a GenOQEC code may not be an OQEC code for a given error map. Nevertheless, the following result shows that whenever a GenOQECC exists, we can still find an OQECC for the map of the same dimension. To find such a code we must consider the ancilla A more carefully.

Theorem 67. *Given a decomposition $S = (A \otimes B) \oplus (A \otimes B)^\perp$ and channel $\mathcal{E}$ on $\mathcal{L}(S)$, suppose there exists σ_A and channel $\mathcal{R}$ on $\mathcal{L}(S)$ such that for all σ_B,*

$$\mathcal{R} \circ \mathcal{E}(\sigma_A \otimes \sigma_B) = \sigma_A \otimes \sigma_B.$$

Then there exists $|\alpha\rangle \in A$ and channel $\mathcal{R}_\alpha$ on $\mathcal{L}(S)$ such that for all σ_B,

$$\mathcal{R}_\alpha \circ \mathcal{E}(|\alpha\rangle\langle\alpha| \otimes \sigma_B) = |\alpha\rangle\langle\alpha| \otimes \sigma_B;$$

in other words, the subspace $\alpha \otimes B$ is an error-correcting code for $\mathcal{E}$.

Proof. First let $|\psi\rangle \in B$ and put $P = |\psi\rangle\langle\psi|$. Let $\{|\alpha_k\rangle\}$ be the normalized eigenvectors of σ_A so that $\sigma_A = \sum_{k=1}^m p_k |\alpha_k\rangle\langle\alpha_k|$ where $0 < p_k \leq 1$. By assumption and using the positivity of $\mathcal{R} \circ \mathcal{E}$ we have for all k,

$$\begin{aligned} 0 &\leq \mathcal{R} \circ \mathcal{E}(p_k|\alpha_k\rangle\langle\alpha_k| \otimes P) \\ &= p_k \mathcal{R} \circ \mathcal{E}(|\alpha_k\rangle\langle\alpha_k| \otimes P) \\ &\leq \mathcal{R} \circ \mathcal{E}(\sigma_A \otimes P) \\ &= \sigma_A \otimes P \\ &= (I_A \otimes P)(\sigma_A \otimes P)(I_A \otimes P). \end{aligned}$$

It follows that there are positive operators $\sigma_{\psi,k} \in \mathcal{L}(A)$ such that $\mathcal{R} \circ \mathcal{E}(p_k|\alpha_k\rangle\langle\alpha_k| \otimes P) = \sigma_{\psi,k} \otimes P$ for all k. We can trace-normalize to write $\mathcal{R} \circ \mathcal{E}(|\alpha_k\rangle\langle\alpha_k| \otimes P) = \sigma_{\psi,k} \otimes P$ for all k, where $\sigma_{\psi,k}$ are now density operators.

In fact, the operators $\sigma_{\psi,k}$ do not depend on $|\psi\rangle$. To verify this claim, for brevity we shall assume $\dim B = 2$. The case of general B easily follows. So let $|\psi_i\rangle$, $i = 1, 2$, be an orthonormal basis for B. Let $P_i = |\psi_i\rangle\langle\psi_i|$, $i = 1, 2$, and put $P_\pm = |\pm\rangle\langle\pm|$ where $|\pm\rangle = \frac{1}{\sqrt{2}}(|\psi_1\rangle \pm |\psi_2\rangle)$. Fix $\alpha = \alpha_k$. By the above argument, there are operators $\sigma_{\pm,\alpha}$ and $\sigma_{i,\alpha}$ on A such that

$$\begin{aligned} &\mathcal{R} \circ \mathcal{E}(|\alpha\rangle\langle\alpha| \otimes P_\pm) = \sigma_{\pm,\alpha} \otimes P_\pm \\ \text{and} \qquad &\mathcal{R} \circ \mathcal{E}(|\alpha\rangle\langle\alpha| \otimes P_i) = \sigma_{i,\alpha} \otimes P_i. \end{aligned}$$

In particular, as $I_B = P_+ + P_- = P_1 + P_2$, we have

$$\begin{aligned} \mathcal{E}(|\alpha\rangle\langle\alpha| \otimes I_B) &= \sigma_{1,\alpha} \otimes P_1 + \sigma_{2,\alpha} \otimes P_2 \\ &= \sigma_{+,\alpha} \otimes P_+ + \sigma_{-,\alpha} \otimes P_-. \end{aligned}$$

If we compress this equation by the projection $I_A \otimes P_1$, we obtain

$$\begin{aligned} &(I_A \otimes P_1)\mathcal{E}(|\alpha\rangle\langle\alpha| \otimes I_B)(I_A \otimes P_1) = \sigma_{1,\alpha} \otimes P_1 \\ &\quad = \frac{1}{2}(\sigma_{+,\alpha} + \sigma_{-,\alpha}) \otimes P_1. \end{aligned}$$

Thus, $\sigma_{1,\alpha} = \frac{1}{2}(\sigma_{+,\alpha} + \sigma_{-,\alpha})$ and since the same identity holds for $\sigma_{2,\alpha}$ when we compress by $I_A \otimes P_2$, we obtain $\sigma_{1,\alpha} = \sigma_{2,\alpha}$. There is nothing particularly special about our use of $|\pm\rangle$ here, and in fact this argument may be applied to show the same operator is obtained for any pure state on A.

The proof is now completed by a simple linearity argument. Indeed, write $\sigma_k := \sigma_{\psi,k}$, so we have $\mathcal{R} \circ \mathcal{E}(|\alpha_k\rangle\langle\alpha_k| \otimes P_\psi) = \sigma_k \otimes P_\psi$, and by linearity P_ψ can be replaced by an arbitrary σ_B. We may then choose a channel $\mathcal{R}_k$ such that $\mathcal{R}_k(\sigma_k \otimes \sigma_B) = |\alpha_k\rangle\langle\alpha_k| \otimes \sigma_B$ for all σ_B. It follows that $|\alpha_k\rangle \otimes B$ is correctable for $\mathcal{E}$, with a recovery operation given by $\mathcal{R}_k \circ \mathcal{R}$.

□

We note that the above argument can be adjusted to show that in fact any eigenspace A' for σ_A determines an OQEC code (which will be a subsystem when $\dim A' > 1$) for the error map of the same size, via the pairing A' and B.

4.2.5 Conclusion & outlook

Private quantum subsystems are subsystem encodings of quantum information that are privatized under the action of a given channel. In this work, we analyzed the development of private subsystems for the special case given by the composition of phase damping channels on many qubit Hilbert spaces. While each individual channel of this form is not private, the composition of such channels were shown to contain a private single qubit subsystem. Yet, for such channels, and for a wide class of more general channels, no private subspace or operator private subsystem exists. Moreover, we discussed how the channel fails to have the corresponding complementary error-correctable pair as in the case of operator subsystems. We have added to our analysis showing that the multi-qubit dephasing channel has a private subsystem without exhibiting a private subspace. We explicitly showed that this private subsystem is not operator private, which is the first such example we are aware of. Additionally, we have provided a set of testable algebraic conditions for private quantum subsystems, expanding the discussion of examples and providing further results for particular forms of the channels and output states.

One of the surprising structural aspects of the most general private quantum subsystem channels (in contrast to operator private subsystems) is that their complementary channel, obtained through the Stinespring dilation, is no longer necessarily correctable on the subsystem, and can in fact be private. In this work, we have given an analysis and discussion of where the quantum information is leaking to in such a setting by dilating to an even

higher dimensional Hilbert space than is required by the usual notion of complementary channels.

Finally, the difference between general private subsystems and operator private subsystems presented suggests there could be an analogous notion of generalized operator quantum error correction. In this work we provided an explicit definition for these codes and showed that the existence of such a code implies the existence of a standard (subspace) QEC code of the same size, determined by the fixed ancillary state used in the generalized code. Thus, the generalized notion does not lead to larger codes than what can be found in standard QEC. However, the same is true when one compares operator QEC to standard QEC; indeed, this is even obvious from the definitions of the two code types in that case. What generated significant interest in OQEC codes beyond the theoretical appeal of the mathematical framework, was that it turned out such codes can have extra features that make them quite valuable for fault tolerant quantum computing. It would be interesting to know whether generalized QEC codes have similar advantages. A next step in the analysis would be to broaden the set of generalized code examples which are neither subspaces nor operator subsystems. We have further expanded upon the mathematical structure of the codes by connecting the theory of private subsystems to quasiorthogonal algebras, this work is presented in Appendix B.

Chapter 5

Practical quantum architectures: qRAM

The preceding chapters have focused mainly on the construction and structure of quantum error correcting codes. Now we will focus on the architectural components of a quantum computer as they span multiple avenues, not just quantum state manipulation and direct computation. One of the primary elements of classical computers is a Random Access Memory (RAM) which serves as a quick lookup table for live computations. The main idea of the RAM is to convert information from a computation into path information to point to particular storage locations, revealing or utilizing the information that is stored therein. As will be covered throughout this chapter, the purpose of a quantum RAM (or qRAM) is similar to that of its classical counterpart, that is to convert the information from a quantum state and couple that information to the classical information that is stored in some classical memory. This process is essential to the running many important quantum algorithm primitives, such as Grover's search, where the state of the computation (that is in a superposition) is coupled to the classical memory by entangling the state of the quantum information to that of the classical memory. Therefore, it is very important that the quantum information is transferred into path information, possibly in superposition, and that this path information is not known to the external observer, in order to avoid decoherence.

The goal of the project presented in this chapter was to investigate the bucket brigade qRAM model proposed in Refs. [57, 56]. In these works, the authors argued that since a memory of size 2^n can be accessed using a tree-like path structure with depth n, the total number of operations that would be needed to access the memory would scale polynomially with n. It was argued that due to this scaling, the error rate of the physical operators would

only have to be inversely proportional to poly(n), therefore potentially negating the need for quantum error correction. This chapter addresses this claim under simple, realistic noise, and concludes that errors are more detrimental than initially expected and would require exponentially low failure rate in order to guarantee success for the implementation of the qRAM for Grover's search.

5.1 qRAM: converting state to path information

This section covers research that was published in Ref. [9], copyrighted by IOP Publishing Ltd and Deutsche Physikalische Gesellschaft. Michele Mosca provided the motivation for this project, all authors contributed to the development of the ideas, analysis of the results, and manuscript writing. Vlad Gheorghiu and myself did the majority of the editing for publication.

5.1.1 Introduction

The typical physical implementation of the addressing mechanism uses the fanout architecture [74, 124], in which the routing scheme corresponds to a binary tree. Each node consists of a pair of transistors which routes the electronic signal down one of the two paths to the subsequent level. In the fanout architecture, a given level has all nodes sharing the same routing direction (left or right), set by the corresponding address bit. An n bit query string determines a unique path in the binary tree, corresponding to the desired memory location. In the process, $\mathcal{O}(2^n)$ transistors are activated.

Alternative routing schemes with $\mathcal{O}(\text{poly}(n))$ activated transistors have been proposed, corresponding to exponentially lower energy consumption. One such example is the "bucket brigade" scheme [57, 56]. However, most of the classical implementations follow the simpler fanout architecture, as the power consumption of RAM is negligible in comparison with the power consumption of other components in the architecture.

The classical RAM addressing scheme can be generalized to a quantum RAM (which we simply call qRAM from here on) scheme, where the input is a quantum state, the routing components are inherently quantum, and the information stored can be either classical, i.e. $|0\rangle$ or $|1\rangle$ but not a superposition of both, or quantum, i.e. any arbitrary superposition of $|0\rangle$ and $|1\rangle$. In the present paper we consider qRAM that stores only classical information.

Such memory allows querying superposition of addresses

$$\sum_j \alpha_j |j\rangle|0\rangle \xrightarrow{qRAM} \sum_j \alpha_j |j\rangle|m_j\rangle, \tag{5.1}$$

where $\sum_j \alpha_j |j\rangle$ is a superposition of queried addresses and $|m_j\rangle$ represents the content of the j-th memory location. A memory that stores classical information but allows queries in superposition is required for quantum algorithms such as Grover's search on a classical database [62], collision finding [29], element distinctness [4], dihedral hidden subgroup problem [94], quantum matrix inversion [68, 138], quantum machine learning [101, 121, 102, 100] and various practical applications mentioned in [8]. In fact, such a quantum memory plays the role of the oracle and is ideal in implementing any oracle-based quantum algorithm, in which the oracle is used to query classical data in superposition. It is important to distinguish between algorithms such as quantum matrix inversion [68, 138] or quantum machine learning [101, 121, 102, 100] that only require a number of queries polynomial in n, and those such as quantum searching [62] that require a number of queries super-polynomial in n. In the former case, as will be seen, the maximum qRAM gate error rate tolerated by the algorithms scales polynomially in n, and qRAM quantum error correction may not be required. In the latter case, which is the one that we concentrate on in this paper, the maximum tolerable qRAM error rate scales super-polynomially in n and quantum error correction is needed.

A conceptually simple physical implementation of a qRAM corresponds to a direct generalization of the fanout architecture used in classical RAMs. However, the number of faulty components that can be tolerated by the quantum architecture is of prime importance due to the difficulty in maintaining quantum coherence. This motivates searching for schemes with fewer faulty components. A fundamental assumption of the qRAM architecture is that "active" gates[1] are the only ones with significant errors.

In this project we investigate the bucket brigade qRAM proposal introduced in [57, 56]. Assuming one requires a constant error probability for the oracle query, then with the

[1]The concept of "active" gates introduced in [57, 56] is somewhat unnatural when extended to quantum gates. At the physical level, a gate is considered active if it physically acts on its input. Since the qRAM may be in a superposition of querying many (or all) possible bit values in the memory, every gate may be in a superposition of being active or not. Implicitly, there is a physical process that is checking whether each gate is active, and then acting in that case, and such a process will not be perfect in practice. Translated into the circuit model, such gates may be modelled as controlled-gates, i.e. gates that act on its input provided that the control qubit is set to $|1\rangle$. Therefore, such a gate is considered "active" if its control is set to $|1\rangle$ and "non-active" otherwise. In practice, even non-active gates will be prone to errors. The implicit assumption is that these errors are much smaller than the errors in active gates, and the focus of the bucket brigade models is to reduce the impact of the higher order errors found in the active part of a gate.

bucket brigade error model it suffices to have an error rate that is on the order of $\mathcal{O}(1/n^2)$. In the bucket brigade model, one assumes that each computational path only contains $\mathcal{O}(n)$ components that are faulty, and that a total of $\mathcal{O}(n^2)$ faulty operations are performed. One can argue that it is optimistic to assume that the so-called "non-active" components will be completely error-free. And, one could counter-argue that the error rates will be much lower, and thus ignored for problem instances of appropriate size. For the purposes of this article, we set aside these concerns and accept the premise of there only being $\mathcal{O}(n)$ faulty components.

In contrast to such a qRAM, if one just used a regular fanout circuit for the lookup, with no error correction, one would need to maintain quantum coherence over an exponential number of components [56]. In order to achieve a constant error rate for the query in this case, one would need to implement a fault-tolerant version of the look-up circuit, which would normally incur an overhead that is polynomial in n. One advantage of bucket brigade qRAM is thus to bypass the poly-log overhead of fault tolerant quantum error correction needed to achieve a constant error rate for a look-up. Such an error rate would be sufficient if the qRAM is used in an algorithm making a constant number of queries, for example, for certain state generation algorithms [106, 63]. In general, for an algorithm with inverse polynomially many queries, it would suffice to reduce the query error rate to be inverse polynomial in n, e.g. [101, 39].

In this work, we firstly shed doubt on the usefulness of a qRAM that provides queries with constant probability of error, when used with algorithms such as quantum searching that make super-polynomially many oracle queries. As an aside, we note that if the imperfect query operation is assumed to be unitary, and if one can apply the inverse of this imperfect query, then one can apply simple amplification methods to achieve queries with arbitrarily small error δ using a number of repetitions that is proportional to $\log(1/\delta)$. It was shown that this logarithmic overhead is not necessary for quantum searching [73] and other problems [35]. However, there is no reason to expect the errors in a realistic qRAM to behave this way, and in this article we consider incoherent errors.

We first show that a very simple model of incoherent physical errors induces an overall query error similar to the one described by Regev and Schiff [122]. Consequently, a qRAM that produces queries with constant error will not permit the quadratic speed-up in Grover's search algorithm [62] or any other quantum search algorithm one might design. We show that one cannot escape achieving an error rate that is super-polynomially small. We conjecture that this error model nullifies the asymptotic speed-ups of other quantum query algorithms as well, and leave as open questions the extension of this result to other important query problems.

This negative result implies the need for some means of error reduction for the qRAM, with a look-up error rate exponential in n. For consistency we assume a physical error rate that is inverse polynomial in n, the logarithm of the size of the database. We thus explore a natural approach, using quantum error correcting codes, to provide this error reduction, and argue that the apparent advantage of qRAM disappears in this case; in principle, one can make the error rate arbitrarily small, however the advantage of a small number of activated gates in the bucket brigade architecture appears to be lost when active error correction has to be performed on each gate. The main motivation for the quantum bucket brigade approach over a straightforward binary-tree approach is that the equivalent of the active gates are the only gates prone to error, and thus an inverse polynomial in n error rate suffices in order to achieve an overall constant error per qRAM look-up.

The remainder of this chapter is organized as follows. In Sec. 5.1.2 we describe the bucket brigade qRAM architecture and prove that for the Regev and Schiff model [122] the error rate per gate must scale as inverse polynomial in the size of the database. In Sec. 5.1.3 we develop and analyze a simple error model that provides intuition for the overall behaviour of the memory with realistic noisy environments. In Sec. 5.1.4, in order to discuss approaches for introducing quantum error correction inside the qRAM architecture, we introduce a circuit model for the bucket brigade architecture. We then argue in Sec. 5.1.5 that a fault-tolerant bucket brigade qRAM loses the advantage of small number of active components. Finally, in Sec. 5.1.6 we conclude and present some open problems and directions for future research.

5.1.2 Quantum RAM architectures

In [57, 56], Giovanetti *et al.* proposed a quantum bucket brigade addressing scheme requiring only $\mathcal{O}(n)$ activations per memory call. The nodes of the routing binary tree are three level quantum systems (qutrits), with an energy spectrum schematically depicted in Fig. 5.1.

The 2^n qutrits at the nodes of the binary tree are initially prepared in the ground state $|\bullet\rangle$, named the "wait" state, and the memory address is specified by the n-qubit state $|a_0 a_1 \dots a_{n-1}\rangle$. At time t_0, the address qubit $|a_0\rangle$ is input at the root of the tree and it interacts with the qutrit at node 0 changing its state from $|\bullet\rangle$ to $|a_0\rangle$. The states $\{|0\rangle, |1\rangle\}$ of the node qutrit are coupled to two spatial directions (paths), right and left respectively. The role of the coupling is to route the following incoming address photon along the correct path of the binary routing tree. At time t_1, the subsequent address qubit $|a_1\rangle$ is input at the root of the tree. The address qubit $|a_1\rangle$ interacts with the qutrit at node 0 and is physically

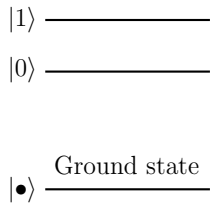


Figure 5.1: Representation of the energy levels of a qutrit used at the nodes of the routing binary tree. The states $|0\rangle$ and $|1\rangle$ form a metastable subspace since the energy difference between the states is required to be much smaller than the difference between the ground state $|\bullet\rangle$ and $|0\rangle$.

routed down the left or right path of the tree depending upon the state $|a_0\rangle$ of node 0. Consequently it changes the state of the corresponding node at level 1 to $|a_1\rangle$. The process continues until all the remaining address qubits are sent through the tree, with the k-th address qubit changing the state of the node at the k-th level from $|\bullet\rangle$ to $|a_k\rangle$. After $\mathcal{O}(n^2)$ time steps[2], a routing path is assigned from the root of the tree to the desired memory location, with only n nodes in the path (one node per level) having a state different from $|\bullet\rangle$. A bucket brigade routing scheme for an 2^3-address qRAM is schematically depicted in Fig. 5.2. The proposed physical implementation of bucket brigade in [56] uses atoms in a cavity as routing nodes and polarization photon states as addressing qubits.

In [56], the authors claim that the bucket brigade scheme is coherent as long as the error per gate, ε scales as $\mathcal{O}(1/n^2)$. For this error scaling, as n increases, the *overall* error rate of the qRAM oracle asymptotically approaches a constant. Although constant or polynomial error rates suffice for some quantum algorithms [101, 39], such error rates are not favourable for some other important quantum algorithms. For example, Regev and Schiff [122] showed that the quadratic speed-up in Grover's searching algorithm vanishes when using oracles with a constant error rate. Namely, in order to regain the quadratic speed-up, the error rate per oracle call should scale no worse than $\mathcal{O}(2^{-n/2})$ (therefore the error rate not only needs to be non-constant but it must vanish at a fast enough rate with increasing n).

In the next few subsections, we construct a simple model of bucket brigade qRAM with errors and show in Appendix C that Regev and Schiff error model [122] resembles the

[2]The k-th address qubit interacts with the first $k-1$ routing nodes, followed by a single interaction with the corresponding node at the k-th level. Considering each interaction takes a single time step, the k-th address qubit changes the state of the corresponding node at the k-th level after k time steps. Considering there are a total of n address qubits, the overall time required is $\mathcal{O}(n^2)$.

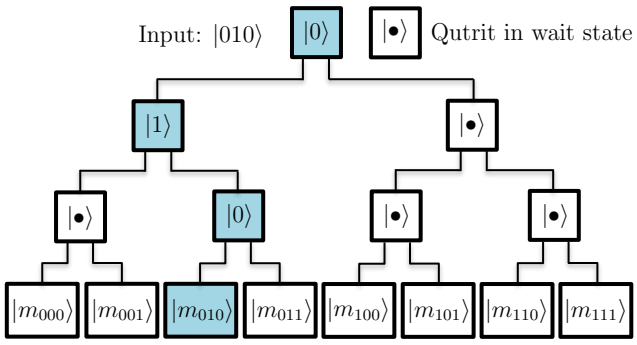


Figure 5.2: Bucket brigade scheme for a qRAM with 8 memory locations. The address register is $|010\rangle$, corresponding to the memory location m_{010}. The path $0 \to 1 \to 0$ is established by sequentially introducing the address qubits $|010\rangle$ into the root of the binary tree.

model we construct. Based on this resemblance and assuming $\mathcal{O}(n^2)$ faulty operations per memory call, we conjecture that in order to implement the qRAM for quantum searching, the overall error rate per memory call has to be in $\mathcal{O}(2^{-n/2})$. In fact, for this to hold, the error rate per gate ε should decrease faster than $1/f(n)$, where $f(n) \in \omega(2^{n/2})$. Thus ε has to be in $o(2^{-n/2})$ and hence much smaller than $\mathcal{O}(1/n^2)$, since the overall error rate per memory call must scale as

$$1 - \left(1 - \frac{1}{f(n)}\right)^{\mathcal{O}(n^2)} \in \Omega\left(\frac{n^2}{f(n)}\right), \tag{5.2}$$

and in order to satisfy

$$1 - \left(1 - \frac{1}{f(n)}\right)^{\mathcal{O}(n^2)} \in \mathcal{O}(2^{-n/2}), \tag{5.3}$$

it is required that

$$\begin{aligned} f(n) \in \Omega\left(n^2 2^{n/2}\right) &\implies \frac{1}{f(n)} \in \mathcal{O}\left(\frac{1}{n^2 2^{n/2}}\right) \\ &\implies \varepsilon \in o(2^{-n/2}). \end{aligned} \tag{5.4}$$

Recently Hong et al. [71] proposed a bucket brigade qRAM scheme in which the number of time steps required per memory call is reduced from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$. While this reduction decreases the overall error rate, the error rate per gate ε must still be in $o(2^{-n/2})$.

The need for super-polynomially small (in n) error rate per gate for real world applications motivates a more thorough analysis of the bucket brigade qRAM scheme and the need for quantum error correction, these topics being the subject of the following subsections.

5.1.3 Error analysis

In this subsection we introduce a simple toy error model for the physical implementation proposed in [56], in which the qutrits are implemented by trapped atoms in cavities. The address qubits are implemented by photons that propagate along the network of cavities, and excite the corresponding qutrit to either of the states $|0\rangle$ or $|1\rangle$, depending on their polarization. In this way, the incoming address photons create a "path" through the binary tree of cavities, leading to the desired memory location. The readout is performed by injecting a "bus" qubit (photon) at the root of the tree that interacts with the desired memory location, copies its value (the states stored by the memory are $|0\rangle$ or $|1\rangle$, and not any superposition), and finally is sent back along the routing tree exiting through the root with the corresponding memory location content. For more details about the physical model an interested reader is referred to [56].

5.1.3.1 Toy error model

In the following we assume that the only source of errors in the above model is due to random flips between the states $|0\rangle$ and $|1\rangle$ of the qutrit. We assume a typical symmetric bit-flip error, in which at each time step the state $|j\rangle$ can either flip to $|j \oplus 1\rangle$ with probability ε or remain unchanged with probability $1 - \varepsilon$. The motivation for considering this error model is that, since the states $\{|0\rangle, |1\rangle\}$ are close together in the energy spectrum, significantly less energy is required to cause a flip between them, hence such flips are more likely to occur. In reality, there may be other sources of errors such as coupling errors, decaying of excited qutrit states to the ground state, loss of photons during the routing process and so on. However, our toy model illustrates the effects of an error that would naturally occur in a realistic physical realization of a qRAM. There is no reason to expect these other sources of errors would help matters (otherwise, one could seek to deliberately introduce or simulate such errors).

It is not hard to observe that any error in the routing process can propagate through the tree resulting in various possibilities. Considering all possible errors in such a model, the possible paths that the bus photon could take in the final step termed as *right path*, *wrong path* and *no-path*, respectively. For convenience, we further assume the operations used to un-compute the path information encoded in the qRAM are error-free.

1) Right path – This scenario occurs when no flips (errors) arise during the routing process. In this ideal scenario, the bus reaches the correct location in the qRAM as specified by the input address. Fig. 5.2 depicts an example of a *right path* given an input address $|010\rangle$.

To compute the probability p_{rp} of such an event, we require that no bit flip occurs at each of the j levels. Taking the intersection of such events for all $n-1$ levels of the binary tree gives the probability of the *right path*

$$\begin{aligned} p_{rp} &= \prod_{j=0}^{n-1}(1-\varepsilon)^{n-j} = (1-\varepsilon)^{\sum_{j=0}^{n-1}(n-j)} \\ &= (1-\varepsilon)^{n(n+1)/2}. \end{aligned} \tag{5.5}$$

2) Wrong path – This error refers to the cases wherein the the bus reaches *any other* location in the qRAM other than the location corresponding to the input address. A *wrong path* error occurs at level i if the state $|j\rangle$ of the active routing qutrit at level i flips to $|j \oplus 1\rangle$ and no other errors occur subsequently (at later time steps). The scenario where another error occurs at a later time step in the levels preceding to the j-th level leads to a *no-path* error which we discuss later. The following two figures illustrate two possible *wrong path*s for the input address $|010\rangle$. In Fig. 5.3, the error is assumed to occur in the third time step, due to which the bus accesses the wrong location corresponding to $|011\rangle$. In Fig. 5.4, the error is assumed to occur in the second time step, with the bus wrongly accessing the location corresponding to $|000\rangle$.

In order to calculate the probability of a *wrong path* occurring, we consider the probability of any path occurring, regardless of whether it is the *right* or *wrong* path, we denote this probability by p_{path}. Suppose the state $|\psi_j\rangle$ is being routed down the qRAM circuit to the j-th level. If any of the $(j-2)$ first routing nodes have flipped then the state will be routed down an unexpected branch and will not excite the j-th level of the tree, resulting in a *no-path*. The probability of success at this given time step is therefore $(1-\varepsilon)^{j-1}$, where ε is the probability of a node flipping, recall we must include the level-0 root node here. This can only for levels 2 and above. The overall probability of success is therefore the product of each of the individual probabilities of success at each time step (including the time step to send the bus qubit down the tree to recover the information stored in the RAM). This probability is given by:

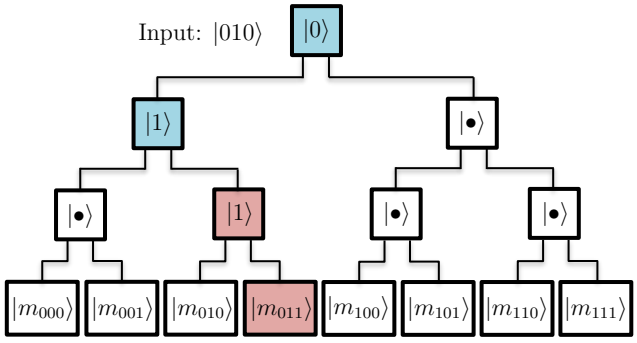


Figure 5.3: Example of a *wrong path* produced by an error at the third time step, given the address $|010\rangle$.

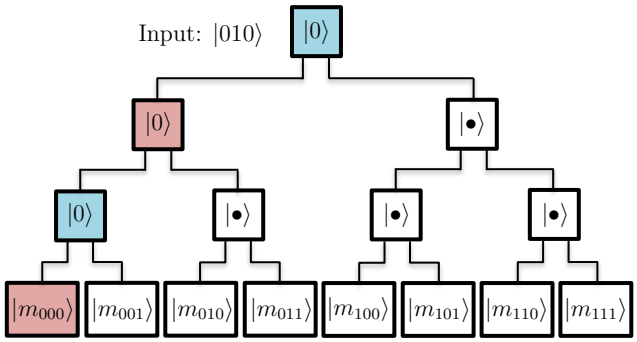


Figure 5.4: Example of a *wrong path* produced by an error at the second time step, given the address $|010\rangle$.

$$p_{path} = p_{wp} + p_{rp} = \prod_{j=2}^{n} (1-\varepsilon)^{j-1}$$
$$= (1-\varepsilon)^{\sum_{j=2}^{n}(j-1)} = (1-\varepsilon)^{n(n-1)/2}. \tag{5.6}$$

As we computed before the probability of a *right path* p_{rp} in Eq. (5.5), the probability

of a *wrong path* is then

$$\begin{aligned} p_{wp} &= p_{path} - p_{rp} \\ &= (1-\varepsilon)^{n(n-1)/2} - (1-\varepsilon)^{n(n+1)/2}. \end{aligned} \tag{5.7}$$

3) No-path – This error refers to the scenario where the bus never reaches *any* location of the qRAM. Such an error arises when a bit flip error occurs in levels 0 to $n-3$. The smallest such tree where this error can occur is therefore a three-level tree (corresponding to a qRAM with 2^3 memory cells), as shown in Fig. 5.5. The difference between a *wrong path* and a *no-path* is that, in the latter, the bus photon does not reach the memory address, hence does not read any information, whereas in the former scenario the bus reaches the wrong address in the qRAM and after the un-computing stage, the bus contains the information of *some* particular address in the qRAM.

We present an example of a *no-path* error in Fig. 5.5, for an input address 010. At the first time instant, the first address photon (i.e. $|0\rangle$) activates the switch (qutrit) in the first layer of the tree. At the second time instant, the address photon $|1\rangle$ interacts with the switch in the first layer, now in state $|0\rangle$, to decide the direction in which it has to be routed. Assuming no error during the second time step, the second address photon is correctly routed to the left path. Assume now that at the third time instant, a flip error occurs on the root qutrit, which flips its state from $|0\rangle$ to $|1\rangle$. The third address photon would then be incorrectly routed to the path on the right. As it can be seen from Fig. 5.5, at the third time instant there are two activated switches in the second level. The readout bus photon can no longer reach any of the memory locations, and will be lost in the second level of routing tree.

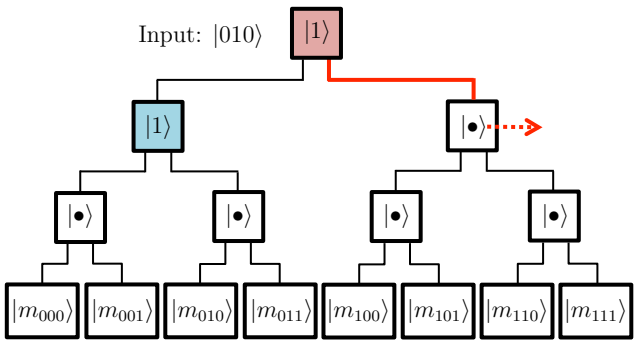


Figure 5.5: Example of a *no-path* given the address $|010\rangle$

The probability of a *no-path* event is simply

$$p_{np} = 1 - p_{wp} - p_{rp} = 1 - (1-\varepsilon)^{n(n-1)/2}. \tag{5.8}$$

If the qRAM is used to implement a quantum oracle O, then O will be faulty, with an error model described by

$$\rho \xrightarrow{O} p_{rp}\hat{O}\rho\hat{O}^\dagger + p_{wp}\mathcal{E}_{wp}(\rho) + p_{np}\mathcal{E}_{np}(\rho), \tag{5.9}$$

with $\hat{O}$ denoting a perfect oracle. Here $\mathcal{E}_{wp}(\cdot)$ and $\mathcal{E}_{np}(\cdot)$ are error channels that corresponds to the *wrong path* and *no-path* errors, respectively.

Our error model Eq. (5.9) is less optimistic than the one of Regev and Schiff [122] of the form $\rho \xrightarrow{O} (1-p)\hat{O}\rho\hat{O}^\dagger + p\rho$. The main difference is that the latter does not mix the amplitudes of the initial starting superposition state in Grover's search algorithm, whereas our model decoheres the system much faster due to the non-trivial errors $\mathcal{E}_{wp}$ and $\mathcal{E}_{np}$. Although we do not have a proof that the quantum query complexity of our model cannot be less than the one considered in [122] (i.e. linear in N), we conjecture (based on a formal proof for a similar decoherence model, see Appendix C) that this is indeed the case.

5.1.3.2 Asymptotic behaviour

In Figs. 5.6, 5.7 and 5.8 we analyze the probabilities of the three types of errors discussed in the previous subsection. The parameters of interest are the error probability per gate, denoted by ε, the overall fidelity of the addressing circuit (i.e. the probability of a *right-path*), denoted by p_{rp}, and the number of levels in the qRAM addressing binary tree denoted by n (corresponding to 2^n memory locations).

For a fixed ε, we see that the *no-path* factor dominates in the error model, asymptotically with n, as depicted in Fig. 5.6.

For a fixed n, again the *no-path* error dominates when the error per gate ε becomes large, see Fig. 5.7.

Finally, for a fixed desired overall fidelity p_{rp}, the maximum allowed error probability per gate ε to achieve the overall fidelity p_{rp} decays exponentially as a function of n, as plotted in Fig. 5.8. From Fig. 5.8 it can be seen that, the error rate per gate of $\mathcal{O}(1/n^2)$ (blue line in Fig. 5.8) as considered in Giovannetti et al. [56] is more optimistic than our error rate $\varepsilon(n)$ (red line in Fig. 5.8)

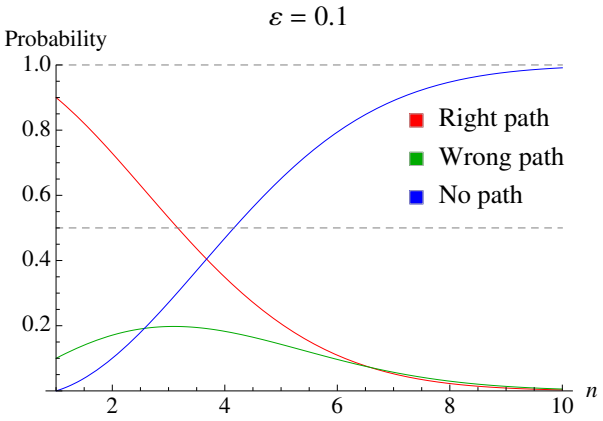


Figure 5.6: Comparison of errors for fixed ε as a function of n.

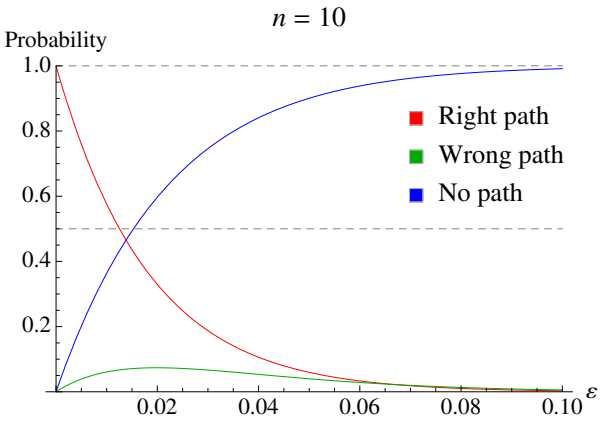


Figure 5.7: Comparison of errors for fixed n as a function of ε.

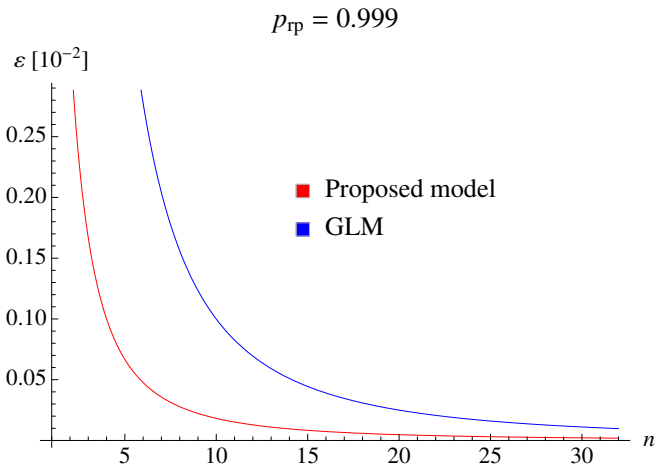


Figure 5.8: Required ε (in dimensionless units of 10^{-2}) as a function of n, for a fixed circuit fidelity. GLM denotes the model proposed in [56].

For larger output fidelity p_{rp}, $\varepsilon(n)$ will always be bounded above by $1/n^2$, with the gap between the two increasing as p_{rp} approaches towards 1. Asymptotically in n, the two graphs converge towards zero.

Simply, the difference between our error $\varepsilon(n)$ and the one in [56] can best be understood by investigating the series expansion

$$p_{rp} = (1-\varepsilon)^{n^2} = 1 + 2\log(1-\varepsilon)\frac{1}{n(n+1)} + \mathcal{O}(\frac{1}{n^4}). \tag{5.10}$$

In [56] the authors considered only the first order $1/n^2$ as a desirable error rate per gate. However, when the output fidelity p_{rp} approaches 1, this approximation is no longer accurate, and higher order terms are important. As mentioned at the end of Sec. 5.1.2, inverse polynomial error rates are not good enough in implementing Grover's search with a qRAM-based oracle. In fact, overall error rates of at most $\mathcal{O}(2^{-n/2})$ are essential.

The dominant *no-path* error term poses a fundamental implementation problem, due to lack of oracle information, similar (see Appendix C) to the noise model investigated by Regev and Schiff [122]. If in the future, qRAM designs could be constructed without the presence of such a *no-path* term (i.e. with *only wrong-path* noise), one can attempt error correction to efficiently reduce the error rate. We demonstrate in Appendix D a possible error correction scheme for a simplified *wrong-path* term governed by bit-flip channels, then show however that the scheme is neither applicable to our error model nor to the Regev and Schiff error model [122].

5.1.4 Circuit model

To facilitate the discussion of error correction, in this section we reformulate the physical model of bucket brigade qRAM in [56] as a quantum circuit. In Fig. 5.9 we present a possible circuit description for an $N = 2^3$ qubit bucket brigade qRAM, in which the memory contains only states in the computational basis $\{|0\rangle, |1\rangle\}$. Our circuit is immediately extendable to $N = 2^n$ and closely simulates the physical model[3] proposed in [56].

The circuit description of the bucket brigade addressing scheme accounts for the temporal aspects of the bucket brigade scheme. Namely, since the address qubits are introduced into the binary tree architecture sequentially, the circuit description should respect this ordering. The input to the circuit are the address qubits $|a_0 a_1 \dots a_n\rangle$. The circuit resembles a binary tree composed of $2^n - 1$ routing nodes, 2^n memory cells and 2^n readout nodes that perform the inverse operations of the routing circuit, used to decouple the qRAM from the address qubits. Additionally, a bus qubit is introduced that interacts with the memory nodes to extract the information stored in the appropriate memory location. It is worth noting that this bus qubit as described may not be physically realistic since it may interact with all the bits in the qRAM. We leave it as such, for simplicity. In practice, if such a non-local qubit is not feasible, one may either work with a phase oracle (as described in Ch. 8 of [82]), or one may use a binary-tree circuit to bring the result of the qRAM look-up to a specific qubit that will be accessed by the quantum algorithm that performs the look-up.

The address qubit $|a_0\rangle$ is used to activate the appropriate branch at the first level of the routing. The address qubit is coupled via a CNOT to an ancillary state prepared in the state $|0\rangle$. This qubit then serves as one of the input qubits along with an additional qubit prepared in the $|1\rangle$ state for the routing node (a CNOT gate with the first qubit as control). Depending on the state of the address qubit, the resulting two-qubit output of the routing node will have a single excited qubit in the $|1\rangle$ state, which we shall call the activated qubit. The activated branch of the tree governs the routing of the subsequent interactions with the address qubits, playing the role of the routing atom in the case of the bucket brigade outlined in [56].

The two qubits at the exit of the level-0 routing node serve as inputs to the second register of the two level-1 routing nodes. These qubits control which of the routing nodes are activated at the next level of the qRAM binary tree architecture. Namely, the qubit that is excited in the $|1\rangle$ state allows for the coupling between the address qubit and an

[3] Further modifications could be made to more closely mimic this model, and are discussed in the caption of Figure 5.9.

introduced $|0\rangle$ state ancilla via a Toffoli gate. Therefore the input to the active routing node is either $|01\rangle$ or $|11\rangle$ depending on the state of the address qubit $|a_1\rangle$. Effectively, the routing operation given by a CNOT gate activates a branch of the tree. For the node that is non-active, the state at the output of the previous level is $|0\rangle$, meaning the Toffoli is not activated and the resulting input and output state to the routing node remains $|00\rangle$. Therefore, after two routing node levels, the output of the routing qubits is composed of 2^2 qubits, with only a single branch being excited depending on the state of the first two address qubits $|a_0 a_1\rangle$. Therefore, this corresponds to an isometry:

$$\begin{aligned} |00\rangle_{a_0 a_1} &\rightarrow |0001\rangle \\ |01\rangle_{a_0 a_1} &\rightarrow |0010\rangle \\ |10\rangle_{a_0 a_1} &\rightarrow |0100\rangle \\ |11\rangle_{a_0 a_1} &\rightarrow |1000\rangle, \end{aligned} \tag{5.11}$$

where the excited output qubits in the $|1\rangle$ state represent an active physical path for the subsequent qRAM operations. This procedure is repeated for n levels, where at the k-th level there are 2^k Toffoli gates and routing nodes. The 2^k Toffoli gates are required to route of the address qubit $|a_k\rangle$ through the previous k levels and the routing node establish the output states in order to route the subsequent address qubits. Since such a circuit performs the appropriate unitary mapping of the address qubits for all computational basis state inputs, by linearity it will extend to all superpositions of input address qubits. An example of the routing procedure for a three-qubit input address state $|010\rangle$ is presented in Fig. 5.9, where the blue highlighted nodes correspond to the activated nodes.

After the completion of the n routing node levels, memory readout is performed. The reading is performed by introducing a bus qubit prepared in the $|0\rangle$ state which is the target of 2^n Toffoli gates. Each of the 2^n qubits at the output of the qRAM routing nodes pair with one of the memory cells to serve as control qubits for the Toffoli gates. Since only a single output qubit from the routing scheme is activated, only a single Toffoli gate couples with a memory location to the bus qubit. The bus qubit is represented by the bottom qubit in Fig. 5.9 while the 2^3 memory qubits are represented between the qRAM routing architecture and the bus qubit.

Having completed the coupling of the address qubit, the state of the qRAM routing qubits must be decoupled from the address and bus qubits. Each of the gates from the routing circuit are performed in reverse order, which corresponds to performing the inverse unitary coupling transformation between the address qubits and the routing qubits. The resulting state couples the address qubits with the corresponding memory qubit and has decouples the routing qubits to their input ancillary states.

5.1.5 Error correction

The results from Sec. 5.1.3 motivate the need for quantum error correction to be implemented at each node in order to protect against errors that may cause detrimental faults in path information.

5.1.5.1 Imposing a quantum error correcting code

In choosing a quantum error correcting code (QECC) to protect the path information that is stored in each node, it is essential to choose an encoding that can be implemented fault-tolerantly, to allow for the generalization to large computational systems. Moreover, the QECC should be chosen such that it can naturally be incorporated from the quantum computer that is accessing the qRAM. In order to analyze the desired error correction properties of a bucket brigade qRAM architecture we consider the circuit presented in Fig. 5.9. The key gate components at each site are the CNOT and Toffoli gates.

The most natural construction of a QECC that can implement such operations with minimal overhead would be the 15-qubit Reed-Muller code. The reason for choosing such a code would be that decomposing the gate operations in the routing circuit as a sequence of CNOT and Toffoli gates has the advantage that each of these gates can be implemented in a transversal manner. Transversality is defined as the ability to implement a logical gate by applying physical gates that have support at at most a single location per encoded codeblock: it is the most natural way to guarantee fault-tolerance. However, if the quantum computing device leads more naturally to another form of quantum error correction encoding, methods such as state distillation or other schemes for universal fault-tolerance can be used [32, 114, 23, 78, 6].

The focus of many fault-tolerant implementations are through the CSS code construction [127, 36, 128]. A CSS quantum code is constructed using two classical error correcting codes, each individually used to address X and Z type errors. Given that any quantum error can be decomposed in terms of a linear combination of Pauli operators, developing an error correcting code that can address both types of errors will be sufficient for the construction of a QECC.

Let $\mathcal{C}_X$ be a classical error correcting code of length n that has the associated parity check matrix H_X, where each 0 in the parity check matrix of the classical code is replaced by the two-dimensional identity matrix I and each 1 in the classical parity check matrix is replaced by the Pauli X operator. Similarly, let $\mathcal{C}_Z$ be a second classical error correcting code of length n with an associated parity check matrix H_Z, where each 1 in the classical

parity check matrix has been now replaced by the Pauli Z operator. If $\mathcal{C}_X^{\perp} \subseteq \mathcal{C}_Z$ then by combining the stabilizers generated from the parity check matrices of both codes, H_X and H_Z, the resulting stabilizer code forms a QECC. The number of physical qubits in the code is n, the number of logical qubits is given by $k_X + k_Z - n$, where k_i is the number of logical states in the given classical code i and the distance of the code is at least the minimum of the distance of the two classical codes. One of the many appealing features of the CSS code construction is the transversality of the CNOT gate, a feature of the X and Z stabilizers being independent. A particular example of a CSS code is the 15-qubit Reed-Muller code mentioned above.

5.1.5.2 Number of activations in a CSS code

In the implementation of Giovanetti *et al.* [56], one of the primary advantages is the number of gate activations that are needed per level of the bucket brigade scheme. More simply, a CNOT (Toffoli) gate in their scheme is "activated" only when the control qubit(s) is (are) in the state $|1\rangle$. Since only one register is in such a state at a given level, the total number of activations can thus be kept low. In a physical implementation, this is relevant as an activated path may represent the presence of a physical excitation without which no physical process occurs, therefore one can think of these non-activated gates as in fact being the identity operation. However, such an advantage vanishes when imposing a CSS code in order to protect from errors due to the symmetry in the number of $|0\rangle$ and $|1\rangle$ states the logical states encoding the path information.

In the CSS code construction, two classical codes were taken to form a QECC. Therefore, given some codeword of the classical code $c \in \mathcal{C}_Z$, the equivalent quantum state written out in computation basis $|c\rangle$ must be stabilized by the Z generators of the code, by definition of being a codeword of the classical code $\mathcal{C}_Z$. However, in order to be a logical state of the CSS code, it must also be stabilized by the elements of the group generated by the X stabilizers. Therefore, the codestate will be the superposition of the application of all X stabilizers upon $|c\rangle$,

$$\left|c + \mathcal{C}_Z^{\perp}\right\rangle = \sum_{x \in \mathcal{C}_Z^{\perp}} |c + x\rangle = \frac{1}{2^{n-k_X}} \prod_i (I^{\otimes n} + S_{X,i})|c\rangle, \tag{5.12}$$

where $\{S_{X,i}\}$ are the generators of the X stabilizer group, equivalently given by the rows of the parity check matrix H_X.

Consider the form of Eq. (5.12), given the state $|c\rangle$ written in the computational basis, the action of the operator $(I^{\otimes n} + S_{X,1})$ will be the equally weighted superposition of the

state $|c\rangle$ and $S_{X,1}|c\rangle$, which will differ at the location where $S_{X,1}$ has a Pauli X in its description. Therefore, at these locations half of the states in the superposition will have a physical $|0\rangle$ state and half will have a physical $|1\rangle$ state. Then acting upon the state with the operator $(I^{\otimes n} + S_{X,2})$ will have the same effect on all the states in the superposition, with now an even number of physical $|0\rangle$ and $|1\rangle$ states occurring at location with Pauli X in $S_{X,2}$. Repeating this for all X generators, any location with a X operator in one of the stabilizers will necessarily have half of the states in the superposition in each of the physical basis states. In order for the code to protect against any arbitrary single qubit error, each physical qubit must be protected by at least one X stabilizer operator with support the given location, otherwise it would be vulnerable to a single Z error at this location. As such, all relevant CSS codestates will have an equal number of each of the physical basis states when writing out the expansion of the state in the computational basis.

In a physical implementation, such as that of Giovanetti *et al.* [56], a qubit in the state $|1\rangle$ represents an activated physical process, and as such the advantage of the bucket brigade scheme is that the number of such processes are kept low. However, due to the symmetry in the number of activations that must exist in both the logical ground and excited states, this advantage no longer exists when considering CSS codes. More generally, non-symmetric codes, that is codes where the logical $|0\rangle$ state and logical $|1\rangle$ have a differing number of physical states in the excited state $|1\rangle$, are not desirable for the purposes of error correction as they will be more susceptible to Z errors. The three-qubit repetition code is an extreme example of such a property.

In principle, for the physical error model discussed in Subsection 5.1.3, one can envision using the detection of a photon lost in the routing structure as a means to correct for *no-path* errors (see Figure 5.5). However, detecting the exact node at which a photon was lost reveals path information about the state being read by the qRAM (since the previous node in the routing structure would have necessarily been activated by the address qubits) which leads to a loss of coherence in the system. Therefore, any photon detection has to identify the level at which the photon was lost, while not revealing exactly where. It is hard to envisage a practical means for experimentally realizing a photon detection with this property (for example, by somehow symmetrizing the loss of the photon across the exponentially many nodes at a given level). Furthermore, even if this is achieved, one still faces the problem that the lost photon contained path information. Thus, destroying the photon with this path information is equivalent to a dephasing error leading to a further loss of coherence.

In conclusion, if one encodes each node of the bucket brigade qRAM in an error correcting code, then all nodes of the circuit are activated at a physical level, and essentially the qRAM architecture becomes equivalent to a fanout architecture. Even it the latter case,

designing a good quantum error correcting code is highly non-trivial. An important issue is that the syndrome measurement should not reveal any information whatsoever about the physical location of the nodes affected by errors. Otherwise, path information is being revealed, which decoheres the system.

5.1.6 Summary & outlook

We analyzed the robustness of the bucket brigade qRAM scheme introduced in [57, 56] under an optimistic error model.

The primary advantage of the bucket brigade scheme is the need for a polynomial in n (rather than exponential) number of gate activations per memory reading. When used for quantum algorithms [68, 138, 101, 121, 102, 100] that require only a polynomial number of queries, the error rate of the qRAM can scale polynomially in n, and error correction may not be required. By contrast, we give evidence that under realistic error models, whenever the qRAM is used as an oracle for quantum searching, its per gate error rate has to scale as $o(2^{-n/2})$ [56], motivating the need for quantum error correction.

We argued that using traditional error correcting techniques offsets the main advantage of the bucket brigade scheme when used with algorithms that make super-polynomially many oracle queries. Since each component of the routing architecture has to be actively error corrected in order to protect against detrimental faults, the overall scheme requires an exponential number of physical gate activations, even if the number of logical gate activations remains polynomial.

An interesting question that remains unanswered is whether there exists a realistic architecture-specific error correction technique that could recover the polynomial number of physical gate activations of the routing scheme while still guaranteeing fault-tolerance. For example, if one tries to use an error correction mechanism whereby one only uses multi-qubit code states along the active path, then one has the problem of extracting syndromes and applying corrections in a way that does not identify which path has the non-trivial syndromes (since such information would lead to decoherence). If in this case one attempts to extract the syndrome without leaving a trace of which node in a given level it came from, then the problem seems at least as challenging as implementing a reliable qRAM.

Moreover, it would be interesting to investigate whether the requirement for super-polynomial suppression of the error rate is a characteristic of quantum searching algorithms or a more general feature of query complexity with faulty oracles.

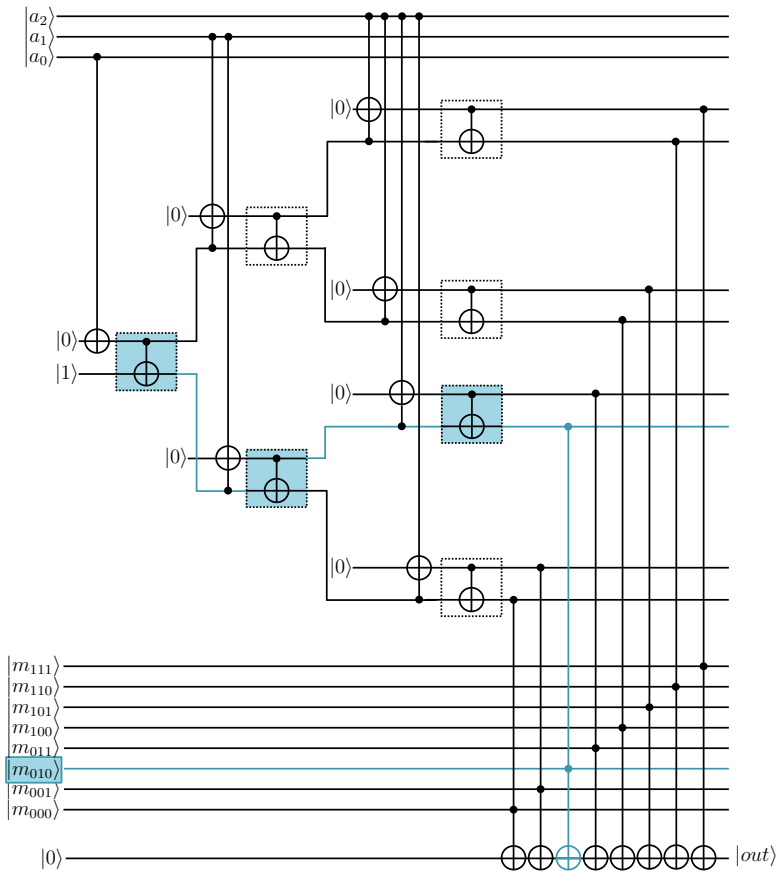


Figure 5.9: Circuit for bucket brigade qRAM. Nodes to the left of the memory cell are *routing nodes*. The dashed squares represents the memory locations. The first layer of nodes immediately to the right of the memory are the *coupling nodes*. Finally, the nodes on the right are the *read out nodes*. A possible input is e.g. $|a_0a_1a_2\rangle = |010\rangle$, for which the circuit reads the memory location m_{010}. The path leading to the location m_{010} is represented in blue colour, and the active routing and readout nodes are highlighted. One could more closely mimic the physical flow of information in the bucket brigade qRAM by adding an additional qubit at each node in the binary tree we see in the diagram. Then, for $k \in \{0, \ldots, n-1\}$, we add an initial controlled-NOT gate to copy a_k to the root node, followed by a series of $\mathcal{O}(2^k)$ controlled-SWAPs that will bring the value of a_k to the unique node in level k defined by the bits $a_0, a_1, \ldots, a_{k-1}$. While this adds exponentially many gates, it does not change the overall gate complexity, and these additional gates only add $\mathcal{O}(k)$ to the depth of the circuit. This also illustrates that the exponential depth implicit in the circuit we describe in the diagram can easily be reduced to polynomial depth by further mimicking the ideas presented in the qRAM proposal. We leave the circuit diagram in this simpler form, since it does not affect our arguments in Subsections 5.1.3 and 5.1.5.

Chapter 6

Conclusion

Over the past decades, quantum computing and quantum information have seen rapid growth in both experimental and theoretical domains. Having demonstrated levels of control to implement small information theoretic tasks, the presence of entanglement and superposition, and even small instances of error correcting codes in a variety of different physical architectures, experiments have began to push to levels of precision that would be required for fault-tolerant computation. As these experimental efforts continue to improve in accuracy and increase the number of qubits which can be coherently controlled, the questions that are asked in this thesis will start to come to the forefront of implementation design.

Chapter 2 provided a solution to the overhead problem that is presented by magic state distillation. Distillation is a beautiful idea in which logical computational requirements are reduced at the expense of more complicated state preparation, and while improvements have been made in its theoretical performance, it will likely dominate the resource overhead of leading architectural proposals [55]. The universal concatenation scheme provides a potential alternative, where distance and protection of the code are traded for the ability to perform universal fault-tolerant gates. Moreover, the theoretical threshold of the code against circuit level depolarizing noise is quite promising being similar to other concatenated proposals. However, the threshold is still roughly an order of magnitude lower than the leading architectural proposal, the surface code, under similar noise models. Therefore, the need to concatenate to further levels will likely outweigh the resource advantage of not requiring magic state distillation. That being said, it would still be an interesting theoretical challenge to implement the presented scheme using other error correcting codes, and perhaps with such an implementation the threshold could be improved. However, it is difficult to imagine a lower threshold without reducing the weight of stabilizer generators of

the concatenated scheme. Moreover, the result of Section 2.4 points to a rather restricted class of transversal operators, many of which already have existing codes. Alternatively, the option of hybrid approaches could be explored where only a subset of the qubits are re-encoded, as was eluded to with the 49-qubit proposal. Such an idea could be generalized to the full class of 2D and 3D color codes and this could potentially provide an avenue to reduce overhead.

Chapter 3 addressed the compatibility of 2D topological error correcting codes with universal fault-tolerant computation. Planar topological architectures are very interesting experimentally as their requirements are much less strict than other error correction schemes. However, the implementation of a universal fault-tolerant gate set had also been limited to state distillation. The idea behind the stacked code design is to provide certain non-local long range interactions between qubits in order to simulate the code as if it was in a 3D architecture. Moreover, Bravyi and Cross showed that one could forgo the need for arbitrary long range operations and could cut such operations into local terms. However, the main drawback of the proposed scheme is that no fault-tolerant threshold exists. This poses an interesting set of open questions: Given the restriction of 2D architectures to gates outside the Clifford hierarchy, what non-local operations are required in order to implement a gate other than a Clifford gate? How can one measure the degree of non-locality that a set of non-local operations are introducing? What is the connection between the dimension in which a local circuit can be embedded and fault-tolerance and can these connections be generalized to topological memories? The upshot is that the structure of quantum information can be quite complex, its manipulation can take many forms, and the theoretical ideas of such operations seem plentiful.

Operationally, quantum error correctable channels admit an interesting duality with private quantum channels. Chapter 4 uncovers the structure of private quantum channels, provides a new form of privacy, and expands the set of private quantum channels. Namely, just as the theory of quantum error correctable subsystems changed which encodings and operations can be realized in an error correction setting, the realm of private channels is enhanced by the structure of private quantum channels. Due to the aforementioned duality between error correction and privacy, we explored whether this expanded notion of privacy would expand the set of error correctable systems. While the answer is negative, it does emphasize an important principle for quantum information: quantum operators may be very well understood in a given setting, however it is always important to clarify the underlying assumptions, and asses whether these assumptions could be challenged. If so, how does the structure of quantum operations change? What is the underlying operational significance and does it affect other fields?

Chapter 5 assessed the viability of a proposed scheme for quantum RAM. It was shown

that mapping computational information to path information, while keeping error rates low, is a complicated task and would likely require a full fault-tolerant architecture. As fault-tolerant architectures improve and increasingly complex algorithms are implemented, it will be imperative that all forms of errors can be addressed within information accessing designs. Developing essential primitives for computing devices will continue to gain importance as the experimental research community scales up fault-tolerant computing architectures.

We are at a very interesting crossroad in quantum information and computation, as technology advances from prototypes to full computational machines. This thesis explores theoretical questions that are fundamental to the development of successful quantum computational machines by correcting for the unavoidable errors that will plague such devices. Quantum information is imperative to the future of science and computing and also provides insight into the structure of quantum mechanics in nature.

References

[1] Dorit Aharonov and Michael Ben-Or. Fault-tolerant quantum computation with constant error. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 176–188. ACM, 1997.

[2] Panos Aliferis, Daniel Gottesman, and John Preskill. Quantum accuracy threshold for concatenated distance–3 codes. *Quant. Inf. Comput.*, 6:97–165, 2006.

[3] Panos Aliferis, Daniel Gottesman, and John Preskill. Accuracy threshold for postselected quantum computation. *Quant. Inf. Comput.*, 8:181–244, 2008.

[4] Andris Ambainis. Quantum walk algorithm for element distinctness. *SIAM Journal on Computing*, 37(1):210–239, 2007.

[5] Andris Ambainis, Michele Mosca, Alain Tapp, and Ronald de Wolf. Private quantum channels. *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 547–553, 2000.

[6] Jonas T. Anderson, Guillaume Duclos-Cianci, and David Poulin. Fault-tolerant conversion between the steane and reed-muller quantum codes. *Phys. Rev. Lett.*, 113:080501, 2014.

[7] Jonas T. Anderson and Tomas Jochym-O'Connor. Classification of transversal gates in qubit stabilizer codes. *Quant. Inf. Comput.*, 16:0771–0802, 2016.

[8] Srinivasan Arunachalam. *Quantum speed-ups for boolean satisfiability and derivative-free optimization.* PhD thesis, University of Waterloo, 2014.

[9] Srinivasan Arunachalam, Vlad Gheorghiu, Tomas Jochym-OConnor, Michele Mosca, and Priyaa Varshinee Srinivasan. On the robustness of bucket brigade quantum ram. *New J. Phys.*, 17(12):123010, 2015.

[10] Dave Bacon. Operator quantum error-correcting subsystems for self-correcting quantum memories. *Phys. Rev. A*, 73(1):012340, 2006.

[11] Somshubhro Bandyopadhyay, P. Oscar Boykin, Vwani Roychowdhury, and Farrokh Vatan. A new proof for the existence of mutually unbiased bases. *Algorithmica*, 34:512–528, 2002.

[12] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White, J. Mutus, A. G. Fowler, B. Campbell, et al. Superconducting quantum circuits at the surface code threshold for fault tolerance. *Nature*, 508(7497):500–503, 2014.

[13] Stephen D. Bartlett, Patrick Hayden, and Robert W. Spekkens. Random subspaces for encryption based on a private shared cartesian frame. *Phys. Rev. A*, 72(5):052329, 2005.

[14] Stephen D. Bartlett, Terry Rudolph, and Robert W. Spekkens. Decoherence-full subsystems and the cryptographic power of a private shared reference frame. *Phys. Rev. A*, 70(3):032307, 2004.

[15] Paul Benioff. The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines. *Journal of Statistical Physics*, 22(5):563–591, 1980.

[16] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM journal on Computing*, 26(5):1510–1523, 1997.

[17] Charles H. Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. In *International Conference on Computer System and Signal Processing, IEEE, 1984*, pages 175–179, 1984.

[18] Charles H. Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres, and William K. Wootters. Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels. *Phys. Rev. Lett.*, 70(13):1895, 1993.

[19] Charles H Bennett and Stephen J. Wiesner. Communication via one-and two-particle operators on einstein-podolsky-rosen states. *Phys. Rev. Lett.*, 69(20):2881, 1992.

[20] Michael E. Beverland and John Preskill. Private communication. 2014.

[21] Alex Bocharov, Yuri Gurevich, and Krysta M. Svore. Efficient decomposition of single-qubit gates into v basis circuits. *Phys. Rev. A*, 88:012313, 2013.

[22] Héctor Bombín. Dimensional jump in quantum error correction. *arXiv:1412.5079*, 2014.

[23] Héctor Bombín. Gauge color codes: optimal transversal gates and gauge fixing in topological stabilizer codes. *New J. Phys.*, 17(8):083002, 2015.

[24] Héctor Bombín. Single-shot fault-tolerant quantum error correction. *Phys. Rev. X*, 5:031043, 2015.

[25] Héctor Bombín and Miguel A. Martin-Delgado. Topological quantum distillation. *Phys. Rev. Lett.*, 97:180501, 2006.

[26] Héctor Bombín and Miguel A. Martin-Delgado. Optimal resources for topological two-dimensional stabilizer codes: Comparative study. *Phys. Rev. A*, 76(1):012305, 2007.

[27] P. Oscar Boykin and Vwani Roychowdhury. Optimal encryption of quantum bits. *Phys. Rev. A*, 67(4):042317, 2003.

[28] P. Oscar Boykin, Meera Sitharam, Pham Huu Tiep, and Pawel Wocjan. Mutually unbiased bases and orthogonal decompositions of Lie algebras. *Quant. Inf. Comput.*, 7:371–382, 2007.

[29] Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum cryptanalysis of hash and claw-free functions. *ACM Sigact News*, 28(2):14–19, 1997.

[30] Sergey Bravyi and Andrew Cross. Doubled color codes. *arXiv:1509.03239*, 2015.

[31] Sergey Bravyi and Jeongwan Haah. Magic-state distillation with low overhead. *Phys. Rev. A*, 86(5):052329, 2012.

[32] Sergey Bravyi and Alexei Kitaev. Universal quantum computation with ideal Clifford gates and noisy ancillas. *Phys. Rev. A*, 71:022316, 2005.

[33] Sergey Bravyi and Robert König. Classification of topologically protected gates for local stabilizer codes. *Phys. Rev. Lett.*, 110(17):170503, 2013.

[34] Benjamin J. Brown, Naomi H. Nickerson, and Dan E. Browne. Fault tolerance with the gauge color code. *arXiv:1503.08217*, 2015.

[35] Harry Buhrman, Ilan Newman, Hein Röhrig, and Ronald De Wolf. Robust polynomials and quantum algorithms. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 593–604. Springer, 2005.

[36] A. Robert Calderbank and Peter W. Shor. Good quantum error-correcting codes exist. *Phys. Rev. A*, 54:1098–1105, 1996.

[37] Christopher Chamberland, Tomas Jochym-O'Connor, and Raymond Laflamme. *In Preparation.*

[38] Christopher Chamberland, Tomas Jochym-O'Connor, and Raymond Laflamme. Thresholds for universal concatenated quantum codes. *Phys. Rev. Lett.*, 117:010501, 2016.

[39] Andrew M. Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A. Spielman. Exponential algorithmic speedup by a quantum walk. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 59–68. ACM, 2003.

[40] Jerry M. Chow, Jay M. Gambetta, Easwar Magesan, David W. Abraham, Andrew W. Cross, B. R. Johnson, Nicholas A. Masluk, Colm A. Ryan, John A. Smolin, Srikanth J. Srinivasan, et al. Implementing a strand of a scalable fault-tolerant quantum computing fabric. *Nature communications*, 5, 2014.

[41] Amber Church, David W Kribs, Rajesh Pereira, and Sarah Plosker. Private quantum channels, conditional expectations, and trace vectors. *Quantum Information & Computation*, 11(9-10):774–783, 2011.

[42] Juan I Cirac and Peter Zoller. Quantum computations with cold trapped ions. *Physical review letters*, 74(20):4091, 1995.

[43] Richard Cleve, Daniel Gottesman, and Hoi-Kwong Lo. How to share a quantum secret. *Phys. Rev. Lett.*, 83(3):648, 1999.

[44] David G. Cory, Amr F. Fahmy, and Timothy F. Havel. Ensemble quantum computing by nmr spectroscopy. *Proceedings of the National Academy of Sciences*, 94(5):1634–1639, 1997.

[45] Claude Crépeau, Daniel Gottesman, and Adam Smith. Secure multi-party quantum computation. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 643–652. ACM, 2002.

[46] Ben Criger. *Practical Advances in Quantum Error Correction & Communication.* PhD thesis, University of Waterloo, 2013.

[47] Andrew Cross, David P. DiVincenzo, and Barbara M. Terhal. A comparative code study for quantum fault tolerance. *Quant. Inf. Comput.*, 9:541–572, 2009.

[48] Kenneth R. Davidson. *C*-algebras by example*. Fields Institute Monograph Series, American Mathematical Society, 1996.

[49] David Deutsch. Quantum theory, the church-turing principle and the universal quantum computer. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 400, pages 97–117. The Royal Society, 1985.

[50] Michel H. Devoret and Robert J. Schoelkopf. Superconducting circuits for quantum information: an outlook. *Science*, 339(6124):1169–1174, 2013.

[51] David P. DiVincenzo and Peter W. Shor. Fault-tolerant error correction with efficient quantum codes. *Phys. Rev. Lett.*, 77(15):3260, 1996.

[52] Bryan Eastin and Emanuel Knill. Restrictions on transversal encoded quantum gate sets. *Phys. Rev. Lett.*, 102:110502, 2009.

[53] Joseph Emerson. Course notes, Open Quantum Systems. 2011.

[54] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21:467, 1982.

[55] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. Surface codes: Towards practical large-scale quantum computation. *Phys. Rev. A*, 86:032324, 2012.

[56] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Architectures for a quantum random access memory. *Phys. Rev. A*, 78(5):052310, 2008.

[57] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum random access memory. *Phys. Rev. Lett.*, 100(16):160501, 2008.

[58] Daniel Gottesman. *Stabilizer Codes and Quantum Error Correction*. PhD thesis, California Institute of Technology, 1997.

[59] Daniel Gottesman. The heisenberg representation of quantum computers, talk at. In *International Conference on Group Theoretic Methods in Physics*. Citeseer, 1998.

[60] Daniel Gottesman. Theory of fault-tolerant quantum computation. *Phys. Rev. A*, 57(1):127, 1998.

[61] Daniel Gottesman and Isaac Chuang. Quantum teleportation is a universal computational primitive. *Nature*, 42:390–393, 1999.

[62] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219. ACM, 1996.

[63] Lov K. Grover and Terry Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. *arXiv: quant-ph/0208112*, 2002.

[64] Mauricio Gutiérrez and Kenneth R. Brown. Comparison of a quantum error-correction threshold for exact and approximate errors. *Phys. Rev. A*, 91(2):022335, 2015.

[65] Mauricio Gutiérrez, Conor Smith, Livia Lulushi, Smitha Janardan, and Kenneth R. Brown. Errors and pseudo-thresholds for incoherent and coherent noise. *arXiv:1605.03604*, 2016.

[66] Mauricio Gutiérrez, Lukas Svec, Alexander Vargo, and Kenneth R. Brown. Approximation of realistic errors by clifford channels and pauli measurements. *Phys. Rev. A*, 87(3):030302, 2013.

[67] Serge Haroche, M. Brune, and Jean-Michel Raimond. Experiments with single atoms in a cabity: entanglement, schrödinger's cats and decoherence. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 355(1733):2367–2380, 1997.

[68] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103(15):150502, 2009.

[69] Charles D. Hill, Austin G. Fowler, David S. Wang, and Lloyd C.L. Hollenberg. Fault-tolerant quantum error correction code conversion. *Quant. Inf. Comput.*, 13(3 & 4):0439–0451), 2013.

[70] Alexander S. Holevo. Complementary channels and the additivity problem. *Theory of Probability & Its Applications*, 51(1):92–100, 2007.

[71] Fang-Yu Hong, Yang Xiang, Zhi-Yan Zhu, Li-zhen Jiang, and Liang-neng Wu. Robust quantum random access memory. *Phys. Rev. A*, 86(1):010306, 2012.

[72] Clare Horsman, Austin G Fowler, Simon Devitt, and Rodney Van Meter. Surface code quantum computing by lattice surgery. *New J. Phys.*, 14(12):123011, 2012.

[73] Peter Høyer, Michele Mosca, and Ronald De Wolf. Quantum search on bounded-error inputs. In *International Colloquium on Automata, Languages, and Programming*, pages 291–299. Springer, 2003.

[74] Richard C. Jaeger and Travis N Blalock. *Microelectronic circuit design*. McGraw-Hill New York, 2003.

[75] Tomas Jochym-O'Connor and Stephen D. Bartlett. Stacked codes: Universal fault-tolerant quantum computation in a two-dimensional layout. *Phys. Rev. A*, 93:022323, 2016.

[76] Tomas Jochym-O'Connor, David W. Kribs, Raymond Laflamme, and Sarah Plosker. Private quantum subsystems. *Phys. Rev. Lett.*, 111:030502, 2013.

[77] Tomas Jochym-O'Connor, David W. Kribs, Raymond Laflamme, and Sarah Plosker. Quantum subsystems: exploring the complementarity of quantum privacy and error correction. *Phys. Rev. A*, 90:032305, 2014.

[78] Tomas Jochym-O'Connor and Raymond Laflamme. Using concatenated quantum codes for universal fault-tolerant quantum gates. *Phys. Rev. Lett.*, 112:010505, 2014.

[79] Tomas Jochym-O'Connor, Yafei Yu, Bassam Helou, and Raymond Laflamme. The robustness of magic state distillation against errors in clifford gates. *Quant. Inf. Comput.*, 13:361–378, 2013.

[80] Cody Jones. Multilevel distillation of magic states for quantum computing. *Phys. Rev. A*, 87(4):042305, 2013.

[81] Cody Jones, Peter Brooks, and Jim Harrington. Gauge color codes in two dimensions. *Phys. Rev. A*, 93(5):052332, 2016.

[82] Phillip Kaye, Raymond Laflamme, and Michele Mosca. *An Introduction to Quantum Computing*. Oxford University Press, Oxford, England, 2007.

[83] J. Kelly, R. Barends, A. G. Fowler, A. Megrant, E. Jeffrey, T. C. White, D. Sank, J. Y. Mutus, B. Campbell, Yu Chen, et al. State preservation by repetitive error detection in a superconducting quantum circuit. *Nature*, 519(7541):66–69, 2015.

[84] Christopher King, Keiji Matsumoto, Michael Nathanson, and Mary Beth Ruskai. Properties of conjugate channels with applications to additivity and multiplicativity. *Markov Processes and Related Fields*, 13:391–423, 2007.

[85] Alexei Y. Kitaev. Quantum computations: algorithms and error correction. *Russian Mathematical Surveys*, 52(6):1191–1249, 1997.

[86] Vadym Kliuchnikov, Dmitri Maslov, and Michele Mosca. Asymptotically optimal approximation of single qubit unitaries by clifford and t circuits using a constant number of ancillary qubits. *Phys. Rev. Lett.*, 110(19):190502, 2013.

[87] Emanuel Knill. Fault-tolerant postselected quantum computation: schemes. *arXiv:quant-ph/0402171*, 2004.

[88] Emanuel Knill and Raymond Laflamme. Theory of quantum error-correcting codes. *Phys. Rev. A*, 55:900–911, 1997.

[89] Emanuel Knill, Raymond Laflamme, and Wojciech H. Zurek. Threshold accuracy for quantum computation. *arXiv: quant-ph/9610011*, 1996.

[90] Dennis Kretschmann, David W Kribs, and Robert W. Spekkens. Complementarity of private and correctable subsystems in quantum cryptography and error correction. *Phys. Rev. A*, 78(3):032330, 2008.

[91] David Kribs, Raymond Laflamme, and David Poulin. Unified and generalized approach to quantum error correction. *Phys. Rev. Lett.*, 94(18):180501, 2005.

[92] David W Kribs, Raymond Laflamme, David Poulin, and Maia Lesosky. Operator quantum error correction. *Quantum Inf. Comput.*, 6:382, 2005.

[93] Aleksander Kubica and Michael E. Beverland. Universal transversal gates with color codes: A simplified approach. *Phys. Rev. A*, 91:032330, 2015.

[94] Greg Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM Journal on Computing*, 35(1):170–188, 2005.

[95] Raymond Laflamme, Cesar Miquel, Juan Pablo Paz, and Wojciech H. Zurek. Bound states for magic state distillation in fault-tolerant quantum computation. *Phys. Rev. Lett.*, 77(198), 1996.

[96] Andrew J. Landahl and Chris Cesare. Complex instruction set computing architecture for performing accurate quantum z rotations with less magic. *arXiv:1302.3240*, 2013.

[97] Andrew J. Landahl and Ciaran Ryan-Anderson. Quantum computing by color-code lattice surgery. *arXiv:1407.5103*, 2014.

[98] Jeremy Levick, Tomas Jochym-O'Connor, David W. Kribs, Raymond Laflamme, and Rajesh Pereira. Private quantum subsystems and quasiorthogonal operator algebras. *Journal of Physics A: Mathematical and Theoretical*, 49(12):125302, 2016.

[99] Seth Lloyd. Universal quantum simulators. *Science*, 237:1073–1077, 1996.

[100] Seth Lloyd, Silvano Garnerone, and Paolo Zanardi. Quantum algorithms for topological and geometric analysis of data. *Nature communications*, 7, 2016.

[101] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum algorithms for supervised and unsupervised machine learning. *arXiv:1307.0411*, 2013.

[102] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(9):631–633, 2014.

[103] J. Majer, J. M. Chow, J. M. Gambetta, Jens Koch, B. R. Johnson, J. A. Schreier, L. Frunzio, D. I. Schuster, A. A. Houck, Andreas Wallraff, et al. Coupling superconducting qubits via a cavity bus. *Nature*, 449(7161):443–447, 2007.

[104] Adam M. Meier, Bryan Eastin, and Emanuel Knill. Magic-state distillation with the four-qubit code. *arXiv:1204.4221*, 2012.

[105] Chris Monroe, D. M. Meekhof, B. E. King, W. M. Itano, and D. J. Wineland. Demonstration of a fundamental quantum logic gate. *Phys. Rev. Lett.*, 75(25):4714, 1995.

[106] Michele Mosca and Phillip Kaye. Quantum networks for generating arbitrary quantum states. In *International Conference on Quantum Information*, page PB28. Optical Society of America, 2001.

[107] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, England, 2000.

[108] Michael A. Nielsen, Emanuel Knill, and Raymond Laflamme. Complete quantum teleportation using nuclear magnetic resonance. *Nature*, 396(6706):52–55, 1998.

[109] Michael A. Nielsen and David Poulin. Algebraic and information-theoretic conditions for operator quantum error correction. *Phys. Rev. A*, 75(6):064304, 2007.

[110] Eesa Nikahd, Mehdi Sedighi, and Morteza Saheb Zamani. Non-uniform code concatenation for universal fault-tolerant quantum computing. *arXiv preprint arXiv:1605.07007*, 2016.

[111] Hiromichi Ohno. Quasi-orthogonal subalgebras of matrix algebras. *Linear Algebra and its Applications*, 429:2146–2158, 2008.

[112] Hiromichi Ohno and Denes Petz. Generalizations of Pauli channels. *Acta Mathematica Hungarica*, 124:165–177, 2009.

[113] Adam Paetznick and Ben W. Reichardt. Fault-tolerant ancilla preparation and noise threshold lower bounds for the 23-qubit golay code. *Quant. Inf. Compt.*, 12:1034–1080, 2011.

[114] Adam Paetznick and Ben W. Reichardt. Universal fault-tolerant quantum computation with only transversal gates and error correction. *Phys. Rev. Lett.*, 111:090505, 2013.

[115] Fernando Pastawski and Beni Yoshida. Fault-tolerant logical gates in quantum error-correcting codes. *Phys. Rev. A*, 91:012305, 2015.

[116] Rajesh Pereira. *Trace vectors in matrix analysis*. PhD thesis, University of Toronto, 2003.

[117] Rajesh Pereira. Representing conditional expectations as elementary operators. *Proceedings of the American Mathematical Society*, 134:253–258, 2006.

[118] Denes Petz and Jonas Kahn. Complementary reductions for two qubits. *Journal of Mathematical Physics*, 48:012107, 2007.

[119] Arthur O. Pittenger and Morton H. Rubin. Mutually unbiased bases, generalized spin matrices and separability. *Linear Algebra and its Applications*, 390:255–278, 2004.

[120] John Preskill. Course notes, Quantum Computation. 2016.

[121] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Phys. Rev. Lett.*, 113(13):130503, 2014.

[122] Oded Regev and Liron Schiff. Impossibility of a quantum speed-up with a faulty oracle. In *International Colloquium on Automata, Languages, and Programming*, pages 773–781. Springer, 2008.

[123] Walter Rudin. *Fourier analysis on groups*. John Wiley & Sons, Toronto, 1990.

[124] Adel S. Sedra and Kenneth Carless Smith. *Microelectronic circuits*, volume 1. New York: Oxford University Press, 1998.

[125] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:2493, 1995.

[126] Peter W. Shor. Fault-tolerant quantum computation. *Proceedings., 37th Annual Symposium on Foundations of Computer Science*, pages 56–65, 1996.

[127] Andrew W. Steane. Error correcting codes in quantum theory. *Phys. Rev. Lett.*, 77:793, 1996.

[128] Andrew W. Steane. Multiple-Particle Interference and Quantum Error Correction. *Proc. Roy. Soc. Lond.*, 452:2551–2577, 1996.

[129] Andrew W. Steane. Quantum reed-muller codes. *arXiv: quant-ph/9608026*, 1996.

[130] Andrew W. Steane. Active stabilization, quantum computation, and quantum state synthesis. *Phys. Rev. Lett.*, 78(11):2252, 1997.

[131] Ashley M. Stephens, Zachary W. E. Evans, Simon J. Devitt, and Lloyd C. L. Hollenberg. Asymmetric quantum error correction via code conversion. *Phys. Rev. A*, 77(6):062335, 2008.

[132] W. Forrest Stinespring. Positive Functions on C*-algebras. *Proceedings of the American Mathematical Society*, pages 211–216, 1955.

[133] Krysta M. Svore, Andrew W. Cross, Isaac L. Chuang, and Alfred V. Aho. A flow-map model for analyzing pseudothresholds in fault-tolerant quantum computing. *Quant. Inf. Comput.*, 6:193–212, 2006.

[134] Maika Takita, A. D. Córcoles, Easwar Magesan, Baleegh Abdo, Markus Brink, Andrew Cross, Jerry M. Chow, and Jay M. Gambetta. Demonstration of weight-four parity measurements in the surface code architecture. *arXiv:1605.01351*, 2016.

[135] Q. A. Turchette, C. S. Wood, B. E. King, C. J. Myatt, D. Leibfried, W. M. Itano, C. Monroe, and D. J. Wineland. Deterministic entanglement of two trapped ions. *Phys. Rev. Lett.*, 81(17):3631, 1998.

[136] John Watrous. Course notes, Theory of Quantum Information. 2011.

[137] Mihály Weiner. On orthogonal systems of matrix algebras. *Linear Algebra and its Applications*, 433:520–533, 2010.

[138] Nathan Wiebe, Daniel Braun, and Seth Lloyd. Quantum algorithm for data fitting. *Phys. Rev. Lett.*, 109(5):050505, 2012.

[139] William K. Wootters and Wojciech H. Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802–803, 1982.

[140] Theodore J Yoder, Ryuji Takagi, and Isaac L Chuang. Universal fault-tolerant gates on nondegenerate stabilizer codes. *arXiv:1603.03948*, 2016.

[141] Bei Zeng, Xie Chen, and Isaac L. Chuang. Semi-clifford operations, structure of $\mathcal{C}_k$ hierarchy, and gate complexity for fault-tolerant quantum computation. *Phys. Rev. A*, 77:042313, 2008.

[142] Bei Zeng, Andrew W. Cross, and Isaac L. Chuang. Transversality Versus Universality for Additive Quantum Codes. *IEEE Transactions on Information Theory*, 57:6272–6284, 2011.

APPENDICES

Appendix A

Conditions on a generalized private quantum channel

The generalized private quantum channel on a two-qubit subspace encoding of a single qubit of information was shown in Equation 4.22 to have the following density matrix output:

$$|ij\rangle\langle kl|_{12}\mathrm{Tr}_E\big(|\alpha|^2|E^0_{ij}\rangle\langle E^0_{kl}| + \alpha\beta^*|E^0_{ij}\rangle\langle E^1_{kl}| + \alpha^*\beta|E^1_{ij}\rangle\langle E^0_{kl}| + |\beta|^2|E^1_{ij}\rangle\langle E^1_{kl}|\big).$$

For such an output to be private, the output state must encode no information about the input state of the channel, therefore must yield no information about the coefficients α and β. The cross terms ($\alpha\beta^*$ and $\alpha^*\beta$) must always be zero as there is no choice of non-zero overlap between the states $\left|E^0_{ij}\right\rangle$ and $|E^1_{kl}\rangle$ that will cancel out all information stored in $\alpha\beta^*$ and its complex conjugate for arbitrary choices of α and β. This should come as no surprise, as the complementary channel should be quantum error correcting for a subspace code, and an overlap of $\left|E^0_{ij}\right\rangle$ and $|E^1_{kl}\rangle$ would violate the Knill-Laflamme conditions.

For any non-zero $|\alpha|^2$ term $\langle E^0_{ij}|E^0_{kl}\rangle$, the corresponding $|\beta|^2$ term $\langle E^1_{ij}|E^1_{kl}\rangle$ must be the same as otherwise information about the relative magnitude of α and β will be contained in the output density matrix, yielding a non-private channel. Therefore, the set of conditions for a private quantum channel on a subspace encoding can be summarized by the following conditions on the environment states when considering the isometric extension:

$$\langle E^0_{ij}|E^0_{kl}\rangle = \langle E^1_{ij}|E^1_{kl}\rangle, \tag{A.1}$$

$$\langle E^0_{ij}|E^1_{kl}\rangle = 0, \tag{A.2}$$

for all choices of i, j, k, l.

The output density matrix in the case of a subsystem encoding, with respect to an enlarged set of environment states is given in Equation 4.23 as follows:

$$\begin{aligned}|ij\rangle\langle kl|_{12}\mathrm{Tr}_E\Big(&|\alpha|^2(|E^0_{ij0}\rangle\langle E^0_{kl0}| + |E^0_{ij1}\rangle\langle E^0_{kl1}|)\\ &+ \alpha\beta^*(|E^0_{ij0}\rangle\langle E^1_{kl0}| + |E^0_{ij1}\rangle\langle E^1_{kl1}|)\\ &+ \alpha^*\beta(|E^1_{ij0}\rangle\langle E^0_{kl0}| + |E^1_{ij1}\rangle\langle E^0_{kl1}|)\\ &+ |\beta|^2(|E^1_{ij0}\rangle\langle E^1_{kl0}| + |E^1_{ij1}\rangle\langle E^1_{kl1}|)\Big),\end{aligned}$$

The increased freedom in choosing this output state to be private comes from the fact that in considering the cross terms ($\alpha\beta^*$ and $\alpha^*\beta$), while the trace over the set of states corresponding to these terms must be zero, there can be cancellation between the two corresponding terms. Therefore, unlike the case of a subspace encoding, one could have $\langle E^0_{ij0}|E^1_{kl0}\rangle \neq 0$, however its corresponding pair must cancel the term out, that is $\langle E^0_{ij0}|E^1_{kl0}\rangle = -\langle E^0_{ij1}|E^1_{kl1}\rangle$. There is additional freedom in the diagonal terms in order for no information about the magnitude of the amplitudes of α and β to be present in the output density matrix. The result conditions for privatization are summarized as follows:

$$\langle E^0_{ij0}|E^0_{kl0}\rangle + \langle E^0_{ij1}|E^0_{kl1}\rangle = \langle E^1_{ij0}|E^1_{kl0}\rangle + \langle E^1_{ij1}|E^1_{kl1}\rangle, \tag{A.3}$$

$$\langle E^0_{ij0}|E^1_{kl0}\rangle = -\langle E^0_{ij1}|E^1_{kl1}\rangle, \tag{A.4}$$

for all choices of i, j, k, l.

Applying the above set of conditions to the case of the two-qubit dephasing channel Λ described throughout this work, we can show that there is insufficient freedom in a two-qubit subspace encoding to privatize a single encoded qubit. That is, no two-qubit subspace encoding will satisfy Equations A.1 and A.2 for the environment states produced by the two-qubit dephasing channel Λ.

Let the following parameters denote an arbitrary two-qubit subspace encoding:

$$\begin{aligned}|0_L\rangle &= \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle\\ |1_L\rangle &= \beta_{00}|00\rangle + \beta_{01}|01\rangle + \beta_{10}|10\rangle + \beta_{11}|11\rangle.\end{aligned}$$

By the uniqueness of the Stinespring dilation Theorem up to the preparation of the ancillary states, we assume that the form extension of the channel to unitary transformation on

a larger Hilbert space by preparing an additional pair of qubits in the $|+\rangle$ state and performing controlled–Z operations on each corresponding physical qubit in the encoding, as described in solid boxed operation in Figure 4.2. The resulting mapping of the logical states is given as follows:

$$|0_L\rangle = \sum_{ij} \alpha_{ij}|ij\rangle \longrightarrow \sum_{ij} \alpha_{ij}|ij\rangle Z^i|+\rangle Z^j|+\rangle$$
$$|1_L\rangle = \sum_{ij} \beta_{ij}|ij\rangle \longrightarrow \sum_{ij} \beta_{ij}|ij\rangle Z^i|+\rangle Z^j|+\rangle,$$

where the operation Z^i is applied to the state $|+\rangle$ depending on the value of the state on qubit 1, and similarly for Z^j and qubit 2. The resulting environment states therefore have the form

$$\left|E^0_{ij}\right\rangle = \alpha_{ij} Z^i|+\rangle Z^j|+\rangle,$$
$$\left|E^1_{ij}\right\rangle = \beta_{ij} Z^i|+\rangle Z^j|+\rangle.$$

The conditions set by Equation A.2 impose restrictions on the values of the coefficients in the subspace encodings. Since $\langle E^0_{ij}|E^1_{ij}\rangle = \alpha^*_{ij}\beta_{ij} = 0$, this implies either $\alpha_{ij} = 0$ or $\beta_{ij} = 0$. Without loss of generality, suppose $\alpha_{ij} = 0$, then the corresponding condition set by Equation A.1 imply $\langle E^0_{ij}|E^0_{ij}\rangle = |\alpha_{ij}|^2 = 0 = |\beta_{ij}|^2 = \langle E^1_{ij}|E^1_{ij}\rangle$. Thus, for all values (ij) the associated coefficients α_{ij} and β_{ij} will be equal to 0, implying that no private subspace encoding exists for the dephasing channel Λ that satisfy the set of conditions outlined by Equations A.1–A.2.

We now show that the set of conditions on a two-qubit subsystem encoding, by introducing a mixing ancilla, can be satisfied by the chosen encoding given by the first two boxes in Figure 4.2.

$$|0_L\rangle = \frac{1}{2}(|000\rangle + i|010\rangle + |101\rangle + i|111\rangle)$$
$$|1_L\rangle = \frac{1}{2}(|100\rangle - i|110\rangle + |001\rangle - i|011\rangle)$$

The resulting mapping as given by the isometric extension of the channel by introducing

two ancillary $|+\rangle$ states and controlled–Z operations will have the form

$$|0_L\rangle = \sum_{ijk} \gamma_{ijk}|ijk\rangle \longrightarrow \sum_{ijk} \gamma_{ijk}|ijk\rangle Z^i|+\rangle Z^j|+\rangle$$
$$|1_L\rangle = \sum_{ijk} \eta_{ijk}|ijk\rangle \longrightarrow \sum_{ijk} \eta_{ijk}|ijk\rangle Z^i|+\rangle Z^j|+\rangle,$$

resulting in the environment states

$$\left|E^0_{ijk}\right\rangle = \gamma_{ijk} Z^i|+\rangle Z^j|+\rangle$$
$$\left|E^1_{ijk}\right\rangle = \eta_{ijk} Z^i|+\rangle Z^j|+\rangle.$$

The environment states are orthogonal unless $(ij) = (kl)$, thus the set of conditions A.3–A.4 will be trivially satisfied unless $(ij) = (kl)$. Therefore let $(ij) = (kl)$, Equation A.3 then becomes

$$|\gamma_{ij0}|^2 + |\gamma_{ij1}|^2 = |\eta_{ij0}|^2 + |\eta_{ij1}|^2.$$

Each side of the above equation will have one non-zero term that will be equal to 1 as all the coefficients in the encoding are of equal magnitude, therefore Equation A.3 will always be satisfied. The condition set out by Equation A.4 will have the following form when $(ij) = (kl)$,

$$\gamma^*_{ij0}\eta_{ij0} = -\gamma^*_{ij1}\eta_{ij1},$$

yet since the logical states have support on differing computational basis states, both sides of the above equation will always be equal to zero as for any (ijm), $\gamma^*_{ijm}\eta_{ijm} = 0$.

Appendix B

Operator private quantum channels

The ideas in this appendix were originally published in Ref. [98], copyrighted by IOP Publishing Ltd. Jeremy Levick and Rajesh Pereira contributed the majority of the mathematical framework for the project. David Kribs, Raymond Laflamme, and myself contributed many of the ideas relaying the mathematical structure to physical systems as well as developed motivating examples. Jeremy Levick, Rajesh Pereira, and myself did the majority of the writing of the manuscript with all authors contributing to the editing and analysis of the results.

B.1 Introduction

This work began with an effort to generalize the example from Chapter 4 more widely; in particular to obtain a large class of simple and easily implemented n-qubit quantum channels, which as a consequence of their simplicity would not privatize any subspace, but that nevertheless do privatize quantum information through more delicate subsystem encodings. In addition to building on previous work, one could imagine such a class of channels and encodings as being useful in quantum computing when deleting information swiftly is a desired outcome. As an outgrowth of our example-based effort, we also uncover an advance in the general theory of private quantum subsystems. Specifically, we show how finite-dimensional private quantum subsystems can be cast in the framework of *quasiorthogonal operator algebras*, which have been studied in quantum information for their intrinsic interest [137, 112, 111, 118] and specific connection with identifying maximal collections of mutually unbiased bases [11, 119, 28]. Combining the quasiorthogonal perspective with

conditional expectation tools from operator algebra [117] and a group-theoretic analysis [123], we then show how quantum channels defined by operators forming an Abelian subgroup of the Pauli group can privatize quantum information encoded into subsystems of n-qubit Hilbert space, even though they can never privatize qubits directly encoded into subspaces of the Hilbert space. This gives a full picture of privatizing quantum information for such subgroups.

This Appendix organized as follows. In the next subsection we present preliminary material on private quantum subsystems. This is followed by a discussion on quasiorthogonal operator algebras and conditional expectations, and we indicate how private subsystems fit into the framework. In the next subsection we turn our attention to the structure of Abelian subgroups of the Pauli group. We conclude by explicitly showing how such subgroups can be used to privatize quantum information.

B.2 Private quantum subsystems & subspaces

The motivating communication scenario for a private quantum code starts with two parties, Alice and Bob, who share a private classical key that Alice uses to inform Bob which of a set of unitary operators $\{U_i\}$ she has used to encode her quantum state (density operator) $\rho \mapsto U_i \rho U_i^\dagger$. Bob can then decode and recover the state ρ without disturbing it. The set of unitaries $\{U_i\}$ and the probability distribution $\{p_i\}$ that makes up the random key which determines the encoding unitary are shared publicly. Hence an eavesdropper Eve's description of the system is given by the random unitary channel $\Phi(\rho) = \sum_i p_i U_i \rho U_i^\dagger$. By carefully selecting the unitary operators and probabilities, the random unitary channel will provide Eve with no information about the input state.

More generally, quantum channels are given by completely positive trace-preserving maps $\Phi : M_n(\mathbb{C}) \to M_n(\mathbb{C})$, and in the context of private communication one looks for sets S of density operators ρ that are mapped by Φ to the same state, $\Phi(\rho) = \rho_0$ for all $\rho \in S$. Analogous to quantum error correction, we focus on sets of states that are associated with underlying Hilbert space structure and hence form a true quantum code in the sense that superpositions of privatized states are also private. We define a qubit to be a two-dimensional subspace of $\mathbb{C}^n$ whose observables are given by the Pauli matrices in that fixed subspace,

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \; Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \; Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Thus the most basic choice for a set of private states is for S to be the set of density matrices whose non-zero action is entirely on a k-qubit, 2^k-dimensional subspace of the Hilbert space $\mathbb{C}^n$; i.e., there exists a subspace $\mathcal{C}$ such that the operators in S, when written in any basis containing $\mathcal{C}$, has the block form

$$\rho = \begin{pmatrix} \rho_{\mathcal{C}} & 0 \\ 0 & 0 \end{pmatrix}.$$

As a convenience we shall identify the full set of such operators with the set of operators $\mathcal{L}(\mathcal{C})$ that act on $\mathcal{C}$, which in this case is isomorphic to $\mathcal{L}(\mathbb{C}^{2^k})$ and thus Φ privatizes k qubits of information. Two of the simplest examples in the single qubit case are given by the complete depolarizing channel, which satisfies $\Phi(\rho) = \frac{1}{2}I$ and I is the identity operator, and the spontaneous emission channel, which satisfies $\Phi(\rho) = |0\rangle\langle 0|$ for all single qubit ρ [107]. The complete depolarizing channel in particular is the random unitary channel implemented by the equally weighted Pauli operators I, X, Y, Z.

However, not all quantum channels admit a private subspace. In particular, as proved in [76, 77], any quantum channel $\Phi(\rho) = \sum_i A_i \rho A_i^\dagger$ whose (normal) Kraus operators A_i are mutually commuting, cannot privatize a (multi-dimensional) subspace. Nevertheless, even in such instances it can still be possible to privatize quantum information. The following phase flip example was presented and discussed in detail in [76, 77]: Let $\Phi : \mathcal{L}(\mathbb{C}^2 \otimes \mathbb{C}^2) \to \mathcal{L}(\mathbb{C}^2 \otimes \mathbb{C}^2)$ be the completely positive map on two-qubit space whose Kraus operators are $\{\frac{1}{2}II, \frac{1}{2}ZI, \frac{1}{2}IZ, \frac{1}{2}ZZ\}$, where we have suppressed tensor symbols so that ZI is $Z \otimes I$, etc. These operators are all diagonal in the computational basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$, and form an orthonormal basis for the diagonal subalgebra of $M_4(\mathbb{C})$ in the normalized trace (Hilbert-Schmidt) inner product. In particular they are mutually commuting and so the channel Φ has no private subspaces. However, the subalgebra spanned by II, IX, YY, YZ is privatized by Φ, and moreover, is isomorphic to the Pauli algebra on one qubit; thus the channel still privatizes a qubit:

$$\Phi\Big(\frac{1}{4}(II + c_2 IX + c_3 YY + c_4 YZ)\Big) = \frac{1}{4}II.$$

This example relies on the more subtle notion of quantum subsystem to privatize a qubit. A space A (or B) is a *subsystem* of a Hilbert space $\mathcal{H}$ if there is a subspace $\mathcal{C}$ of $\mathcal{H}$ such that $\mathcal{C}$ admits a tensor decomposition as $\mathcal{C} = A \otimes B$.

Definition 68. *Let $\Phi : \mathcal{L}(\mathcal{H}) \to \mathcal{L}(\mathcal{H})$ be a channel and let B be a subsystem of $\mathcal{H}$. Then B is a* private subsystem *for Φ if there is a $\rho_0 \in \mathcal{L}(\mathcal{H})$ and $\sigma_A \in \mathcal{L}(A)$ such that*

$$\Phi(\sigma_A \otimes \sigma_B) = \rho_0 \quad \forall \sigma_B \in \mathcal{L}(B). \tag{B.1}$$

As shown in [76, 77], the above example channel privatizes a qubit subsystem in the sense of Eq. (B.1), yet does not privatize a subspace. Moreover, the mapping is "non-operator" in the language of [76, 77] in the sense that Eq. (B.1) holds for a particular state $\sigma_A \in \mathcal{L}(A)$ but not for every state on A. We seek to understand this example and generalize it, and we shall make use of conditional expectations and quasiorthogonality of two subalgebras, notions to which we now turn.

B.3 Quasiorthogonal subalgebras & conditional expectations

We begin this subsection by reviewing basic notions from operator algebras and then we indicate how private subsystems can be cast in the quasiorthogonal subalgebra framework and discuss examples.

A *unital subalgebra* of the $N \times N$ complex matrices $M_N(\mathbb{C})$ is a subset $\mathcal{A} \subseteq M_N(\mathbb{C})$ that is closed under matrix addition, scalar multiplication, and multiplication, and includes the identity matrix I_N, which we denote by I when the dimension is clear. It follows from the representation theory of finite-dimensional C*-algebras [48], that any unital subalgebra $\mathcal{A}$ of $M_N(\mathbb{C})$ that is also closed under taking adjoints is unitarily equivalent to an algebra of the form

$$\bigoplus_{i=1}^{m} I_{k_i} \otimes M_{q_i}(\mathbb{C}), \tag{B.2}$$

where $\sum_i k_i q_i = N$.

Definition 69. *Let $\mathcal{A}$ be a unital subalgebra of $M_N(\mathbb{C})$. Then $\Phi_{\mathcal{A}} : M_N(\mathbb{C}) \to M_N(\mathbb{C})$ is called the* trace-preserving conditional expectation onto $\mathcal{A}$ *when it is the unique completely positive trace-preserving map satisfying*

1. $\Phi_{\mathcal{A}}(a) = a \quad \forall a \in \mathcal{A}$
2. $\Phi_{\mathcal{A}}(a_1 x a_2) = a_1 \Phi_{\mathcal{A}}(x) a_2 \quad \forall a_1, a_2 \in \mathcal{A}, \quad \forall x \in M_N(\mathbb{C})$
3. $\Phi_{\mathcal{A}}(x) \geq 0 \quad \forall x \geq 0.$

The trace-preserving conditional expectation onto a unital subalgebra is unique as a linear map, and is always a completely positive map. Any unital subalgebra induces an

inner product, the *left-regular trace inner product*, which is the Hilbert-Schmidt inner product in the induced left-regular trace given by the subalgebra $\mathcal{A}$. More explicitly, let $a \in \mathcal{A}$ and consider the left-regular representation of a, $L_a(x) = ax$ for $x \in M_N(\mathbb{C})$. The left-regular trace of a is the trace of the operator L_a; $tr_{\mathcal{A}}(a) := tr(L_a)$. This gives the left-regular trace inner product as $\langle a, b \rangle_{\mathcal{A}} = tr_{\mathcal{A}}(a^* b)$.

Denote by $\mathcal{A}'$ the *commutant* of $\mathcal{A}$; $\mathcal{A}' = \{x \in M_N(\mathbb{C}) : xa = ax \quad \forall a \in \mathcal{A}\}$. Importantly, the Kraus operators for $\Phi_{\mathcal{A}}$ must be an orthonormal basis for $\mathcal{A}'$ in the left-regular inner product induced by $\mathcal{A}$. In addition, recall that two different choices of Kraus operators must be related by a partial isometry conjugation. For more on conditional expectations onto matrix algebras, including for proofs and further references, see [117].

For the phase flip example discussed in the previous subsection, the map $\Phi = \Phi_{\Delta_4}$ is the trace-preserving conditional expectation onto the 4×4 diagonal subalgebra, $\Delta_4 = \{\sum_{i,j=0}^{1} a_{ij} |ij\rangle\langle ij| : a_{ij} \in \mathbb{C}\} \cong \mathbb{C} \oplus \mathbb{C} \oplus \mathbb{C} \oplus \mathbb{C}$. Since $\Delta_4' = \Delta_4$, the Kraus operators for Φ should be a basis for the diagonal matrices, and indeed they are.

Geometrically, quasiorthogonal algebras arise through a natural broadening of the notion of orthogonality for algebras; namely, unital algebras $\mathcal{A}$, $\mathcal{B}$ satisfy quasiorthogonality if $tr(ab) = 0$ whenever $tr(a) = 0 = tr(b)$ and $a \in \mathcal{A}$, $b \in \mathcal{B}$. The algebras are not orthogonal themselves as they both contain the identity operator, however the sets $\mathcal{A} \ominus \mathbb{C}I$ and $\mathcal{B} \ominus \mathbb{C}I$ are orthogonal in the Hilbert-Schmidt inner product $< a, b >= tr(b^* a)$. We state equivalent forms of this definition in our current notation.

Definition 70. *Two unital subalgebras $\mathcal{A}, \mathcal{B} \subseteq M_N(\mathbb{C})$ are said to be* quasiorthogonal *if any of the following equivalent conditions hold:*

1. $tr\big((a - \frac{tr(a)}{N} I)(b - \frac{tr(b)}{N} I)\big) = 0 \quad \forall a \in \mathcal{A}, \quad \forall b \in \mathcal{B}$
2. $\frac{1}{N} tr(ab) = \frac{1}{N} tr(a) \frac{1}{N} tr(b) \quad \forall a \in \mathcal{A}, \quad \forall b \in \mathcal{B}$
3. $\Phi_{\mathcal{A}}(b) = \frac{tr(b)}{N} I$ *and* $\Phi_{\mathcal{B}}(a) = \frac{tr(a)}{N} I \quad \forall a \in \mathcal{A}, \quad \forall b \in \mathcal{B}$
4. $\Phi_{\mathcal{A}} \circ \Phi_{\mathcal{B}}(\rho) = \Phi_{\mathcal{B}} \circ \Phi_{\mathcal{A}}(\rho) = \frac{tr(\rho)}{N} I$ *for all* ρ.

In particular, condition (3) says that the trace-preserving conditional expectation onto $\mathcal{A}$ privatizes an element of $\mathcal{B}$ (and vice-versa) precisely when the two subalgebras are quasiorthogonal to each other.

In the extremal case with $\mathcal{A} = M_N(\mathbb{C})$ and $\mathcal{B} = \mathbb{C}I = \mathcal{A}'$, $\Phi_{\mathcal{B}}$ is the complete depolarizing channel with Kraus operators given by a complete orthonormal set (in the

Hilbert-Schmidt inner product) of operators, and $\Phi_{\mathcal{B}}(a) = \frac{tr(a)}{N} I$ for all $a \in \mathcal{A}$. As a simple subsystem example, consider the first qubit algebra $\mathcal{A}$ in two-qubit space generated by $\{II, XI, YI, ZI\}$. The commutant $\mathcal{A}' = I_2 \otimes M_2(\mathbb{C})$ is generated by the orthonormal set $\{II, IX, IY, IZ\}$, which also act as Kraus operators (after normalizing) for the map $\Phi_{\mathcal{A}}$. Thus, $\mathcal{A}$ and $\mathcal{A}'$ form a quasiorthogonal pair and in particular the second qubit is a private subsystem for the channel $\Phi_{\mathcal{A}}$. Not all quasiorthogonal algebra pairs arise through the commutant in this way though; indeed, as an example note that the phase flip example above is determined by the algebra pairing $\mathcal{A} = \Delta_4$ and $\mathcal{B}$ generated by $\{II, IX, YY, YZ\}$, which is unitarily equivalent to $I_2 \otimes M_2(\mathbb{C})$.

This behaviour is not unique to qubit systems and can be generalized to multi-dimensional systems. The following example exhibits the behaviour of privatized sub-algebras for elementary systems composed of qutrits.

Example 71. *Let* $X = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$ *and* $Z = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \omega & 0 \\ 0 & 0 & \omega^2 \end{pmatrix}$, $\omega = e^{2i\pi/3}$, *be the generalized Pauli operators for qutrits. Consider the channel on two qutrits given by*

$$\Phi(\rho) = \frac{1}{9} \sum_{i,j=0}^{2} (X^{2i} Z^i \otimes X^j Z^j) \rho (X^{2i} Z^i \otimes X^j Z^j)^\dagger.$$

The Kraus operators for this channel are Abelian, and thus do not admit a private subspace. However, one may check that the sub-algebra $\mathcal{A}$ *generated by* $X^2 \otimes X, XZ^2 \otimes Z$ *is privatized by* Φ*, and moreover that* $\mathcal{A}$ *is isomorphic as an algebra to the one-qutrit sub-algebra generated by* $I \otimes X$, $I \otimes Z$.

Proof. Explicitly, $X^3 = Z^3 = I$, and $XZ = \omega ZX$, so the group generated by X, Z has a bicharacter given by $[g, h] = \chi(g, h) I$, yielding the following character matrix:

$$F = \begin{array}{c} \\ I \\ X \\ X^2 \\ Z \\ XZ \\ X^2Z \\ Z^2 \\ XZ^2 \\ X^2Z^2 \end{array} \begin{array}{c} \begin{array}{ccccccccc} I & X & X^2 & Z & XZ & X^2Z & Z^2 & XZ^2 & X^2Z^2 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & \omega & \omega & \omega & \omega^2 & \omega^2 & \omega^2 \\ 1 & 1 & 1 & \omega^2 & \omega^2 & \omega^2 & \omega & \omega & \omega \\ 1 & \omega^2 & \omega & 1 & \omega^2 & \omega & 1 & \omega^2 & \omega \\ 1 & \omega^2 & \omega & \omega & 1 & \omega^2 & \omega^2 & \omega & 1 \\ 1 & \omega^2 & \omega & \omega^2 & \omega & 1 & \omega & 1 & \omega^2 \\ 1 & \omega & \omega^2 & 1 & \omega & \omega^2 & 1 & \omega & \omega^2 \\ 1 & \omega & \omega^2 & \omega & \omega^2 & 1 & \omega^2 & 1 & \omega \\ 1 & \omega & \omega^2 & \omega^2 & 1 & \omega & \omega & \omega^2 & 1 \end{pmatrix} \end{array} \tag{B.3}$$

In two qutrit space, the matrix of commutation relations is $F \otimes F$; any all-1s submatrix of $F \otimes F$ yields an Abelian subalgebra. Our choice corresponds to the tensor product of the $\{1, 6, 8\}$ submatrix of F with itself union the tensor product of the $\{1, 5, 9\}$ submatrix of F with itself.
To find a quasiorthogonal algebra to this Abelian algebra, we take a quotient by the subalgebra to obtain the following equivalence classes:

$I \otimes I$	$I \otimes X$	$I \otimes X^2$	$X \otimes I$	$X^2 \otimes I$
$I \otimes XZ$	$I \otimes X^2Z$	$I \otimes Z$	$X \otimes XZ$	$X^2 \otimes XZ$
$I \otimes X^2Z^2$	$I \otimes Z^2$	$I \otimes XZ^2$	$X \otimes X^2Z^2$	$X^2 \otimes X^2Z^2$
$X^2Z \otimes I$	$X^2Z \otimes X$	$X^2Z \otimes X^2$	$Z \otimes I$	$XZ \otimes I$
$X^2Z \otimes XZ$	$X^2Z \otimes X^2Z$	$X^2Z \otimes Z$	$Z \otimes XZ$	$XZ \otimes XZ$
$X^2Z \otimes X^2Z^2$	$X^2Z \otimes Z^2$	$X^2Z \otimes XZ^2$	$Z \otimes X^2Z^2$	$XZ \otimes X^2Z^2$
$XZ^2 \otimes I$	$XZ^2 \otimes X$	$XZ^2 \otimes X^2$	$X^2Z^2 \otimes I$	$Z^2 \otimes I$
$XZ^2 \otimes XZ$	$XZ^2 \otimes X^2Z$	$XZ^2 \otimes Z$	$X^2Z^2 \otimes XZ$	$Z^2 \otimes XZ$
$XZ^2 \otimes X^2Z^2$	$XZ^2 \otimes Z^2$	$XZ^2 \otimes XZ^2$	$X^2Z^2 \otimes X^2Z^2$	$Z^2 \otimes X^2Z^2$

$X \otimes X$	$X \otimes X^2$	$X^2 \otimes X$	$X^2 \otimes X^2$
$X \otimes X^2Z$	$X \otimes Z$	$X^2 \otimes X^2Z$	$X^2 \otimes Z$
$X \otimes Z^2$	$X \otimes XZ^2$	$X^2 \otimes Z^2$	$X^2 \otimes XZ^2$
$Z \otimes X$	$Z \otimes X^2$	$XZ \otimes X$	$XZ \otimes X^2$
$Z \otimes X^2Z$	$Z \otimes Z$	$XZ \otimes X^2Z$	$XZ \otimes Z$
$Z \otimes Z^2$	$Z \otimes XZ^2$	$XZ \otimes Z^2$	$XZ \otimes XZ^2$
$X^2Z^2 \otimes X$	$X^2Z^2 \otimes X^2$	$Z^2 \otimes X$	$Z^2 \otimes X^2$
$X^2Z^2 \otimes X^2Z$	$X^2Z^2 \otimes Z$	$Z^2 \otimes X^2Z$	$Z^2 \otimes Z$
$X^2Z^2 \otimes Z^2$	$X^2Z^2 \otimes XZ^2$	$Z^2 \otimes Z^2$	$Z^2 \otimes XZ^2$

Notice that cosets 3, 5, 8, and 9 are obtained by squaring cosets 2, 4, 7 and 6 respectively, so a choice of two generators for a quasiorthogonal subalgebra should make sure to avoid these pairs. Our choice corresponds to choosing $X^2 \otimes X$ from coset 8 and $XZ^2 \otimes Z$ from coset 3. To see that this choice of generators is isomorphic to a qutrit algebra, note that the block unitary

$$U = \left(\begin{array}{c|c|c} I & X^2Z^2 & XZ \\ \hline XZ^2 & Z & X^2 \\ \hline X^2Z & X & Z \end{array}\right)$$

acts by on $I \otimes X$ and $I \otimes Z$ by

$$U(I \otimes X)U^* = \omega^2 XZ^2 \otimes Z$$
$$U(I \otimes Z)U^* = X^2 \otimes X$$

□

As a next-step generalization of our motivating phase flip example, for the rest of the paper we consider conditional expectations onto algebras generated by Abelian subgroups of the n-qubit Pauli group. The subalgebras generated in this way will be Abelian, so by choosing a maximal Abelian subgroup of the n-qubit Pauli group that contains the initial subgroup, we obtain a maximal Abelian subalgebra generated by Pauli operators. The commutant of such an algebra will be itself, and so the Kraus operators for the trace-preserving conditional expectation will simply be the elements of the Abelian group, suitably weighted. Thus, having mutually commuting Kraus operators, there can be no private subspace for such a channel. However, if the Abelian subalgebra has a quasiorthogonal partner with a non-scalar component as in Eq. (B.2), the channel will still have a private subsystem. This motivates us to learn more about the structure of maximal Abelian subgroups of the n-qubit Pauli group, $\mathcal{P}_n$, a topic to which we now turn.

B.4 Commutation relations of the Pauli group

Given $n \geq 1$, let $\mathcal{P}_n$ be the n-qubit Pauli group, which is the unitary subgroup of $M_{2^n}(\mathbb{C})$ generated by all n-fold tensor products of the Pauli matrices X, Y, Z.

Note that the group-theoretic commutator $[\sigma, \tau] = \sigma\tau\sigma^{-1}\tau^{-1}$ (as opposed to the more typically considered Lie algebra commutator) of any two n-qubit Pauli matrices is equal to $\pm I$. Also, $[\sigma, \tau] = [c_1\sigma, c_2\tau]$ for $\sigma, \tau \in \mathcal{P}_n$ and $c_1, c_2 \in \{\pm 1, \pm i\}$. Hence, the function $\chi : P_n \times P_n \to \mathbb{C}$ defined by $\chi([\sigma], [\tau])I = [\sigma, \tau]$ is well-defined on the central quotient $P_n = \mathcal{P}_n/\{\pm I, \pm iI\}$. Notice that P_1 is isomorphic to the Klein four-group $V = \{e, v_1, v_2, v_3\}$ where e is the identity element, each v_i is its own inverse, and $v_i v_j = v_k$. Similarly, $P_n \simeq V^n$, where V^n is the (Abelian) direct product of n copies of the Klein four-group with itself.

We recall that a *bicharacter* on a group G is a function $B(\cdot, \cdot) : G \times G \to \mathbb{C}$ satisfying the following:

1. $B(e, g) = B(g, e) = 1$ for all $g \in G$,
2. $B(g, hk) = B(g, h)B(g, k)$ and $B(hk, g) = B(h, g)B(k, g)$ $\forall$ $g, h, k \in G$.

Moreover, a bicharacter is non-degenerate if for all non-identity $g \in G$ there exists some element $h \in G$ such that $B(g, h) \neq 1$.

Notice that a bicharacter, when restricted in either argument to a fixed $g \in G$ becomes a character on G.

Proposition 5. *The function $\chi(\cdot,\cdot)$ defined above is a non-degenerate bicharacter on P_n.*

Proof. Firstly, $\chi([I],[\sigma]) = [I,\sigma] = I$, hence $\chi([I],[\sigma]) = 1$ and similarly $\chi([\sigma],[I]) = 1$. To prove the second condition holds, note that $\sigma\rho = \chi([\sigma],[\rho])\rho\sigma$ and $\tau\rho = \chi([\tau],[\rho])\tau\sigma$. Hence

$$\begin{aligned} \sigma\tau\rho &= \chi([\sigma\tau],[\rho])\rho(\sigma\tau) \\ &= \chi([\tau],[\rho])\sigma\rho\tau \\ &= \chi([\tau],[\rho])\chi([\sigma],[\rho])\rho\sigma\tau, \end{aligned}$$

and the other identity is similarly proved. The claim that $\chi(\cdot,\cdot)$ is non-degenerate is equivalent to the claim that for any non-identity $[\sigma] \in P_n$, there is some $[\tau] \in P_n$ that anti-commutes with $[\sigma]$, which is easily seen to be the case. □

Consider the character matrix for P_1 given by

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix},$$

where the columns and rows index the Pauli matrices and the entries of H record whether elements of the Pauli basis commute or anti-commute. Then H is the so-called bicharacter matrix of $\chi(\cdot,\cdot)$ on P_1; that is $H_{\sigma,\tau} = \chi([\sigma],[\tau])$. Moreover, since the restriction of $\chi([\sigma],[\tau])$ to any particular σ yields a character of the Abelian group P_1, H is a character matrix.

The group P_1 is isomorphic to the Klein four-group, $V = \{e, x, y, z\}$, an Abelian group defined by $x^2 = y^2 = z^2 = e$ and $xy = z$, $xz = y$, $yz = x$. Thus, P_n is isomorphic to V^n, the direct product of V with itself n times, and $H_n := H^{\otimes n}$ is a character matrix for the Abelian group P_n, and records the commutation relations between basis elements for the n-qubit Pauli group. Lastly, we point out that a maximal Abelian subgroup of P_n corresponds to a maximal submatrix of $H^{\otimes n}$ containing all 1's.

We recall another notion from group theory to continue. Let G be a group with dual group $\hat{G}$ and K a subgroup of G. Then the *annihilator* of K in G, $Ann_G(K)$ is the set of characters $\chi_i \in \hat{G}$ satisfying $\chi_i(k) = 1$ for all $k \in K$. We cite a well-known fact about the annihilator subgroup [123].

Lemma 72. *Let $G, K, \widehat{G}$ be as above; then $Ann_G(K)$ is isomorphically homeomorphic to the dual group of G/K.*

In particular, this implies that when G is finite and Abelian, that $|Ann_G(K)| = |G/K| = |G|/|K|$. The following is a direct consequence of this statement.

Corollary 73. *Let $G = P_n$ and K be any subgroup of P_n. Let I be column indices of $H^{\otimes n}$ associated to the elements of K. Let $J = \chi^{-1}(Ann_G(K))$ be the row indices associated to the annihilator of K. Then we have $|I||J| = 4^n$.*

Corollary 74. *Let K be an Abelian subgroup of P_n with $|K| = 2^k < 2^n$. Then K can be extended to an Abelian subgroup of size 2^n.*

Proof. We will show that so long as $k < n$, there exists a $g \in P_n \setminus K$ such that g commutes with every element of K; and so that $< K, g >$, the group spanned by K and g, must be Abelian.

We use the previous corollary as follows: let I be the indices associated to the elements of K; since K is Abelian the submatrix $H_n[I]$ is an all 1 submatrix. Thus, by the previous result, there exist $4^n|I|^{-1} = 4^n 2^{-k}$ rows whose intersection with each column in I contains only 1's. As $k < n$, we have $2^k < \frac{4^n}{2^k}$, and so there is at least one row not already in I with this property. Call this row i. By the symmetry of H_n, the intersection of column i with the rows of I is all 1's, and $H_{n_{ii}=1}$, and so the submatrix indexed by $I \cup \{i\}$ is an all 1 submatrix. Let g be the element of P_n associated with the i^{th} column; then $g \in P_n$ is an element of $P_n \setminus K$ that commutes with every element of K. Hence $< K, g >$ is an Abelian group of size at least 2^{k+1}.

We can iterate the procedure until $\frac{4^n}{2^k} = 2^k$, or $k = n$. □

We are now ready to prove the following theorem which will be useful in constructing examples that generalize our motivating phase flip example.

Theorem 75. *Any maximal Abelian subalgebra $\mathcal{A}$ of $M_{2^n}(\mathbb{C})$ generated by elements of $\mathcal{P}_n$ has dimension 2^n.*

Proof. To begin, we note that if $\sigma \in \mathcal{P}_n$ is an element of $\mathcal{A}$, then so are $-\sigma, \pm i\sigma$, and hence independent generators of $\mathcal{A}$ must all come from different conjugacy classes. Thus, we may regard our generators as coming from P_n, rather than $\mathcal{P}_n$. It follows that a generating set for a maximal Abelian subalgebra is simply a maximal Abelian subgroup $K \leq P_n$, which by the previous corollary has size 2^n. Since such a subgroup is already closed under matrix multiplication, we have $\dim < K >= 2^n$. □

Corollary 76. *Any subalgebra $\mathcal{A}$ contained in the span of elements from P_n has the property that $|\mathcal{A}||\mathcal{A}'| = 4^n$.*

Now we may turn our attention to the private structure of quantum channels whose normalized Kraus operators form an Abelian subgroup of the Pauli group.

B.5 Privatizing qubits with Abelian subgroups

We begin by discussing the simplest case, the channel whose Kraus operators are a normalized basis for the diagonal algebra, on arbitrary n-qubit space.

Let Δ_{2^n} be the diagonal algebra on $2^n \times 2^n$ complex matrices, with basis the group Δ generated by $\{Z_1, Z_2, \cdots, Z_n\}$ where $Z_1 = Z \otimes I \otimes \ldots \otimes I$, etc. Then $\Delta'_{2^n} = \Delta_{2^n}$, and thus the conditional expectation $\Phi_{\Delta_{2^n}}$ onto Δ_{2^n}, has as its Kraus operators the elements of Δ, multiplied by $\frac{1}{2^{n/2}}$.

Of course, we now know the answer is any algebra quasiorthogonal to the diagonal algebra Δ_{2^n}. We can give an abstract description of such algebras, and also a concrete construction. The abstract result follows from the following result of [116] and provides an upper bound on the number of privatized qubits.

Lemma 77. *Let $\mathcal{A}$ be a subalgebra of $M_N(\mathbb{C})$ unitarily equivalent to $\bigoplus_{k=1}^m I_{i_k} \otimes M_{j_k}(\mathbb{C})$. Then the following conditions are equivalent:*

1. *$\mathcal{A}$ is quasiorthogonal to Δ_N;*
2. *$i_k \geq j_k$ for all $1 \leq k \leq m$.*

As k-qubits can be encoded into the unital algebra $I_{2^{n-k}} \otimes M_{2^k}(\mathbb{C})$, the conditional expectation onto Δ_{2^n} can privatize k qubits if and only if $n - k \geq k$. In other words, the conditional expectation onto Δ_{2^n} can privatize at most $\lfloor \frac{n}{2} \rfloor$ qubits.

One explicit construction of such a subalgebra is to make use of the private subalgebra from our motivating example. For $1 \leq i \leq \lfloor \frac{n}{2} \rfloor$ let $\hat{X}_i = \bigotimes_{k=1}^n \sigma_k$, where

$$\sigma_k = \begin{cases} X & \text{if } k = 2i \\ I & \text{otherwise} \end{cases}$$

and let $\hat{Y}_i = \bigotimes_{k=1}^{n} \sigma_k$, where

$$\sigma_k = \begin{cases} Y & \text{if } k = 2i-1, 2i \\ I & \text{otherwise} \end{cases}$$

Then $\mathcal{B} = Alg\{\hat{X}_i, \hat{Y}_j : 1 \leq i, j \leq \lfloor \frac{n}{2} \rfloor\}$, where $Alg\{S\}$ is the unital algebra generated by operators in S, is quasiorthogonal to Δ_{2^n}, precisely because $\hat{X}_i, \hat{Y}_i$ and all their products are never diagonal, unless they are the identity. This algebra encodes $\lfloor \frac{n}{2} \rfloor$ qubits in the obvious way, where $\hat{X}_i$, and $\hat{Y}_i$ act as the Pauli matrices X_i, Y_i on $\lfloor \frac{n}{2} \rfloor$ qubits with an X or Y in the i^{th} tensor spot respectively.

We may use this explicit construction to prove the following in the general maximal Abelian case.

Theorem 78. *Let Φ be a a completely positive trace-preserving map on $M_{2^n}(\mathbb{C})$ whose Kraus operators are equally weighted elements of a maximal Abelian subgroup $G \leq \mathcal{P}_n$. Then Φ can privatize $\lfloor \frac{n}{2} \rfloor$ qubits.*

Proof. Let $\mathcal{A} = Alg\{G\}$ be the algebra generated by G. Then $\mathcal{A}$ is a maximal Abelian subalgebra of $M_N(\mathbb{C})$. Thus $\mathcal{A}' = \mathcal{A}$. The elements of G are clearly a basis for $\mathcal{A}$, and hence Φ is the trace-preserving conditional expectation onto $\mathcal{A}' = \mathcal{A}$. Moreover, since $\mathcal{A}$ is maximal Abelian, by simultaneously diagonalizing, there is a change of basis such that $U\mathcal{A}U^* = \Delta_{2^n}$. Denote by $\mathcal{B}$ the subalgebra generated by $\{\hat{X}_i, \hat{Y}_j\}$ where $\hat{X}_i$ and $\hat{Y}_j$ are as above. Then $U^*\mathcal{B}U$ is privatized by Φ; and following the discussion above, we see that $\mathcal{B}$ is sufficient to encode $\lfloor \frac{n}{2} \rfloor$ qubits. □

We now examine the case that $\mathcal{A}$ is a non-maximal Abelian algebra generated by Pauli operators. Corollary 76 says that for any subalgebra $\mathcal{A}$ generated by elements of $\mathcal{P}_n$, $\dim(\mathcal{A})\dim(\mathcal{A}') = 4^n = \dim(\mathcal{H})^2$. The algebras for which this equality holds can be characterized as follows from [116].

Lemma 79. *Let $\mathcal{A}$ be a unital subalgebra of $M_n(\mathbb{C})$, then the following are equivalent:*

1. $\dim(\mathcal{A})\dim(\mathcal{A}') = N^2$
2. $\mathcal{A} = U(\bigoplus_{i=1}^{m} I_{sk_i} \otimes M_{rk_i})U^*$ *for some unitary matrix U*

Lemma 80. *Every commutative subalgebra $\mathcal{A} \subseteq M_N(\mathbb{C})$ generated by Pauli operators is unitarily equivalent to one of the form $I_{2^{n-k}} \otimes Alg\{Z_i\}_{i=1}^{k}$ for some $k \leq n$.*

Proof. By Lemma 79, any commutative subalgebra $\mathcal{A}$ generated by commuting Pauli operators satisfies $rk_i = 1$ for all i, and hence for some s, decomposes as

$$\mathcal{A} = U(\bigoplus_{i=1}^{m} I_s \otimes \mathbb{C})U^* = U(I_{2^{n-k}} \otimes \Delta_{2^k})U^*,$$

for some k. Finally, as $\{Z_i\}_{i=1}^k$ generate Δ_{2^k}, we have that $I_{2^{n-k}} \otimes Alg\{Z_i\}_{i=1}^k = I_{2^{n-k}} \otimes \Delta_{2^k}$. □

Theorem 81. *Let $\Phi : M_N(\mathbb{C}) \to M_N(\mathbb{C})$ be a completely positive map whose Kraus operators $\{K_i\}_{i=1}^{2^k}$ are equally weighted members of an Abelian subgroup of $\mathcal{P}_n$. Then there is a $\frac{k}{2}$-qubit algebra privatized by Φ.*

Proof. By Lemma 80, the algebra generated by the Kraus operators must be unitarily equivalent to $I_{2^{n-k}} \otimes \Delta_{2^k}$; hence after diagonalizing and restricting to the subalgebra spanned by only the last k qubits, we obtain the maximal Abelian subalgebra on k qubits; theorem 78 tells us how to privatize $\frac{k}{2}$ qubits in this scheme. Tensoring any private algebra encoding $\frac{k}{2}$ qubits for the $2^k \times 2^k$ maximal Abelian case with a $2^{n-k} \times 2^{n-k}$ identity yields an algebra encoding $\frac{k}{2}$ qubits that is private for Φ. □

The results in this Subsection can be thought of as the subsystem analog of the results from Ref. [5], where there they showed that k unitaries could be used to privatize a $\frac{k}{2}$-qubit subspace. Here, we have shown that we can privatize a $\frac{k}{2}$-qubit subsystem algebra using k elements from an Abelian subgroup of $\mathcal{P}_n$.

B.6 Summary

In this work, we demonstrate the underlying mathematical principles behind private quantum subsystems. Namely, by using the theory of quasiorthogonal algebras and developing the set of tools therein, we can make definite statements on the dimension of the subsystems that can be privatized by commuting Kraus operators. We show that by taking elements of a maximal Abelian subgroup G of the n-qubit Pauli group $\mathcal{P}_n$, the elements of such a group can privatize $\lfloor \frac{n}{2} \rfloor$ qubits when used as equally weighted Kraus operators of the channel. Moreover, we show that this result can be generalized to fewer qubits when the size of the Abelian subgroup is not taken to be maximal.

Appendix C

A simple decoherence model for bucket brigade qRAM

Let us consider the error model considered in [122],

$$\mathcal{R}_p(\rho) := (1-p)\hat{O}\rho\hat{O}^\dagger + p\rho, \tag{C.1}$$

with $\hat{O}$ denoting the perfect oracle for quantum searching and let us define

$$\mathcal{D}_q(\rho) := (1-q)\rho + q\vec{X}\rho\vec{X}^\dagger, \tag{C.2}$$

as the multi-qubit bit-flip channel where $\vec{X}$ is a shorthand notation for a tensor product of σ_X bit-flip operators acting on some fixed subset of the qubits. The proof technique presented below for $\mathcal{D}_q$ also applies to the case of multi-qubit dephasing channels).

The error model proposed in this paper (see Eq. (5.9)) is

$$O(\rho) := p_{rp}\hat{O}\rho\hat{O}^\dagger + p_{wp}\mathcal{E}_{wp}(\rho) + p_{np}\mathcal{E}_{np}(\rho). \tag{C.3}$$

We show that the composition $\mathcal{R}_p \circ \mathcal{D}_q$ resembles (although not exactly the same) our error model Eq. (5.9), for suitable chosen p and q. It follows immediately that the $\Omega(N)$ lower bound for the searching algorithm considered in [122] is also a lower bound for the composition $\mathcal{R}_p \circ \mathcal{D}_q$, since channel composition cannot decrease the query complexity (one can simply incorporate $\mathcal{D}_q$ into an appropriate unitary for the $\mathcal{R}_p$ algorithm). A simple calculation yields:

$$\begin{aligned}
\mathcal{R}_p \circ \mathcal{D}_q(\rho) &= (1-p)\hat{O}\mathcal{D}_q(\rho)\hat{O}^\dagger + p\mathcal{D}_q(\rho) \\
&= (1-p)(1-q)\hat{O}\rho\hat{O}^\dagger + (1-p)q\hat{O}(\vec{X}\rho\vec{X}^\dagger)\hat{O}^\dagger + p(1-q)\rho + pq\vec{X}\rho\vec{X}^\dagger \\
&= (1-p)(1-q)\hat{O}\rho\hat{O}^\dagger + (1-p)q\hat{O}(\vec{X}\rho\vec{X}^\dagger)\hat{O}^\dagger + p\mathcal{D}_q(\rho).
\end{aligned} \tag{C.4}$$

We now identify the coefficients in Eq. (C.3) and Eq. (C.4)

$$\begin{cases} p_{rp} = (1-p)(1-q) \\ p_{wp} = (1-p)q \\ p_{np} = p, \end{cases} \tag{C.5}$$

and note that for any given probabilities p_{rp}, p_{wp}, p_{np} satisfying $p_{rp}+p_{wp}+p_{np} = 1$, equations Eq. (C.5) have the solution

$$\begin{cases} p = p_{np} \\ q = \dfrac{p_{wp}}{p_{wp}+p_{rp}}. \end{cases} \tag{C.6}$$

We can therefore write

$$\mathcal{R}_p \circ \mathcal{D}_q(\rho) = p_{rp}\hat{O}\rho\hat{O}^\dagger + p_{wp}\hat{O}(\vec{X}\rho\vec{X}^\dagger)\hat{O}^\dagger + p_{np}\mathcal{D}_q(\rho). \tag{C.7}$$

Comparing Eq. (C.3) and Eq. (C.7), we observe that the term $\hat{O}(\vec{X}\rho\vec{X}^\dagger)\hat{O}^\dagger$ is very similar to our wrong path term $\mathcal{E}_{wp}(\rho)$ (the error that corresponds to reading out an incorrect memory location). The last term term $\mathcal{D}_q(\rho)$ in Eq. (C.7) is not of the form of our no-path error term $\mathcal{E}_{np}(\rho)$, as the latter consists of depolarizing channels of different strengths depending on the position of the address qubit (i.e., the qubits are affected in decreasing order of significance, that is, the first qubit is affected the most, whilst the last one the least). However, $\mathcal{D}_q(\rho)$ is a decohering term, which seems to be a "weaker" form of noise than $\mathcal{E}_{np}(\rho)$. We showed above that even with this weaker decoherence term the quadratic speedup of any searching algorithm is lost. Therefore we have strong reasons to believe that adding a stronger decoherence term will not lower the quantum query complexity for the quantum searching problem. A rigorous proof of this conjecture remains an open problem.

Appendix D

Error correction schemes for qRAM

D.1 Correcting simple bit-flip errors

We show below that for a qRAM governed by a toy error model of the form

$$O(\rho) = (1-p)\hat{O}\rho\hat{O}^\dagger + p\hat{O}(\vec{X}\rho\vec{X}^\dagger)\hat{O}^\dagger, \tag{D.1}$$

the query error rate can be made arbitrarily small by using quantum error correction. Here $\hat{O}$ denotes the perfect oracle and $\vec{X}$ represents a multi-qubit bit-flip channel (a tensor product of individual bit-flip operators acting on an arbitrary subset of qubits). While such error models are not realistic for the architecture presented in this work, it may be that future designs allow for simpler error propagation. Such schemes could benefit from quantum error correction to sufficiently reduce their error rate to enable Grover search.

As Grover's algorithm requires $\mathcal{O}(\sqrt{N})$ steps, one desires a target logical error rate of $\delta = \mathcal{O}(1/\sqrt{N})$. Since the faulty oracle has an error model that consists of a bit flip channel followed by the perfect oracle call, one can use a quantum error correcting code and apply the oracle in parallel along the qubits composing the code. The parallelism of the oracle calls mimics majority counting and allows for error correction to be performed between logical oracle call steps. For simplicity, we provide an example that corrects against bit flip errors only using the repetition code, however such an analysis could be extended to correct for phase flips using code families such as the color codes [25], higher dimensional Shor codes [125], or triorthogonal codes [31].

For example, consider an oracle of the form $|a\rangle|b\rangle \to |a\rangle|b \oplus f(a)\rangle$, where $a,\ b \in \{0,1\}$. A logical oracle call that uses an n-qubit repetition code behaves as follows for states in

the computational basis:

$$|a\rangle|b\rangle \xrightarrow{V} |a\rangle^{\otimes n}|b\rangle^{\otimes n} \xrightarrow{\hat{O}^{\otimes n}} |a\rangle^{\otimes n}|b \oplus f(a)\rangle^{\otimes n}, \tag{D.2}$$

where V denotes the isometric encoding. Therefore, given a repetition code of length d, the code corrects for all errors up to $d/2-1$ physical bit flips by majority counting, using non-destructive Z-type stabilizer measurements. Therefore, the logical error rate becomes $p_L = p^{d/2}$. Choosing d large enough allows the logical error rate to satisfy $p_L = p^{d/2} < \delta$, where δ is the desired target fidelity. Therefore

$$d > \frac{2\log\delta}{\log p} = \frac{2\log\left(1/\sqrt{N}\right)}{\log p} = \frac{n}{\log\left(1/p\right)}. \tag{D.3}$$

Each of the n address qubits that serve as input to the oracle call must be encoded into a repetition code of length d. Hence, the total number of oracle calls for the complete Grover search algorithm is $\mathcal{O}(nd\sqrt{N}) = \mathcal{O}(n^2\sqrt{N}) = \mathcal{O}(\sqrt{N}(\log N)^2)$. As such, there is a logarithmic penalty for error correction, yet the scaling is not linear as in the error model of Regev and Schiff [122].

D.2 The failure of repetition codes for Regev and Schiff error model

The above error correction scheme is not applicable to the error model presented in [122], described by $\mathcal{R}_p(\rho) = (1-p)\hat{O}\rho\hat{O}^\dagger + p\rho$, since the failure of an oracle call can lead to an uncorrectable error, as demonstrated below. Consider the following example of the 3-qubit repetition code, where rather than all three oracles calls succeeding, the oracle call on the first set of qubits fails. The computational states evolve as:

$$|000\rangle|000\rangle \xrightarrow{\hat{O}_2\hat{O}_3} |000\rangle|0f(0)f(0)\rangle \tag{D.4}$$

$$|111\rangle|000\rangle \xrightarrow{\hat{O}_2\hat{O}_3} |111\rangle|0f(1)f(1)\rangle. \tag{D.5}$$

Consider the action of such a faulty oracle on the encoded state $(|000\rangle + |111\rangle)/\sqrt{2}$, for $f(0) = 0$ and $f(1) = 1$. The resulting mapping is

$$\begin{aligned} &\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle) \otimes |000\rangle \\ &\qquad \xrightarrow{\hat{O}_2\hat{O}_3} \frac{1}{\sqrt{2}}(|000\rangle|000\rangle + |111\rangle|011\rangle). \end{aligned} \tag{D.6}$$

The syndrome check operators for the repetition code are the parity check operators $\{Z_1Z_2, Z_2Z_3\}$. They are used to determine if an oracle call has failed by measuring the ancilla qubits. However, the measurement collapses the state to either $|000\rangle|000\rangle$ or $|111\rangle|011\rangle$. Upon applying the appropriate correction based on the measured syndromes, the resulting state becomes either $|000\rangle|000\rangle$ or $|111\rangle|111\rangle$. Therefore, the logical oracle call has failed, since the correct result must yield the superposition $(|000\rangle|000\rangle+|111\rangle|111\rangle)/\sqrt{2}$.

As expected, the error correction properties of the repetition code are not in violation of the results of Ref. [122], which state that a linear number of noisy black-box oracle calls are required, even with the addition of error correction.

D.3 The failure of repetition codes for our error model

Consider the oracle error model:

$$O(\rho) := p_{rp}\hat{O}\rho\hat{O}^\dagger + p_{wp}\mathcal{E}_{wp}(\rho) + p_{np}\mathcal{E}_{np}(\rho), \tag{D.7}$$

where $\hat{O}$ is the perfect oracle call while $\mathcal{E}_{wp}(\rho)$ and $\mathcal{E}_{np}(\rho)$ are the *wrong path* and *no-path* terms, respectively. We model the *wrong path* term as a convex combinations of bit-flip channels followed by perfect oracle calls. An example of one of those terms is the second term in Equation D.1. We model the *no-path* term as taking any input state and mapping it to a fixed state $|g\rangle$, which represents the loss of a qubit to be replaced by any fixed ancillary state. It should be noted that in the *no-path* case, the readout ancilla state does not change. Consider the action of the noisy channel on the five-qubit repetition code. Each instance of the channel has a certain probability of failure given by the associated weights. Focusing on one particular instance where the first address photon is lost and the second is affected by a bit flip, the resulting mapping on the computational basis states is given by:

$$|00000\rangle|00000\rangle \longrightarrow |g1000\rangle|0f(1)f(0)f(0)f(0)\rangle \tag{D.8}$$
$$|11111\rangle|00000\rangle \longrightarrow |g0111\rangle|0f(0)f(1)f(1)f(1)\rangle. \tag{D.9}$$

Again choosing $f(0) = 0$ and $f(1) = 1$, a superposition of input states in the computational basis evolves as

$$\frac{1}{2}\mathrm{P}\left[(|00000\rangle + |11111\rangle) \otimes |00000\rangle\right] \longrightarrow$$
$$\frac{1}{2}\left(\mathrm{P}\left[|g1000\rangle \otimes |01000\rangle\right] + \mathrm{P}\left[|g0111\rangle \otimes |00111\rangle\right]\right), \tag{D.10}$$

where P[•] denotes the projector onto its argument. The measurement of the stabilizers of the 5-qubit code on the ancillary states results in the collapse of the state into one of two terms depending on the syndrome measured. Note that the *no-path* term is the term that destroys coherence, similarly to the error term in the Regev and Schiff model [122].