Defects in Topologically Ordered Lattice Models

Benjamin James Brown

December 20, 2013

Supervised by Myungshik Kim and Vlatko Vedral

Submitted in part fulfilment of the requirements for the degree of Doctor of Philosophy in Department of Physics of Imperial College London and the Diploma of Imperial College London
Copyright Statement

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.
Statement on Collaborative Work

The work conducted in this Thesis is the result of many fruitful discussions and collaborations. We briefly describe the contributions of the present author to the work described in this Thesis.

The ideas used in Chapter 3 were conceived from discussions between the present author and Sean Barrett. The work was conducted in collaboration with Stephen Bartlett and Andrew Doherty. The calculations were conducted and the research was primarily led by the present author. The Chapter is based on work found in Reference [23] which has been accepted for publication in Physical Review Letters.

The work presented in Chapter 4 was completed in collaboration with Hussain Anwar, Earl Campbell and Dan Browne. All numerical data presented in this Chapter 4 was performed by the present author. The work of Chapter 4 is based on work present in Reference [4], which was co-led by Hussain Anwar and the present author.

The new ideas in Chapter 5 were formed following discussions between the present author and Jiannis Pachos. The research was completed in collaboration with Abbas Al-Shimary and Jiannis Pachos. The research was led primarily by the present author. The Chapter is based on work presented in Reference [22].

Work in this Thesis stems from a collaboration with Wonmin Son, Christina Kraus, Rosario Fazio and Vlatko Vedral published in [24] which is not presented here.
In memory of Adam and Sean, great friends and colleagues
Acknowledgements

The work produced and presented in this Thesis could not have been com-
pleted without the help of a great many people who I mention here. I begin
by thanking my family, who have supported me throughout my education. I
thank also Jarvis Cocker for his radio show, *Jarvis Cocker’s Sunday Service*,
and for putting boredom back into Sundays.

This Thesis could not have been completed without the encouragement
and guidance of my supervisors, Myungshik Kim and Vlatko Vedral. Most
importantly I thank them for the infinite freedom they have provided me
to work on the projects of my choosing. I thank also all those who have
worked with me and led me through the physics that is present in the
Chapters of this Thesis. In order of appearance they are Wonmin Son,
Christina Kraus, Rosario Fazio, Vlatko Vedral, Stephen Bartlett, Andrew
Doherty, Sean Barrett, Abbas Al-Shimary, Jiannis Pachos, Hussain Anwar,
Earl Campbell and Dan Browne. All of whom have been a pleasure to work
with. I thank also Sean Barrett, Simon Benjamin, Jiannis Pachos and Terry
Rudolph for moral support and career guidance they have given me during
my Ph.D.

My experience in London would not have been the same without the stim-
ulating environment provided by Terry Rudolph, Danny Segal and Myun-
shik Kim, and all the students of the Doctoral Training Centre. To mention
my favourites, they are Adam Billing, Joe Goodwin, Sania Jevtic, Aisha
Kaushik, Valentina Zhelyazkova, Danni Youziel, Matt Pusey, Naomi Nick-
erson, Howard Dale, Jake Iles-Smith, Mercedes Gimeno-Segovia and Chris
Self. Not to mention Kyle McEnery, Doug Plato, Mehdi Ahmadi and David
Herrera Martí with whom I have also shared office space. Graduate stud-
ies would not have been nearly as enjoyable/bearable without these people.
Adam and Sania are mentioned once again in this paragraph for many excel-
 lent tea breaks and evenings spent in London. I thank also Miranda Smith,
and my other mother, Lilian Wanjohi, for administering the whole Doctoral
Training Centre operation.

I must thank most of the quantum information community for their hospitality during my travels over the last three years. In particular, in geographical order I thank the groups at the University of Leeds, the University of Oxford, University College London, the California Institute of Technology, the University of Southern California, the Université de Sherbrooke, Perimeter Institute, the University of Waterloo, the National University of Singapore, MacQuarie University, the University of Sydney, the University of Queensland, NEST, Scuola Normale Superiore, Pisa, the Universidad Complutense Madrid, the Universitàt Basel, the Universiteit van Amsterdam, the Freie Universität Berlin, Leibniz Universität, Hannover, and RWTH Aachen. The research in this Thesis and my travels have benefitted greatly from the many researchers of these institutions, to name a few, Courtney Brell, Simon Burton, Guillaume Duclos-Cianci, Olivier Landon-Cardinal, Jonas Anderson, Jeongwan Haah, Isaac Kim, Giandomenico Palumbo and James Wootton. On the topic of travel I acknowledge also Courtney Brell, Matthew Turp, Lucy Houghton, Paul Stone, Kyle Hall, Kirtsy Garrod, Daniel Page, Sania Jevtic, Adam Billing, Joe Goodwin, Aisha Kaushik, Valentina Zhelyazkova, Ramil Nigmatullin and Mischa Woods for making excellent travelling companions during the last three years.

I would finally like to thank Sean Barrett, without whom this research would not have been possible. From Sean I have learned the importance of an inquisitive nature, and of speaking one’s mind. Many of the early ideas in this Thesis began from discussions with him, and I hope that all of the research in the Thesis echoes fairly both his ideas and the manner in which he would have had me carry it out.
Abstract

Developing quantum systems which are robust against noise are of prime importance to the realisation of quantum technologies. Without fault tolerance, we will never be able to preserve delicate quantum states for macroscopic time scales, a necessary requirement in the construction of scalable quantum computer. Topologically ordered models offer very beautiful mechanisms for preserving quantum states. In such models, quantum information is encoded globally over a degenerate topological Hilbert space which offers a natural robustness against local environmental noise. Remarkably, topological order is present in certain realistic condensed-matter systems. This provides a platform for accessing topological order in a laboratory. Moreover, considering condensed-matter systems enables us to combine topological features with other physical effects to enhance their behaviour.

In this Thesis we study the physics of topologically ordered lattice models with defects. We seek practical applications of lattice defects for the realisation of a fault-tolerant quantum computer. The first result we present shows that anyonic data of point-like lattice defects, called twists, can be found by measuring the entanglement of the ground state of the host system. The data we learn relates to the capacity of a twist defect to perform quantum computational tasks. The second result in this Thesis concerns the dynamics of the qudit toric code model with line-like defects coupled to a thermal bath. We show we can entropically inspire fragile glassy dynamics in the system. Such dynamics qualitatively improve the coherence times of quantum information encoded in the ground state of the lattice.

A novel way of achieving fault-tolerant quantum computation is by producing and manipulating twist defects of topologically ordered systems. In many ways, this paradigm of topological quantum computation is analogous to quantum computation using anyons. The first area of study in this Thesis extends the analogy between anyons and twists. Specifically, we show that the anyonic data of twists in Kitaev’s toric code model can be extracted
using topological entanglement entropy calculations in the same way as the same data can be extracted from anyons. We show this using a rigorously solvable lattice model as an example to produce exact analytic results. In particular, our results show we can extract the quantum dimension of a twist, and that we can study the quantum dimension of their fusion products. We compare the obtained results with the anyonic data of the Ising anyon model, further probing an analogy drawn previously in the literature.

The second result presented in this Thesis shows that the application of lattice defects can introduce novel dynamics to a two-dimensional topologically ordered quantum memory where excitations carry different masses. A two-dimensional topological model which supports an anyon model with a splitting structure allowed by its fusion channels should be able to entropically achieve high-energy excitation configurations before quantum information encoded in its ground state decoheres. We introduce a grid of lattice defects to a local two-dimensional Hamiltonian model, which, when coupled to a thermal bath, will dynamically steer the excitations into high energy configurations with high probability within a suitable temperature regime. We demonstrate the proliferating dynamics using numerical simulations in a low-temperature regime where we show polynomial improvements in coherence times by increasing the system size for small system sizes, as well as coherence times which scale weakly super exponentially with the inverse temperature of the bath. The dynamics we demonstrate provide the first example of a system which entropically steers excitations into high-energy configurations. The dynamics we demonstrate may lead to the development of experimentally tractable architectures for low dimensional quantum memories.

The study of thermal stability in this Thesis requires the development of methods of correcting topologically ordered lattices which have suffered errors. Namely, we require a decoding algorithm for the qudit generalisation of Kitaev’s toric code. Further work presented in this Thesis compares rigorously two different decoding algorithms which use renormalisation-group techniques to process classical syndrome information about noise suffered by the topological code. In particular, we study the improvement in their thresholds as the local dimension of its physical systems increase. Our numerical results enable us to analyse and identify the limitations of the different methods of decoding.
Contents

1. Introduction 15

2. Preliminaries 19
   2.1. Anyon Models 19
      2.1.1. Charge Labels 20
      2.1.2. Fusion Rules 21
      2.1.3. F-Moves 22
      2.1.4. R-Moves 23
      2.1.5. Constraints of Anyon Models 23
      2.1.6. Other Characteristics of Anyon Models 24
   2.2. Domain Walls 26
      2.2.1. A Mathematical Description of a Domain Wall 27
      2.2.2. A Spatial Realisation of a Domain Wall 27

3. Topological Entanglement Entropy with a Twist 29
   3.1. The Toric Code Model 30
      3.1.1. Excitations of the Toric Code Hamiltonian 32
      3.1.2. Anyons on the Toric Code Model 35
   3.2. The Toric Code Model with Twists 38
      3.2.1. Ising Anyons 38
      3.2.2. Twists 39
      3.2.3. Ising Anyon Fusion Rules from Twists 41
   3.3. Topological Entanglement Entropy 43
      3.3.1. The von Neumann Entropy 44
      3.3.2. The von Neumann Entropy of a Topologically Ordered Ground State 45
      3.3.3. Topological Entanglement Entropy 46
      3.3.4. Evaluating Quantum Dimensions of Anyons by Entropy Calculations 46
<table>
<thead>
<tr>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4. Topological Entanglement Entropy with a Twist</td>
</tr>
<tr>
<td>3.4.1. The Topological Entanglement Entropy of a Lattice with Twists</td>
</tr>
<tr>
<td>3.4.2. The Annular Topological Entanglement Entropy of a Twist</td>
</tr>
<tr>
<td>3.4.3. The Annular Topological Entanglement Entropy Pairs of Fused Twists</td>
</tr>
<tr>
<td>3.5. Conclusions</td>
</tr>
<tr>
<td>4. Qudit Toric Code Decoding Algorithms</td>
</tr>
<tr>
<td>4.1. The Qudit Toric Code</td>
</tr>
<tr>
<td>4.2. HDRG Decoder</td>
</tr>
<tr>
<td>4.2.1. Clustering a Syndrome</td>
</tr>
<tr>
<td>4.2.2. Optimising Decoding by Degeneracy</td>
</tr>
<tr>
<td>4.2.3. Cluster Levels</td>
</tr>
<tr>
<td>4.2.4. Neutralising a Cluster</td>
</tr>
<tr>
<td>4.3. SDRG Decoder</td>
</tr>
<tr>
<td>4.3.1. Exhaustive Decoding</td>
</tr>
<tr>
<td>4.3.2. Efficient Decoding which Approximates Exhaustive Decoding</td>
</tr>
<tr>
<td>4.3.3. Decoder Implementation</td>
</tr>
<tr>
<td>4.3.4. Renormalisation Cells</td>
</tr>
<tr>
<td>4.3.5. Belief Propagation</td>
</tr>
<tr>
<td>4.4. The Noise Model and Threshold Estimation</td>
</tr>
<tr>
<td>4.5. Results and Analysis</td>
</tr>
<tr>
<td>4.6. Conclusions</td>
</tr>
<tr>
<td>5. Two-Dimensional Quantum Memories with Line Defects</td>
</tr>
<tr>
<td>5.1. The Qudit Toric Code</td>
</tr>
<tr>
<td>5.1.1. The Hamiltonian</td>
</tr>
<tr>
<td>5.1.2. Excitations of the Qudit Toric Code Model</td>
</tr>
<tr>
<td>5.1.3. Ground State Degeneracy of the Qudit Toric Code</td>
</tr>
<tr>
<td>5.2. Simulating the Toric Code Hamiltonian in a Thermal Bath</td>
</tr>
<tr>
<td>5.3. The Behaviour of the Qudit Toric Code Coupled to a Thermal Bath</td>
</tr>
<tr>
<td>5.4. Charge Modifying Defect Lines</td>
</tr>
<tr>
<td>Section</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>5.5. Modified Excitation Dynamics in the Presence of Defect Lines</td>
</tr>
<tr>
<td>5.6. A Single Excitation on a Lattice with Defect Lines</td>
</tr>
<tr>
<td>5.7. Coherence Time Improvements on a Lattice with Defect Lines</td>
</tr>
<tr>
<td>5.7.1. Arrhenius Scaling</td>
</tr>
<tr>
<td>5.7.2. Weak Super-Exponential Inverse Temperature Scaling</td>
</tr>
<tr>
<td>5.8. Conclusions</td>
</tr>
<tr>
<td>6. Summary and Outlook</td>
</tr>
<tr>
<td>A. Stabilizers</td>
</tr>
<tr>
<td>B. HDRG Decoder Threshold Curves</td>
</tr>
<tr>
<td>C. SDRG Decoder Threshold Curves</td>
</tr>
</tbody>
</table>
List of Figures

2.1. Fusing two anyons. ................................. 20
2.2. Fusing three anyons. ............................... 21
2.3. An F-move. ......................................... 22
2.4. An R-move ......................................... 23
2.5. The hexagon identity ................................. 25
2.6. A domain wall. ..................................... 28

3.1. Interactions of the toric code model .................. 31
3.2. Excitations on the toric code. ......................... 32
3.3. A dion on the toric code. ............................. 33
3.4. Homotopically equivalent string operators. ............. 34
3.5. Braiding anyons on the toric code. ................. 36
3.6. A twist interaction on the toric code ................ 40
3.7. Logical operators for four twists. ................. 42
3.8. A twist absorbing a dion. ........................... 44
3.9. A region on a lattice ................................ 45
3.10. Regions used to to evaluate topological entanglement entropy 47
3.11. The von Neumann entropy of a region with a twist .......... 49
3.12. Topological entanglement entropy regions with twists. .... 51
3.13. Annular topological entanglement entropy with a twist ..... 52
3.14. Annular topological entanglement entropy with two twists . 53
3.15. The von Neumann entropy of an annular region of a stabilizer state. ......................... 54

4.1. Non-trivial cycles of a torus .......................... 58
4.2. Qudit toric code star and plaquette operators ............ 61
4.3. (a) Logical operators $X_2$ and $Z_1$ are shown in red and blue, respectively. (b) Logical operators $X_1$ and $Z_2$ are shown in red and blue, respectively. ......................... 61
<table>
<thead>
<tr>
<th>Number</th>
<th>Figure Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4</td>
<td>Errors and a syndrome on the toric code.</td>
<td>62</td>
</tr>
<tr>
<td>4.5</td>
<td>Correction operators to a trivial syndrome.</td>
<td>65</td>
</tr>
<tr>
<td>4.6</td>
<td>An example syndrome.</td>
<td>67</td>
</tr>
<tr>
<td>4.7</td>
<td>Clustering levels of the broom decoder.</td>
<td>68</td>
</tr>
<tr>
<td>4.8</td>
<td>A syndrome and matching with degeneracy.</td>
<td>69</td>
</tr>
<tr>
<td>4.9</td>
<td>Clustering distance at the $\lambda$-clustering level.</td>
<td>71</td>
</tr>
<tr>
<td>4.10</td>
<td>A correction operator within a cluster.</td>
<td>72</td>
</tr>
<tr>
<td>4.11</td>
<td>SDRG cells on the qudit toric code lattice.</td>
<td>75</td>
</tr>
<tr>
<td>4.12</td>
<td>A single SDRG cell.</td>
<td>76</td>
</tr>
<tr>
<td>4.13</td>
<td>HDRG threshold calculation, $d = 5$.</td>
<td>81</td>
</tr>
<tr>
<td>4.14</td>
<td>SDRG threshold calculation, $d = 3$.</td>
<td>81</td>
</tr>
<tr>
<td>4.15</td>
<td>Decoder thresholds for varying local dimensions.</td>
<td>82</td>
</tr>
<tr>
<td>5.1</td>
<td>Qudit toric code primal and dual operators.</td>
<td>91</td>
</tr>
<tr>
<td>5.2</td>
<td>Strings on the qudit toric code.</td>
<td>93</td>
</tr>
<tr>
<td>5.3</td>
<td>Logical operators on the qudit toric code.</td>
<td>94</td>
</tr>
<tr>
<td>5.4</td>
<td>Excitation types populations as a function of inverse temperature.</td>
<td>98</td>
</tr>
<tr>
<td>5.5</td>
<td>High-mass excitation and low-mass excitation ratio plot against inverse temperature.</td>
<td>99</td>
</tr>
<tr>
<td>5.6</td>
<td>Memory time vs. inverse temperature plot for the qudit toric code without defect lines.</td>
<td>100</td>
</tr>
<tr>
<td>5.7</td>
<td>String operators on a charge 2-modifying defect line.</td>
<td>102</td>
</tr>
<tr>
<td>5.8</td>
<td>Stabilizers on charge $M$-modifying defect lines.</td>
<td>103</td>
</tr>
<tr>
<td>5.9</td>
<td>Excitation dynamics on a defected lattice.</td>
<td>104</td>
</tr>
<tr>
<td>5.10</td>
<td>An excitation on a lattice with defect lines.</td>
<td>107</td>
</tr>
<tr>
<td>5.11</td>
<td>Single excitation dynamics for the qudit toric code with twists.</td>
<td>108</td>
</tr>
<tr>
<td>5.12</td>
<td>A single lattice unit used in coherence time scalings.</td>
<td>112</td>
</tr>
<tr>
<td>5.13</td>
<td>Memory time vs. inverse temperature plot for the qudit toric code with defect lines.</td>
<td>113</td>
</tr>
<tr>
<td>5.14</td>
<td>Memory time vs. system size plot for the qudit toric code with domain walls</td>
<td>115</td>
</tr>
<tr>
<td>B.1</td>
<td>Qudit toric code HDRG decoder threshold, $d = 2$.</td>
<td>123</td>
</tr>
<tr>
<td>B.2</td>
<td>Qudit toric code HDRG decoder threshold, $d = 3$.</td>
<td>123</td>
</tr>
<tr>
<td>B.3</td>
<td>Qudit toric code HDRG decoder threshold, $d = 5$.</td>
<td>124</td>
</tr>
</tbody>
</table>
B.4. Qudit toric code HDRG decoder threshold, $d = 7$  . . . . . . 124
B.5. Qudit toric code HDRG decoder threshold, $d = 11$  . . . . . . 125
B.6. Qudit toric code HDRG decoder threshold, $d = 13$  . . . . . . 125
B.7. Qudit toric code HDRG decoder threshold, $d = 17$  . . . . . . 126
B.8. Qudit toric code HDRG decoder threshold, $d = 19$  . . . . . . 126

C.1. Qudit toric code SDRG decoder threshold, $d = 2$  . . . . . . 128
C.2. Qudit toric code SDRG decoder threshold, $d = 3$  . . . . . . 128
C.3. Qudit toric code SDRG decoder threshold, $d = 5$  . . . . . . 129
C.4. Qudit toric code SDRG decoder threshold, $d = 7$  . . . . . . 129
C.5. Qudit toric code SDRG decoder threshold, $d = 11$  . . . . . . 130
C.6. Qudit toric code SDRG decoder threshold, $d = 13$  . . . . . . 130
C.7. Qudit toric code SDRG decoder threshold, $d = 17$  . . . . . . 131
C.8. Qudit toric code SDRG decoder threshold, $d = 19$  . . . . . . 131
1. Introduction

In recent years condensed-matter physics has undergone a major revolution upon the discovery of topologically ordered phases of matter [52, 70, 72, 73, 74]. Topologically ordered phases of matter go beyond Landau’s paradigm of phase transitions as, remarkably, different topologically ordered phases cannot be distinguished by their symmetries. In fact, there does not yet exist a universal characterisation for such phases. Topologically ordered phases are identified by exotic features such as a topological ground-state degeneracy that depends on the topology of the manifolds that embeds the system [40], anyonic quasi-particle statistics [5, 57], and stability against local perturbations [18, 76].

The study of topologically ordered phases has been greatly accelerated due to exciting applications in quantum information science [58, 62]. Indeed it has been shown anyonic excitations can provide efficient methods of evaluating the Jones polynomial [77] and further that universal quantum computation can be performed using the anyonic excitations of particular topologically ordered phases of matter [38].

Topologically ordered phases of matter also stand out as natural architectures to construct a quantum computer as their defining features are inherently tolerant to faults. One of the greatest challenges in developing quantum technologies is shielding delicate quantum states from harmful environmental noise which decoheres encoded information. Topological phases have global degrees of freedom that are naturally robust against local imperfections in the system. In particular, it is proven that the ground state of a topologically ordered system is stable [18] against local static perturbations in the zero temperature limit. This means that the gap of a topologically ordered system will remain finite in the presence of local perturbations as the system becomes large, thus maintaining an energy gap between states encoded in the ground space of the system which protects encoded information. Moreover, the splitting of the ground state degeneracy in a perturbing
field is exponentially suppressed with the size of the system. These features mean topologically ordered systems synthesised in the laboratory are in principle robust against typical systematic faults such as stray magnetic fields.

While topologically ordered systems have characteristic resilience against faults at zero temperature, it is not clear what types of topologically ordered systems will protect encoded quantum information at finite temperatures [21, 44, 45, 51, 61]. Pioneering results in three-dimensional systems [16, 26, 56] have begun to emerge in the literature which suggest that thermal stability in topological systems may be possible although it still remains to be demonstrated. In spite of this, the impressive natural fault-tolerance inherent in topologically ordered models, coupled with their computational capabilities, offer an exciting platform to develop candidate systems potentially capable of fault-tolerant quantum computation in the presence of all types of environmental noise. The desire to study the physics of topologically ordered models have lead to the design of families of rigorously solvable lattice models that provide a beautiful test bed for the analysis of topologically ordered phases of matter [15, 37, 50, 47, 54].

An exciting recent development in the field of topologically ordered lattice models is the study of topological defects, which is the topic of this Thesis. It has been shown that lattice models [11, 12, 13, 48], and more generally in other topologically ordered phases [7, 8, 10] that topological defects can be introduced into the model by introducing inhomogeneities to the system. These defects interact with anyonic excitations providing rich and exotic new physics. Astonishingly, it is shown that such topological defects can be moved and manipulated such that they can perform computational tasks not otherwise expected of their host model [50, 47].

In Chapter 3 we explore new techniques for probing the computational properties of topological defects. Specifically we show that data about the computational capacity of lattice defects can be learned by studying the entanglement of the ground state of the Hamiltonian of the model. It is already known that we can learn information which characterises anyonic quasi-particle excitations of a topologically ordered phase can be extracted using entanglement calculations [29, 49, 55]. We demonstrate that these known results can be extended to study twist defects using a simple rigorously solvable lattice model which admits exact results. The methods we
develop offer new tools in the study of lattice defects. These tools may be used to identify particular topological phases which can support universal quantum computation by the addition of lattice defects. Moreover, in the specific case we consider, our results interrogate further an analogy between lattice defects and anyons. In particular, it has been demonstrated in the literature that the defects in the model we consider behave like a particular type of anyon, namely the Ising anyon, of the Ising anyon model [12]. All of the diagnostics using entanglement measures we use show that twists admit the same results we expect of Ising anyons, extending further this analogy.

In Chapter 4 we take a small detour from lattice defects to consider methods of decoding topological error correcting codes, namely, Kitaev’s qudit toric code model [50]. The methods we develop in this Chapter are required for the rigorous numerical analysis we give in Chapter 5 where we show that we can improve the thermal stability of a topologically ordered model by introducing defect lines. A decoding algorithm considers classical syndrome information measured from a quantum error correcting code, and attempts to find the optimal operation to reverse errors suffered by the system [16, 28, 30, 43, 82]. Furthermore, the analysis we present on different methods of decoding is interesting in its own right, as it provides a good demonstration of how topologically ordered systems are robust against one particular local source of noise, which is a key theme in this Thesis. We study two methods of decoding the qudit toric code model. We numerically study each decoder by calculating their threshold. The threshold is the maximum rate of noise the error correcting code can suffer such that we are still able to recover the encoded information with arbitrarily high probability by increasing the size of the code. In particular, we study how the threshold of the decoders varies as we change the dimension of the local Hilbert space of the model. Certain quantum architectures for quantum error correcting codes may show better performance using physical systems with more than two levels. The results we present in this Chapter give a lower bound on maximum coherence times and noise rates such architectures will require.

In Chapter 5 we study the thermodynamics of a two-dimensional quantum memory where we introduce a grid of defect lines to the lattice. A quantum memory is designed such that the Hamiltonian interactions dynamically protect quantum information encoded in its ground space from corruption from environmental noise [3, 16, 26, 56]. We show that the introduction
of defects to a lattice model can entropically encourage the dynamics of the typical excitations of the model into high energy configurations within a particular parameter regime. Such entropic effects may be useful in the development of low-dimensional quantum memories. We witness these effects using extensive numerics. In particular we study the dynamics of a single excitation, as well as the coherence-time scaling of states encoded in the memory with inverse temperature and system size. We argue that weak super-exponential inverse-temperature scaling is an artefact of the high energy excitations we describe. The study of entropic suppression of error propagation may offer new paths towards realising an efficient quantum memory with experimentally-amenable low dimensionality. We finally conclude by discussing the limitations of the entropic effects we introduce in the zero-temperature limit, and we suggest new research directions to overcome these problems.

In the final Chapter we summarise the work completed in this Thesis, and review the prospects of developing useful technologies for quantum computation using defects in topologically ordered lattice models.
2. Preliminaries

Topological phases of matter can be characterised by the anyonic quasi-particle excitations they can support. In this Chapter we discuss the abstract mathematical framework that describes an anyon model of a topological phase. We go on consider domain walls between which separate two distinct topological phases [11, 48]. A domain wall is an essential concept for describing the defect lines we consider in this Thesis. The defect lines we introduce separate two topological phases on a lattice. A natural description of a defect line then is provided using the language of domain walls. We conclude this Section by considering how defect lines provide physical realisations of domain walls on lattice models.

2.1. Anyon Models

A topological phase can be characterised by the anyons it can support. It is the existence of anyons that give topological phases their unusual properties, such as a topological degeneracy which depends on the genus of the manifold on which it is embedded, as well as its stability against zero temperature perturbations [50]. We now give a mathematical framework which describes the features of anyon models discussed in this Thesis. This Section will provide sufficient information about anyon models to understand this Thesis, but further information about the abstract theory of anyons can be found in appendix E of Reference [47] or Reference [62].

Anyons are particles which exist in two dimensions. An exotic consequence of this is that anyons have non-trivial exchange statistics. This is a special feature of objects which exist in two dimensions, the reason for this is that in two dimensions the world lines of two particles which braid is homotopically non trivial. Specifically, when two anyons are exchanged, they can acquire a phase of $e^{i\theta}$. This is a generalisation of bosonic and fermionic particles which acquire phases of +1 and -1 under exchange, respectively.
2.1 Anyon Models

Figure 2.1.: The world lines of two anyons, $a$ and $b$, which fuse to give anyonic charge $c$ where time runs downwards. The diagram describes the state of the system of anyons $a$ and $b$. However, we omit the ket vector, for brevity of presentation.

Another unique feature of anyons is that they have a degenerate Hilbert space associated to them, which we describe as a topological Hilbert space, or sometimes a fusion space. In general, the braiding of anyons will perform unitary transformations over the topological Hilbert space.

An anyon model requires charge labels, which describe the anyons available in an anyon model. The evolution of a system of anyons is then described by fusion rules, F-moves and R-moves. These four objects uniquely describe an anyon model, and therefore a phase of matter. It is also convenient to define the total quantum dimension of an anyon model, the quantum dimension of an individual anyon, and the S-matrix of an anyon model. These are discussed in later Sections.

2.1.1. Charge Labels

To begin describing an anyon model, we first require labels for all the different anyonic charges. Unique anyonic excitations are given labels such as $a$, $b$, $c$, and so on. Further, all anyon models must have a vacuum charge, which describes ‘no charges’, and is labeled by 1.

A remarkable feature of anyons is that systems of anyons have global degrees of freedom associated to them, encoded in a topological Hilbert space, $\mathcal{H}_{\text{topo}}$. The state of these degrees of freedom can only be accessed globally by fusion measurements. Anyonic charges are the observables of fusion measurements. It is vital to stress that the state of these global degrees of freedom cannot be accessed by local charge measurements. Fusion occurs when two charges are moved close together, which results in the generation of a new charge, which we call the fusion product. It is this
2.1 Anyon Models

Figure 2.2.: The state of three anyons which fuse, represented pictorially by their world lines, where time runs downwards. Anyon $a$ and $b$ fuse to give anyon $i$, which subsequently fuses with $c$ to give anyon $d$. This is sometimes known as a fusion tree.

feature which gives $\mathcal{H}_{\text{topo}}$ its natural fault tolerance, as the information about the fusion product of a pair of anyons cannot be accessed if the two anyons are spatially separated. We discuss fusion with more rigour in the following Subsection.

2.1.2. Fusion Rules

The state of the system of anyons is measured by performing fusion operations. Fusion occurs when two anyons are moved close to one another, where they fuse to give a different observable anyonic charge. Anyonic charges are conserved quantities, which restricts the allowed fusion channels. Fusion channels are given by the tensor $N_{ac}^{bc}$, where

$$a \times b = \sum_c N_{ac}^{bc} c,$$

(2.1)

where $\times$ represents fusion. $N_{ac}^{bc} = 0$ is a forbidden fusion channel and $N_{ac}^{bc} \geq 1$ are allowed fusion channels. The $N_{ac}^{bc}$ tensor gives the fusion rules of an anyon model.

We can represent the state of a system of anyons diagrammatically as in Figure 2.1. The Figure shows a system of two anyons $a$ and $b$ which fuse to give anyon $c$. The world lines of anyons are moved close together, and subsequently fuse.

In general there can be multiple allowed fusion outcomes between two anyons. For instance we can have a fusion rule $a \times b = c + d$. This fusion rule means when $a$ and $b$ fuse, there can be two distinct observable charges,
2.1 Anyon Models

We describe a system of anyons following a different sequence of fusion using different basis vectors. We change between bases using the F-moves of the anyon model.

c or d. Furthermore, the state where a fuses with b to give c is orthogonal to the state where a fuses with b to give d. This enables us to encode quantum information in the fusion space of a system of anyons. We can write states in $\mathcal{H}_{\text{Topo}}$ with a labelling $|(a, b)_c\rangle$, where fusing the left anyon a with right anyon b gives fusion outcome c.

We may require a description for a state of a system of many anyons, for instance, if anyons a and b fuse to give fusion outcome i, which subsequently fuses with anyon c to give d, we write the state $|((a, b)_i, c)_d\rangle$, shown pictorially in Figure 2.2.

2.1.3. F-Moves

The order in which a system of anyons is fused together is important. Given a system of three or more anyons, we are able to choose the order in which we fuse anyons. Consider a system of three anyons, a, b and c, prepared in the state $|((a, b)_i, c)_d\rangle \in \mathcal{H}_{\text{Topo}}$, as shown in Figure 2.2, where a and b fuse together first to give i, and c fuses with i to give d. The same system of anyons, which is in the same topological Hilbert space can be fused in a different order, denoted by the state $|((a, (b, c))_j)_d\rangle \in \mathcal{H}_{\text{Topo}}$ which describes a different basis which spans the topological Hilbert space. We require a mapping to change between bases. The mapping is encoded in the tensor $(F_{d}^{abc})_{ij}$, otherwise known as the F-moves of the anyon model, where

$$|(a, (b, c))_j\rangle = \sum_i (F_{d}^{abc})_{ij} |((a, b)_i, c)_d\rangle.$$  \hspace{1cm} (2.2)
2.1 Anyon Models

\[ R_{ab} = R^{ba}_c \]

Figure 2.4.: The phase acquired by exchanging two anyons is captured by the R-moves of an anyon model.

An F-move is shown pictorially in Figure 2.3. The F-moves of an anyon model cannot be chosen arbitrarily. They are constrained by the pentagon and hexagon identities. These identities are described later in Subsection 2.1.5.

2.1.4. R-Moves

To complete the description of an anyon model, we require information describing the change in the system when two particles are exchanged before they are fused. This is described by an \textit{R-move}, \( R_{ab} \). We show an R-move pictorially in Figure 2.4. Exchanging two particles will change the Hilbert space of the system from the fusion space where anyon \( a \) fuses from the left and anyon \( b \) fuses from the right to the fusion space where \( b \) fuses from the left and \( a \) fuses from the right. We consider the case where this the fusion space is one dimensional, and the fusion outcome is anyon \( c \) with certainty. Two one-dimensional Hilbert spaces can differ by a phase, which we write \( R^{ba}_c \). This phase corresponds to the phase acquired when anyon \( a \) and anyon \( b \) are exchanged, as shown in Figure 2.4. In the situation where the fusion space is multi-dimensional, \( R_{ab} \) will act as a unitary operation on the fusion space of the anyons.

2.1.5. Constraints of Anyon Models

We have described all the data necessary to describe an anyon model. However, an anyon model must follow some physical constraints, namely the pentagon and hexagon identities.
2.1 Anyon Models

The Pentagon Identity

The pentagon identity keeps F-moves consistent. We consider the state \(|(a, b), c), d)\rangle_c\), which we wish to write in the basis \(|(a, b, (c, d))\rangle_{cl}\). We change between bases using F-moves. There are two different sequences of F-moves that can achieve such a basis change. The two different sequences must commute with one another. This gives the pentagon identity. The first sequence is

\[
|(a, b), c), d)\rangle_c = \sum_k \sum_l (F^e_{idc})^I_k (F^e_{abd})^I_l |(a, (b, (c, d))k)\rangle_c, \tag{2.3}
\]

and the other is

\[
|(a, b), c), d)\rangle_c = \sum_k \sum l \sum n (F^i_{abc})^n (F^i_{bcd})^n |(a, (b, (c, d))k)\rangle_c. \tag{2.4}
\]

Both of these sequences must commute, so it follows that

\[
\sum_n (F^e_{bcd})^n_k (F^e_{and})^l_i (F^d_{bcd})^n_l = (F^e_{abd})^l_i (F^e_{idc})^I_k. \tag{2.5}
\]

which is the pentagon identity.

The second identity, the hexagon identity, keeps consistency between F-moves and R-moves. It is

\[
\sum_k (F^d_{bac})^I_j R^d_{ak} (F^d_{abc})^i_k = R^d_{ab} (F^d_{bac})^I_j R^d_{ab}, \tag{2.6}
\]

which we show diagrammatically in Figure 2.5. The state transforms equivalent by two different sequences of F- and R-moves, which is captured by the hexagon identity.

An anyon model is physical provided both Equations (2.5) and (2.6) are satisfied for all anyons in the model.

2.1.6. Other Characteristics of Anyon Models

We conclude this Section by mentioning other useful data we can use to characterise anyons and anyon models. Namely the quantum dimension of an anyon, and the S-matrix and the total quantum dimension of an anyon model.
Figure 2.5.: Two commuting transformations composed of F- and R-moves make the hexagon identity.
Quantum Dimensions

We have seen that anyons in general have some topological Hilbert space associated to them. It is convenient to have anyonic data that characterises how the dimension of the fusion space grows with the number of anyons in the system. This quantity is known as the quantum dimension $d_a$. The quantum dimension relates to the dimension of the Hilbert space by the equation

$$\text{dim}(\mathcal{H}_{\text{topo}}) \approx \frac{d_a^n}{D^2},$$

in the limit of large $n$, where $n$ is the number of anyons and $D$ is the total quantum dimension of the anyon model. The total quantum dimension can be used to characterise an anyon model and is defined as

$$D = \sqrt{\sum_a d_a^2},$$

where we sum over all the anyons of the anyon model.

Anyon models where $d_a = 1$ for all $a$ are known as Abelian anyon models. Otherwise, they are non-Abelian anyon models.

The S-Matrix

We finally review the S-matrix. The S-matrix describes the phase acquired when an anyon undergoes a monodromy with another anyon. This is where one anyon is braided around another anyon, and then returned to its original position. The $ab$-th element of the S-matrix, $S_{ab}$, gives the phase acquired when anyon $a$ is braided around anyon $b$, divided by the total quantum dimension of the anyon model.

2.2. Domain Walls

In this Section we give a description of a domain wall \cite{11, 48}, a key concept in this Thesis which we use to describe the defect lines we introduce to a lattice. Precisely, we describe transparent domain walls, a concept which we build towards in this Section. We finally explain how defect lines relate to domain walls. We note that all the defect lines used in this Thesis act as transparent domain walls.
2.2 Domain Walls

2.2.1. A Mathematical Description of a Domain Wall

A domain wall separates two phases. We describe a domain wall with a mapping $G$, which maps phase $\mathcal{C}$ onto phase $\mathcal{D}$, such that

$$G : \mathcal{C} \to \mathcal{D}.$$ 

We note that at this point that the map $G$ may not be invertible. Naturally, we have topological phases in mind. Phases $\mathcal{C}$ and $\mathcal{D}$ are both anyon models with a set of charge labels associated to them. We note that we can consider non-topological models by considering the topological model that contains only the vacuum charge, if we so wish. The mapping $G$ maps topological charges, such that $G(a)$ is an anyonic charge of $\mathcal{D}$. The mapping $G$ is well defined for all $a \in \mathcal{C}$.

The domain walls of interest in this Thesis are automorphisms $F$ over $\mathcal{C}$, such that

$$F : \mathcal{C} \to \mathcal{C},$$

where $F$ maps charges of $\mathcal{C}$ onto charges of the same phase.

As previously mentioned we are interested in transparent domain walls. Such domain walls are invertible automorphisms. Transparency is ensured if $F$ satisfies the additional condition that

$$S_{ab} = S_{F(a)F(b)},$$ \hspace{1cm} (2.9)

for all $a$ and $b$ where $S_{ab}$ are elements of the S-matrix of $\mathcal{C}$, as described in the previous Section. This condition says that braiding is preserved when a phase transits a domain wall.

2.2.2. A Spatial Realisation of a Domain Wall

The subject of this Thesis is defect lines, which are well described within the framework of domain walls. We consider now a lattice which we divide into two regions. One region of the lattice is in phase $\mathcal{C}$ and the other region is in phase $\mathcal{D}$. The division is arbitrary provided both regions have non-zero volume. For instance, we may consider the lattice like a fried egg\textsuperscript{1}, where the yolk is in phase $\mathcal{C}$ and the white of the egg is in phase $\mathcal{D}$. The boundary

\textsuperscript{1}I’m not yolking.
2.2 Domain Walls

(a) (b)

Figure 2.6: (a) The anyons of an anyon model, \( C \), with charge labels \( a \), \( b \) and \( c \) shown in red blue and green, and the vacuum particle is shown in white. (b) The domain wall mapping \( F \) is represented by a red line. Moving anyons across this line maps anyons of anyon model \( C \) onto different anyons of \( C \).

between these two regions is a physical line described by a domain wall. On a lattice we can physically realise such a domain wall using a defect line, where excitations of \( C \) are mapped onto excitations of \( D \) by moving them across the defect line.

In the example of an egg, the defect line is continuous. The discussion we present in Chapter 3 requires terminating defect lines. Consider some charges in \( C \), as shown in Figure 2.6(a). Terminating domain walls are physically realised by defect lines which separate two copies of the same phase, as shown in Figure 2.6(b).

Finally, we note that all the defect lines we consider in this Thesis are physical realisations of transparent domain walls.
3. Topological Entanglement

Entropy with a Twist

Topologically ordered lattices give rise to anyonic excitations. These excitations can be manipulated and braided to perform computational tasks. Typically, these lattices are regular, and have some type of symmetry associated to them. However, the physics of the model can be enriched by adding lattice defects which break symmetries of the model. Twists are the termination points of line defects on a topologically ordered lattice model. Twists are the point of study of this Chapter.

In a similar way to which anyons can be used to perform topological quantum computation, twists can be created on a topologically ordered model, braided and fused to perform quantum computations as well. It has been shown that twists introduced to the toric code model have common attributes with non-Abelian anyons, in particular the Ising anyon model [12]. This is remarkable as the excitations of the toric code are Abelian anyons. Further, it has been demonstrated that twists added to different topologically ordered lattice models can give rise to other characteristics. For instance, adding twists to the color code model [15], another Abelian anyon model, enables the lattice to reproduce the Clifford gate set by the braiding of twists [13]. The Clifford gate set is of particular interest from the point of view of quantum computation, as the addition of a single noisy operation outside of the Pauli group give a universal gate set [20, 46]. It has also been shown that introducing twists to a system which gives rise to a bilayer Ising anyon model gives rise to universal quantum computation, without the need for noisy operations [8, 10]. This is achieved because braiding twists simulates modular transformations necessary to make the Ising anyon model universal [36].

Twists have many features in common with non-Abelian anyons. Like anyons, twists are conserved topological charges such that it is only possible
to create a twist if its corresponding anti twist is also created. Twists also have a topologically protected degenerate fusion space associated to them which gives them their non-Abelian characteristic. However, the analogy between twists and non-Abelian anyons is not exact [12]. Anyons arise as low energy excitations of a topologically ordered Hamiltonian, whereas twists are prepared by dislocating or deforming Hamiltonian interactions along a line. A consequence of this is that braiding is not defined in the same as for twists as it is for anyons.

It is of fundamental and practical interest then to know to what extent we are able to regard twists as anyon-like quasi-particle excitations. This will enable us to better understand their computational capabilities. In this Chapter we address this question by further probing the analogy between twists in the toric code model and Ising anyons, drawn by Bombin in Reference [12]. In this work, Bombin maps twists on the toric code, as well as a subset of the anyonic excitations of the model onto anyons from the Ising anyon model, and shows analogous behaviour. We use the topological entanglement entropy [49, 55, 29] to interrogate this analogy. Remarkably, we find that all of the diagnostics used show twists to give the outcomes expected from Ising anyons. Specifically, we recover the the total quantum dimension of the Ising anyon model, we find the quantum dimensions for the corresponding twist charges, as well as the correct quantum dimensions of fusion products of pairs of twists.

We begin this Chapter by reviewing the toric code model. We go on to review Ising anyons and the correspondence between twists and Ising anyons. In particular we look at how we recover the fusion rules of the Ising anyon model using twists. We go on to review topological entanglement entropy. Finally we use all of these tools to calculate the topological entanglement entropy of various configurations of twists on the toric code lattice, and show that twists reproduce the expected results of Ising anyons.

3.1. The Toric Code Model

We now review Kitaev's quantum double model [50]. This model is a beautiful realisation of a topologically ordered lattice model whose low energy excitations are anyons. In general one can write down a quantum double model for anyonic models with any group structure [50], some of which are
3.1 The Toric Code Model

Figure 3.1: A square lattice with spins lying on the vertices. The toric code model interacts via primal and dual four-body interactions shown respectively around green and blue faces. While the primal and dual interactions are the same, it will become important to draw a distinction between the two interaction types.

universal for quantum computation [50], and these models have been generalised to Levin-Wen models which can have any quasi-particle excitations with a modular tensor category structure [54]. In this Thesis however, we study only the Abelian quantum-double models. The first is the well studied toric code model, a spin-half model on a square lattice with four-body Hamiltonian interaction terms.

We consider the quantum double of the $\mathbb{Z}_2$ group, more commonly known as the toric code model. It is convenient to use the formulation of the toric code by Wen [75]. The toric code model is defined on a square lattice of $L \times L$ vertices with even $L$ and periodic boundary conditions where physical qubits lying on the vertices of the lattice, as is shown in Figure 3.1. The spins then interact via a four-body local Hamiltonian

$$H = - \sum_p W_p,$$  \hfill (3.1)

where $W_p = X \otimes Z \otimes Z \otimes X$ are the Hamiltonian interaction terms associated to the blue and green faces of plaquettes, $p$, of the square lattice, shown in
3.1 The Toric Code Model

Figure 3.2: Two string operators which create a pair of anyons. The blue string operator will anti-commute with two dual $W_p$ Hamiltonian interactions, thus creating two $m$ particles. The green string operator will anti-commute with two primal $W_p$ operators on green faces, thus creating two $e$ particles on the lattice.

Figure 3.1 where $X$ and $Z$ are the standard Pauli matrices, where

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (3.2)$$

There are two types of interaction terms, primal interactions, that act around a green face of the lattice, and dual interactions, that act around blue faces of the lattice. It is important to draw this distinction when we begin to consider the quasi-particle excitations on the lattice later in this Section. The interaction terms of Hamiltonian (3.1) commute which means they are simultaneously diagonalised. Furthermore, the ground states of the Hamiltonian, $|\psi_j\rangle$, are stabilizer states of the $W_p$ operators [59], such that $W_p |\psi_j\rangle = |\psi_j\rangle$, which we review in Appendix A.

3.1.1. Excitations of the Toric Code Hamiltonian

The excitations of the toric code Hamiltonian are anyonic. There are four particle types, the vacuum, labelled 1, the electric charge, $e$, the magnetic vortex, $m$, and the dion particle, $\epsilon$. The dion is the fusion product of an $e$ and an $m$ anyon. Electric charges and magnetic vortices are excitations which lie on the primal and dual faces of the lattice respectively. The anyons
3.1 The Toric Code Model

are created by string operators of Pauli matrices, shown in Figure 3.2. Excitations emerge at the end points of the string operators. As strings have two ends, excitations will always be created in pairs. Electric charges are created on primal faces of the lattice and magnetic vortices are created on dual faces of the lattice. String operators will only follow one colour of face, and only move in diagonal directions, like the bishop pieces on a chessboard\(^1\). This will determine what type of particles they create. Dions are the fusion product of an \(e\) anyon and an \(m\) anyon. We show a dion in Figure 3.3.

An important feature of topological models is that excited states are independent of the string used to create the excitations on the lattice, up to its homotopy class, provided the excitations remain in the same place. We give an example with the string operator \(s\) shown in Figure 3.4 (a), we begin with a ground state of the lattice \(|\psi_0\rangle\), and we apply string operator \(s\) to give excited state \(s|\psi_0\rangle\). We compare this state to \(s'|\psi_0\rangle\), where \(s'\) is shown in Figure 3.4 (b). The string operator \(s'\) is equal to \(W_1 W_2 W_3 S\), where plaquettes 1, 2 and 3 are shown in Figure 3.4 (a). It is easily checked that \(W_1, W_2\) and \(W_3\) all commute with \(s\), so it follows that \(s'|\psi_0\rangle = W_1 W_2 W_3 s|\psi_0\rangle = s W_1 W_2 W_3 |\psi_0\rangle = s |\psi_0\rangle\), which shows that an excited state is independent of the string that creates its excitations, up to

\(^1\)It’s really that black and white.
3.1 The Toric Code Model

Figure 3.4: (a) A string operator $s$ which creates two $e$ excitations at its end points. We label the positions of three primal faces 1, 2 and 3. (b) String operator $s'$. This operator is homotopically equivalent to string $s$, as it differs only by multiplication by stabilizer operators $W_1$, $W_2$ and $W_3$. 
3.1 The Toric Code Model

3.1.2. Anyons on the Toric Code Model

We finally consider in detail the anyon model generated by the excitations of the toric code model. As we mentioned in the previous Section we have excitations 1, e, m and ε. These anyons have fusion rules

\[ e \times e = 1, \quad m \times m = 1 \]
\[ e \times m = \epsilon, \quad \epsilon \times \epsilon = 1. \]

We also have the trivial fusion rules \( a \times 1 = a, \quad \forall a = 1, e, m, \epsilon \). All of these excitations are Abelian, so we have

\[ d_1 = d_e = d_m = d_\epsilon = 1. \quad (3.3) \]

We can use the string operators to create the anyons on the toric code. We can find the braiding statistics of the anyons of the toric code by looking at the commutation relations between anyons. We look at the phase a state acquires upon braiding to evaluate elements of the S-matrix of the anyon model. We remark that usually we would require a careful choice of paths, and compare the outcomes of two braids, one where anyons have been braided, and another where it hasn’t in order to correction determine elements of the S-matrix. Reference [53] provides such a scheme to distinguish the dynamical phase acquired by moving anyons around the lattice from the topological phase picked up by braiding one anyon around another. The toy model we consider does not give rise to dynamical phases, so we need worry about such details here, and we continue without worrying about such details.

Before we consider the outcome of braiding one anyon around another, we have to discuss how to move anyons across the lattice. We consider the lattice with an e and an m excitation, created by the string operators \( s_e \) and \( s_m \), respectively, such that we have the state \( s_m s_e |\psi_0\rangle \), shown in Figure 3.5 (a).

Ultimately, we will braid the m anyon around the e anyon, and return it to its initial position, but we must first consider the operators required to move anyons. In the general case, if we wish to move an anyon of type a
3.1 The Toric Code Model

Figure 3.5.: (a) A lattice in a state with an $e$ excitation and an $m$ excitation, created by strings $s_e$ and $s_m$, respectively. (b) Shown in red, the string operator used to move the $m$ anyon to an adjacent face. (c) The string operator, shown in red, that moves the $m$ anyon around the $e$ anyon, and return it to its initial position.
which lies at position \( j \), and we wish to move the anyon to site \( k \), we move \( a \) with a string operator which creates the anti-particle of \( a \), which we label \( \bar{a} \), at position \( j \), and which terminates with an \( a \) particle at site \( k \). Following fusion rules \( a \times \bar{a} = 1 \), the anyon at position \( j \) is annihilated, and an \( a \) anyon at site \( k \) is created.

In the case of the toric code, all of the anyons are their own anti-particle, and we use the standard string operators we have already discussed to move such particles around. We give an example of a string operator that moves the \( m \) anyon in Figure 3.5 (a) to an adjacent dual face. This is shown in red in Figure 3.5 (b).

Having established how to move anyons, we are now able to braid anyons around one another to find elements of the \( S \) matrix. We do this by moving the \( m \) anyon around the \( e \) anyon, and returning it to its initial position, as shown in Figure 3.5 (c). We denote the string operator that creates this path is \( s_m' \), such that the lattice is now in the state \( s_m's_m's_e|\psi_0\rangle \). The operators \( s_m' \) commutes with \( s_m \), but it anti-commutes with the \( s_e \) operator, such that we have \( s_m's_m's_e|\psi_0\rangle = -s_m's_e's_m'|\psi_0\rangle \) We note at this point that the operator \( s_m' \) is the product of the three primal \( W_p \) operators enclosed by the loop drawn by the path of the \( m \) anyon, from which it follows that \( s_m'|\psi_0\rangle = |\psi_0\rangle \), and finally we have

\[
s_m's_m's_e|\psi_0\rangle = -s_m's_e|\psi_0\rangle , \tag{3.4}
\]

showing that the state acquires a phase factor of \(-1\) upon braiding an \( m \) anyon around an \( e \) anyon which corresponds to a global phase of \( \pi \). We can repeat this method with all anyons of the model to obtain its \( S \)-matrix

\[
S_{Z_2} = \frac{1}{2} \begin{pmatrix}
1 & 1 & 1 & 1 \\
1 & 1 & -1 & -1 \\
1 & -1 & 1 & -1 \\
1 & -1 & -1 & 1 
\end{pmatrix}, \tag{3.5}
\]

where we label the rows and columns in the order \( 1, e, m, e \).
3.2. The Toric Code Model with Twists

In this Chapter we compare the topological data of twists with that of Ising anyons. In this Section we review Ising anyons, twists, and we demonstrate how to recover the fusion rules of Ising anyons using twists.

3.2.1. Ising Anyons

Ising anyons [47] are one of the simplest non-Abelian anyon models. The anyon model cannot be used for universal topological quantum computation without using additional noisy operations [20], or by dynamically changing the topology of the manifold which embed the anyons [19, 36, 58].

The model consists of three charges, the vacuum 1, the fermion, labeled ψ and the Ising anyon σ. The fusion rules of the model are

\[ σ \times σ = 1 + ψ, \]  
(3.6)

\[ σ \times ψ = σ, \]  
(3.7)

and

\[ ψ \times ψ = 1. \]  
(3.8)

The first of these fusion rules (3.6) demonstrates an essential feature which gives the non-Abelian anyon model its degenerate fusion space; it shows that when two Ising anyons are fused, there are two possible fusion outcomes, the vacuum, or the fermion.

Importantly, where anyons can give multiple fusion outcomes, the outcome is persistent. This means, if we begin from the vacuum state, and prepare two Ising anyons, when these two Ising anyons are fused, they will give the vacuum with certainty. Similarly, if two Ising anyons are prepared from a fermion, then the two prepared Ising anyons will fuse to give the fermion outcome. This guarantees that the total anyonic charge of the system is conserved. Like any particle, anyons must satisfy certain conservation laws, such as conservation of charge. If a system begins from the vacuum state, to create an anyonic charge, we must also create it corresponding anti charge. Otherwise, if we fuse all the anyons in the system, we will not recover the vacuum state. The persistency of fusion outcomes between pairs
of anyons is an extension of charge conservation.

It is this persistency that gives non-Abelian anyons their fault-tolerant characteristic. If we encode qubits in the fusion space of the anyons, errors will occur when fermions tunnel between Ising anyons. However, the probability of tunnelling at zero temperature is exponentially suppressed with separation between Ising anyons. This property enable us to robustly encode quantum information over pairs of Ising anyons, where we can measure the encoded state of the Ising anyons by fusing them together.

The fusion rule (3.7) shows that information about the number of fermions absorbed by a pair of Ising anyons is locally inaccessible by local charge measurements of individual Ising anyons. When we measure the state of a pair of Ising anyons by fusion as in (3.6), we learn the number of fermions absorbed by the pair of Ising anyons modulo 2. However, this measurement can only be achieved by fusing two Ising anyons which occurs when the two fused Ising anyons are moved close together and their collective charge is measured. Fusion rule (3.7) says that different parities of fermion number between two Ising anyons are indistinguishable by local charge measurements made on a single Ising anyon. It is this remarkable property that means it is impossible to learn the state encoded in the fusion space of a pair of Ising anyons locally, an essential feature for topological fault tolerance.

The final fusion rule (3.8) says that the fermion particle is its own anti particle, and so when two fermions fuse we obtain the vacuum particle.

3.2.2. Twists

We next consider twists on the toric code model. Consider first the toric code model introduced in Section 3.1. The plaquettes of this model are bi-coloured like a chessboard and the excitations of this model cannot move from one colour to another; $e$ anyons live only on green plaquettes and $m$ anyons live on the blue plaquettes. We draw a line along the lattice which terminates at some point. Along this line we dislocate the geometry of the lattice, and the Hamiltonian interaction terms accordingly. The termination points are the purple plaquettes, shown by the string in Figure 3.6. At the termination points of this line defect we add a five-body interaction to the Hamiltonian term where $W_t = X \otimes Z \otimes Z \otimes Y \otimes X$, also shown in 3.6, and $t$
Figure 3.6.: A line along which the lattice geometry is dislocated, such that there is no uniform bi-colouring possible over the entire lattice. The Hamiltonian interaction terms along the line are modified accordingly to match the new geometry. The points where dislocation lines terminate are known as twists, shown in purple. A five-body interaction term at a twist is shown explicitly on the left twist. Dislocation lines affect the $e$ and $m$ anyons on the lattice by exchanging their particle types when they are moved across the line. This is shown by the string operator that crosses the pink domain wall in the middle of the diagram.

is the location of the five-body plaquette. We refer to these points as twists. It is easily checked that the twist interaction terms commute with the other interaction terms of the model.

A curious result of this dislocation is that now, $e$ anyons moving across the line will become $m$ anyons and visa versa, as shown by the string operator in Figure 3.6. This provides a simple example of a domain wall, described by mapping over the excitations of the toric code model, $B$, as described in Section 2.2. We write the mapping $B$ explicitly

\[
\begin{align*}
B &: 1 \rightarrow 1, \\
B &: e \rightarrow m, \\
B &: m \rightarrow e, \\
B &: \epsilon \rightarrow \epsilon.
\end{align*}
\]
As well as non-trivially affecting anyons on the lattice, introducing pairs of twists to the lattice increases the ground-state degeneracy of the Hamiltonian. This is seen easily by counting the number of interaction terms in the Hamiltonian as twists are added. Namely, we see that the number of commuting interaction terms is reduced by one with each pair of twists on the lattice. It is instructive to look at the new degrees of freedom this introduces to the lattice. These degrees of freedom are manipulated by the stabilizers shown in Figure 3.7. These are topological degrees of freedom, as the support of the logical operators that manipulate the state encoded over the twists grows with the separation of the twists. It is the large support of the logical operators that protects the quantum degrees of freedom encoded over a twist.

### 3.2.3. Ising Anyon Fusion Rules from Twists

Having introduced twists, we conclude this Section by observing how twists on the toric code can mimic the fusion rules of Ising anyons. In the analogy, we consider twists, with charge label \( T \), as \( \sigma \) anyons, we regard \( \epsilon \) anyons as fermions, \( \psi \), of the Ising anyon model, and the vacuum will map to itself. For this discussion we ignore \( e \) and \( m \) anyons of the toric code. This gives fusion rules for the twist model

\[
T \times T = 1 + \epsilon, \tag{3.9}
\]

\[
T \times \epsilon = T, \tag{3.10}
\]

and

\[
\epsilon \times \epsilon = 1. \tag{3.11}
\]

We now show how twists give fusion rules (3.9), (3.10) and (3.11). We first consider the fusion rule (3.9). This can be seen by considering logical operators shown in Figure 3.7. We prepare a lattice with four twists in the \(+1\) eigenstate of the \( Z \) operator, shown in Figure 3.7(a). We can regard the measurement of this operator as a fusion measurement where the dionic parity of the two top twists of the diagram is measured, where as we have mentioned, dion particles behave as fermions. The \(+1\) measurement outcome corresponds to an even dionic parity. We can change the dionic parity...
Figure 3.7.: (a) and (b) show respectively possible logical operators, $X$ and $Z$, for a possible qubit encoding over four twists. Importantly both of these string-like logical operators can be deformed using stabilizers as explained in Section 3.1.1
of the two twists enclosed by the loop made by the $\mathbf{Z}$ operator by moving an $\epsilon$ particle inside the region, where one of the twists at the top of the diagram absorb the $\epsilon$ particle. The operator that changes the dionic parity is the $\mathbf{X}$ operator over the top two twists shown in Figure 3.7(b). Applying this unitary operator to the lattice will change the measurement outcome of $\mathbf{Z}$ from +1 to -1. The -1 measurement outcome corresponds to an odd parity of dions absorbed by the two fused twists. This operator physically corresponds to moving a dion string between the bottom left twist and the top left twist. As we can measure the dionic parity of a pair of twists, and change their parity by moving a dion towards one of the measured twists, we recover the fusion rule (3.9).

We next study the fusion rule (3.10). Importantly, if a $\epsilon$ particle moves towards a twist, the twist is able to absorb the $\epsilon$ particle. We see this in Figure 3.8. An $\epsilon$ particle moves towards the twist by unitary operations. It splits into an $e$ and an $m$ particle, then one of these two particles then wraps around the twist, which changes its charge, and then annihilates with the other, such that there are no excitations in the region of the twist. As there are no excitations anymore, there are no local charge measurements that can be made in the region of the twist that can detect the presence of the dion, which demonstrates the fusion rule (3.10). We note that upon absorption of a dion, the measurement outcome of fusion between two twists which includes the twist that has just absorbed a dion has changed.

The fusion rule (3.11) falls immediately from the fusion rules of the toric code model.

### 3.3. Topological Entanglement Entropy

Quite remarkably, information about the anyonic excitations of a topologically ordered Hamiltonian can be extracted by studying the entanglement of its ground states and the low energy excited states. We use the topological entanglement entropy to probe such information. Specifically, the topological entanglement entropy enables us to find the total quantum dimension of the low energy excitations of the model, as well the quantum dimensions of its individual anyons, and the quantum dimensions of possible fusion outcomes between anyons. In this chapter we use topological entanglement entropy to probe the topological data of twists. We first review the known
3.3 Topological Entanglement Entropy

Figure 3.8: A single twist that has absorbed a dion particle that arrived from the top of the diagram.

results of topological entanglement entropy. We begin by reviewing known results about measuring the von Neumann entropy of topologically ordered lattices, which leads to explain why it is necessary to perform topological entanglement entropy calculations.

3.3.1. The von Neumann Entropy

The von Neumann entropy is a measure of bipartite entanglement between two subsystems of a many-body quantum state. Consider a pure many-body quantum state \( \rho = |\psi\rangle \langle \psi| \) of \( n \) qubits

\[
|\psi\rangle = \sum_{j_1=0}^{1} \sum_{j_2=0}^{1} \cdots \sum_{j_n=0}^{1} A_{j_1,j_2,\ldots,j_n} |j_1\rangle \otimes |j_2\rangle \otimes \cdots \otimes |j_n\rangle ,
\]

where \( \sum_{j_1} \sum_{j_2} \cdots \sum_{j_n} |A_{j_1,j_2,\ldots,j_n}|^2 = 1 \). We wish to evaluate the entanglement shared between two subsystems, \( A \) and \( B \), where without loss of generality, qubits 1, 2, \ldots, \( k \in A \) and \( k + 1, k + 2, \ldots, n \in B \). We measure entanglement by calculating the von Neumann entropy of the reduced state, \( \rho_A \), of subsystem \( A \) with the formula

\[
S_A = -\text{Tr} (\rho_A \log_2 \rho_A) ,
\]

44
3.3 Topological Entanglement Entropy

![Figure 3.9: A region $R$ on a lattice whose boundary is shown in red. Here, $n_R = 2$.](image)

where we find the reduced density matrix of subsystem $A$ by taking the partial trace of subsystem $B$, $\rho_A = \text{Tr}_B \rho$. For pure state, we have $S_A = S_B$. The von Neumann entropy of $\rho_A$ quantifies the entanglement shared between region $A$ and $B$.

3.3.2. The von Neumann Entropy of a Topologically Ordered Ground State

For the ground states of quantum many-body systems with local Hamiltonians, the von Neumann entropy $S_R$ of the reduced state on a region $R$ is expected to satisfy an area law [34, 78].

Restricting to local two-dimensional systems with a gap, the von Neumann entropy of a region $R$ of a topologically ordered system is expected to have a universal correction $\gamma$ to $S_R$ such that [49, 55]

$$S_R = \alpha L_R - n_R \gamma,$$

(3.12)

where $L_R$ is the length of the boundary of region $R$ (which should be smooth), $n_R$ is the number of disconnected boundaries enclosing region $R$ and $\alpha$ is a constant that depends on the microscopic properties of the Hamiltonian. An example of a region of a lattice with two disconnected boundary regions is shown in Figure 3.9.

For a given topological model, the total quantum dimension is related to $\gamma$ via $\gamma = \log D$ [49, 55]. As such, the von Neumann entropy can be used to probe the anyonic properties of the system. This number gives an indication that the low energy excitations of a Hamiltonian are anyonic, if $D > 1$.

Remarkably, information about the quasi-particle excitations of a Hamil-
tonian can be learned by studying only the entanglement properties of the Hamiltonian’s ground state. Quantum dimensions of individual quasi-particle excitations can also be found by calculating the von Neumann entropy of excited states of the Hamiltonian. If we evaluate the region of a lattice containing a single \(a\) anyon, the von Neumann entropy is modified such that

\[
S_R(a) = \alpha L_R - \log\left(\frac{D}{d_a}\right),
\]

where \(d_a\) is the quantum dimension of anyon \(a\).

### 3.3.3. Topological Entanglement Entropy

In a general setting, it might be difficult to accurately measure the length of a boundary. In particular, on lattices where the boundaries are not necessarily smooth, this can have an effect on the topological corrections obtained in Equation (3.12) and (3.13). It is useful then to eliminate the the boundary contribution of the von Neumann entropy of regions of a topologically ordered model when trying to identify topological characteristics. This is achieved by computing a linear combination of the von Neumann entropies of overlapping regions as in Figure 3.10 (a) to obtain the topological entanglement entropy

\[
S_{\text{topo}} = S_A + S_B + S_C - S_{A \cup B} - S_{B \cup C} - S_{C \cup A} + S_{A \cup B \cup C},
\]

where \(S_R\) is the von Neumann entropy of region \(R\) [49, 55].

The topological entanglement entropy is a function solely of the total quantum dimension of the model,

\[
S_{\text{topo}} = -\log D.
\]

This calculation is insensitive to the presence of anyonic excitations on the lattice.

### 3.3.4. Evaluating Quantum Dimensions of Anyons by Entropy Calculations

We may desire to learn data about anyons that can appear on a topologically ordered lattice model. Additional information about the anyonic properties
3.3 Topological Entanglement Entropy

Figure 3.10: (a) Regions $A$, $B$, $C$ used for calculating the topological entanglement entropy. (b) The annular topological entanglement entropy. The topological entanglement entropy calculated using these regions is sensitive to topological charges on region $D_1$.

of the model can be revealed by defining analogous entropic quantities using regions with noncontractable boundaries [84], as in Figure 3.10 (b). Again, the boundary contribution can be removed with a suitable combination of the entropies on a nontrivial partition; a suitable choice is an annular region shown in Figure 3.10 (b) defining a disconnected region $D = D_1 \cup D_2$. For an excited state of the Hamiltonian where we have an anyon of type $a$ on region $D_1$, and its corresponding anti particle in region $D_2$, and the rest of the lattice is in the vacuum state, the entropy

$$S_{\text{ann}}(a) = S_A + S_B + S_C - S_{A \cup B} - S_{B \cup C} - S_{C \cup A} + S_{A \cup B \cup C}$$  \hspace{1cm} (3.16)$$

gives a value

$$S_{\text{ann}}(a) = -2 \log \frac{D}{d_a},$$  \hspace{1cm} (3.17)$$

where the factor of 2 arises from the boundary connectivity as in Equation (3.12).

In this Chapter we use $S_{\text{ann}}$ to infer fusion rules for both anyons and twists. For this purpose we must determine how $S_{\text{ann}}$ behaves in the situation where there are multiple anyons in region $D_1$. If the fusion space of the anyons, $\mathcal{H}_{\text{topo}}$, is in a state where the anyons $a_1, a_2, \ldots$ in $D_1$ fuse to a definite fusion product $x_1$, then $S_{\text{ann}}(a_1, a_2, \ldots) = S_{\text{ann}}(x_1)$. However, in general, anyons can fuse through multiple channels, with the probability
amplitude of a given fusion outcome determined by the state encoded in \( \mathcal{H}_{\text{topo}} \). Considering only states where the connected region \( ABC \) as well as the entire lattice fuse to the vacuum, the outcome \( x_1 \) determines explicitly \( x_2 \), the fusion product of the anyons on \( D_2 \). In this important special case we find the explicit formula

\[
S_{\text{ann}}(\{x\}) = -2 \log D - \sum_x P_x \log[P_x/(d_{x_1}d_{x_2})],
\]

where the sum is over all fusion outcomes \( x = (x_1, x_2) \), each of which occur with probability \( P_x \). This equation provides the principal diagnostic we will use to characterise the anyonic properties of a model’s quasi-particles and defects.

3.4. Topological Entanglement Entropy with a Twist

In this Section we show the topological entanglement entropy of a model with twists using annular regions with a non-contractable boundary will yield identical results to the analogous entropic calculation using excited states of a model with Ising anyons in place of twists. All of the calculations here are exact, and use stabilizer formalism to calculate von Neumann entropies for regions of the vacuum and excited states of the toric code model with twists, using the method of Reference [35].

3.4.1. The Topological Entanglement Entropy of a Lattice with Twists

We begin by considering the toric code model with twists on the lattice and show that we find the total quantum dimension of the Ising anyon model. We first calculate the von Neumann entropy for a simply connected region enclosing a single twist, as shown in Figure 3.11. We will find that \( S_A(T) = \frac{1}{2}L_A - \frac{1}{2} \), as expected for a region of an Ising anyon model containing an \( \sigma \)-type Ising anyon with \( d_\sigma = \sqrt{2} \). We now elaborate on the calculation.

We define the length of the boundary \( L_A \) to correspond to the number of plaquettes that cross the boundary. Figure 3.11 shows the boundary of region \( A \) in red, which has length \( L_A = 17 \).
We can exploit the stabilizer formalism to calculate von Neumann entropies for regions of the vacuum and excited states of these topological models, using the method of Ref. [35]. This method makes use of a canonical form of a generating set of the stabilizer group defined by a bipartition of the lattice into regions $A$ and $\overline{A}$. The stabilizer generators can be grouped into those entirely in $A$, those entirely in $\overline{A}$ and those that have support on both partitions. For the set of stabilizer generators that cross the bipartition, a canonical form can be constructed such that the local support of the generators $K_i$ on $A$, denoted by $\tilde{K}_i$, either commutes with all other $\tilde{K}_j$, or anti-commutes with only one other such element $\tilde{K}_\ell$. These pairs of stabilizer generators that anti-commute when restricted to $A$ describe a pair of maximally entangled qubits sitting across the boundary. Counting these pairs gives the von Neumann entropy $S_A$.

Although there exist stabilizers that take the form of dion strings with ends terminating at twists, we can multiply by local stabilizers to deform such strings into loops encircling the twist in region $A$ using methods described in Section 3.1.1, as shown in Figure 3.8 such that they have no support on $A$.

We next explain how to find the canonical form of the generators cut
by \( \partial A \). For plaquette operators \( W_l \) that cross the partition \( \partial A \), we define \( \tilde{W}_l \) to be the support of these operators on \( A \), labelling them with a single index \( 1 \leq l \leq L_A \) in a sequential order, as shown in Figure 3.11. The \( W_l \) operators are not in canonical form, as each \( \tilde{W}_l \) anti-commutes with both \( \tilde{W}_{l-1} \) and \( \tilde{W}_{l+1} \). To achieve canonical form, we replace one generator, \( W_1 \), with the generator \( \prod_l W_l \), where the product is over all stabilizers with support on the boundary of \( A \); the support of the operator \( \prod_l \tilde{W}_l \) on region \( A \) commutes with all \( \tilde{W}_l \). (Note that, in the untwisted toric code, one can construct two such operators using the natural partition of toric code stabilizers into two commuting subsets. With the presence of a twist, however, there is only a single such operator.) The remaining \( \tilde{W}_l \) can be transformed such that they all pairwise anti-commute, giving the desired canonical form. Explicitly, this is achieved by replacing all the generators \( W_l \) with even \( l \), with the generator \( \prod_{j=1}^{L_A} W_j \). Counting the pairs of locally anti-commuting generators then gives the expected result

\[
S_A(T) = \frac{1}{2}L_A - \frac{1}{2}.
\]  

(3.19)

We note that this result, and following results in this Chapter, are independent of the topology of the lattice. This is because homologically non-trivial stabilizers, such as those that we will discuss in later Chapters of a lattice embedded on a topologically non-trivial manifold can always be deformed away from the regions considered in the present entropy calculations [41] using string deformations described in Section 3.1.1. With this in mind, we consider performing all the following calculations on a manifold whose topology is equivalent to that of a disk, whose ground state is non degenerate.

This method allows us to calculate the effect of twists on the topological entanglement entropy of the lattice, using Equation (3.14) and regions defined as in Figure 3.10(a). We find that for a pure vacuum state of a model with any number of twists, the topological entanglement entropy is \( S_{\text{topo}} = -1 \) independent of the locations of the twists. This outcome corresponds to a model with total quantum dimension \( D = 2 \), consistent with a model which supports Ising anyons.

We give the example lattice shown in Figure 3.12. We need to consider the von Neumann entropy of all the regions \( A, B, C, A \cup B, B \cup C, C \cup A \)
3.4 Topological Entanglement Entropy with a Twist

Figure 3.12: A twist on region $A$ and a twist on region $C$ of the topological entanglement entropy regions of 3.10(a). In this calculation we obtain $S_{\text{topo}} = -1$.

and $A \cup B \cup C$. We know that a region of the toric code with no twists has $S_R = \alpha L_R - 1$, due to the anyons of the model. For this reason for region $B$ we have $S_B = \alpha L_B - 1$. Similarly, we can use that $S_R = S_T$ to find the entropy of region $A \cup B \cup C$ and $C \cup A$, because regions $D$ and $B \cup D$ contain no twists, and so give $S_{C \cup A} = \alpha L_{C \cup A} - 1$ and $S_{A \cup B \cup C} = \alpha L_{A \cup B \cup C} - 1$.

The remaining regions $A$, $C$, $A \cup B$ and $B \cup C$ each contain only a single twist, so using Equation (3.19), we have $S_R = \alpha L_R - 1/2$, $\forall R = A$, $C$, $A \cup B$ and $B \cup C$. Finding the topological entanglement entropy we then have

$$S_{\text{topo}} = S_A + S_B + S_C - S_{A \cup B} - S_{B \cup C} - S_{C \cup A} + S_{A \cup B \cup C} = -1, \quad (3.20)$$

where the length contributions cancel exactly. In general we will always obtain this result, provided all the topological degrees of freedom are in an eigenstate of a logical operator, such as those in 3.7.

We note that, the total quantum dimension is the same as that which we expect where the lattice has no twists.

3.4.2. The Annular Topological Entanglement Entropy of a Twist

We next show that, using the annular regions with non-connected boundaries as in Figure 3.10(b) which can probe the properties of anyons, the entropy $S_{\text{ann}}(T)$ for annular region of a model containing twists is equivalent to the entropy $S_{\text{ann}}(\sigma)$ of a state of an Ising anyon model with $\sigma$-type Ising anyons. We use a partitioning as in Fig. 3.13, with a twist on region $D_1$ and another on $D_2$. In this calculation, we can always deform logical
operators outside the region of interest using string deformations described in Section 3.1.1, similar to that shown in Figure 3.8. For this reason we do not worry about logical operators in this calculation. We achieve canonical form for region $ABC$ using the method described above, but now with disconnected boundaries we must consider the canonical form of the stabilizer generators for each boundary separately. As above, any dion string stabilizer that crosses region $ABC$ can be deformed entirely onto region $D_2$.

Calculating the entropy for the various regions, we find $S_{ABC} = \frac{1}{2} L_{ABC} - 1$, with all other regions giving $S_R = \frac{1}{2} L_R - 1$, and so $S_{\text{topo}} = -1$. We compare this result to a calculation of $S_{\text{ann}}(\sigma)$ for a state of an Ising anyon model with a $\sigma$-type Ising anyon in each region $D_1$ and $D_2$, using Equation (3.17), and obtain an identical result, $S_{\text{ann}}(\sigma) = -1$. Thus, from an entropic perspective, the twists appear as $\sigma$-type Ising anyons. A similar calculation involving a $\epsilon$ quasi-particle on region $D_1$ and another on region $D_2$ gives $S_{\text{ann}}(\epsilon) = S_{\text{ann}}(\psi) = 1$ as expected from a Ising fermion $\psi$ in the Ising anyon model.

### 3.4.3. The Annular Topological Entanglement Entropy Pairs of Fused Twists

Finally, we use TEE to probe the fusion rules associated with quasi-particles and twists. We show that pairs of twists fuse according to the more sophisticated non-Abelian fusion rules of two $\sigma$-type Ising anyons, and modify the TEE according to our Equation (3.18) in a way that depends on the specific fusion channel. We place two twists on region $D_1$ and two twists on region $D_2$. The channel by which the two twists on region $D_1$ fuse together is dependent on the logical state, as characterised by the logical operators...
Figure 3.14: (a) The lattice configuration where two twists lie on $D_1$ and two lie on $D_2$, in an eigenstate of $Z$ defined as a dion chain stabilizer as that shown in Figure 3.7 (b). (b) A configuration with four twists in an eigenstate of $X$, shown microscopically in Figure 3.7 (a).

that are given by dion strings between twists.

We first consider the lattice prepared in an eigenstate of the logical-$Z$ operator $Z$ shown in Figure 3.14 (a). The $+1$ eigenstate of $Z$ corresponds to two $\sigma$-type anyons on region $D_1$ fusing to the vacuum 1 with certainty, whereas the $-1$ eigenstate fuses to the fermion $\psi$ with certainty. In either case, we find $S_{\text{ann}}(\{\sigma, \sigma \rightarrow 1\}) = S_{\text{ann}}(\{\sigma, \sigma \rightarrow \psi\}) = -2$. Comparing this with Equation (3.18), we have the result we anticipate from the Ising anyon model.

We next consider the case where the lattice is prepared in an eigenstate of the logical-$X$ operator $\bar{X}$ shown in Figure 3.14 (b). This is analogous to the Ising anyon model where one $\sigma$-type anyon on region $D_1$ and one $\sigma$-type anyon on region $D_2$ will fuse to give one fusion outcome with certainty. However, fusing the two particles on region $D_1$ can produce the vacuum 1 or a fermion $\psi$ with equal probability $P_{(1,1)} = P_{(\psi, \psi)} = \frac{1}{2}$. The details of the TEE calculation are more involved in this case, as the operator $\bar{X}$ cannot be deformed away from region $ABC$. We therefore require a different method to find a canonical form for region $ABC$. Denoting the two simply connected segments of the boundary by $\partial ABC_\beta$, for $\beta = 1, 2$, where $\partial ABC_\beta$ is the boundary for region $D_\beta$, we rewrite the indices of the plaquettes cut these boundaries as $W_{\beta l}$, with $1 \leq l \leq L_\beta$ numbering the generators around the boundary as shown in Figure 3.15. We deform $\bar{X}$ such that it only has common support with the generators $W_{1,1}$, $W_{1,2}$, $W_{2,1}$, and $W_{2,2}$ on the boundaries. The is achieved by deforming the logical operator onto the qubits of the cyan band on Figure 3.15. To achieve canonical
Topological Entanglement Entropy with a Twist

Figure 3.15: The two disjoint boundaries of region $ABC$ are shown in red where two twists are entirely on region $D_1$ in the centre of the diagram, and the other two twists lie on region $D_2$. The logical operator $X$ shown in Figure 3.7 (a) is deformed to have support only on the cyan band connecting two twists, which overlaps only with cut stabilizer generators $W_{1,1}$, $W_{1,2}$, $W_{2,1}$ and $W_{2,2}$. The cut stabilizers are labeled by two numbers, the first denotes the boundary which cuts the stabilizer, and the second numbers the different stabilizers cut by the same boundary.
form, we replace the generators $W_{1,1} \rightarrow \prod_{\partial ABC_1} W_{1,l} / W_{2,1} \rightarrow \prod_{\partial ABC_2} W_{2,l}$, and $W_{1,2} \rightarrow \prod_{\partial ABC_1,k=\text{even}} W_{1,k} \prod_{\partial ABC_2,l=\text{even}} W_{2,l}$, such that their local support commutes with the local support all the other generators. The remaining cut generators, including $X$, are then transformed such that their local supports pairwise anti-commute, giving the required canonical form. This gives the result $S_{ABC} = \frac{1}{2} L_{ABC} - 1$. The other regions used are equivalent to regions of the homogeneous toric code model and fuse to vacuum, giving $S_{\text{ann}}(\{(\sigma, \sigma) \rightarrow 1 \text{ or } \psi\}) = -1$, which from Equation (3.18) is the result we expect from the analogous Ising anyon case.

### 3.5. Conclusions

In this Chapter we have used topological entanglement entropy calculations to show further that the twists on the toric code model behave like non-Abelian Ising anyons. In particular, we have shown that the toric code model with twists reproduces the same total quantum dimension as the Ising anyon model. We probe also the individual particles within the analogy, and we show that twists reproduce the same quantum dimensions as the Ising anyon model, and also that the fusion products of two twists give the entropy results we expect, which depends on the state encoded over the twists.

The parallel we have extended is quite remarkable, considering that the toric code model only gives rise to Abelian anyons. Curiously, similar parallels between toric code anyons and Ising anyons have been drawn before in Reference [81], where interestingly, superpositions of toric code states give rise to certain behaviours of the Ising anyon model. This draws questions as to whether the studied comparison between these two topologically ordered phases is more fundamental than we currently appreciate.

Importantly, the work we have presented in this Chapter shows that properties of twists related to their computational capacity can be learned by studying them entropically. Here we have only considered a simple model with a limited computational capacity. We chose this system as it is rigorously solvable, which has enabled us to achieve exact results. The methods adopted here, where we use entanglement measures to study the computational ability of a lattice model with defects could potentially be extended to find lattice models with line defects that are capable of fault-tolerant quantum computation.
To complete the analogy we have studied, one might seek to identify the braiding properties of twists. Techniques to probe such behaviour using entropic calculations using anyons is shown in Reference [84]. However, as noted by Bombin [12], braiding of twists are not expected to be well defined, as twists break translational invariance in the lattice. For this reason we have not considered this calculation in this work.
4. Qudit Toric Code Decoding Algorithms

It has been mentioned frequently throughout this thesis that topologically ordered models show strong tolerance against local noise. However we have not mentioned how topologically ordered models achieve such robustness. In the present Chapter we take a small interlude from lattice defects to reflect on how globally encoding information in topologically ordered quantum systems introduces a natural protection against stochastic local noise. We demonstrate this robustness by considering error correcting codes [64, 66], and in particular, we consider decoding algorithms used to recover information stored in topologically ordered states. We remark that the HDRG decoder developed in this Section is used in Chapter 5 to analyse the thermal stability of a topological quantum memory with line defects.

A quantum error correcting code [66] is a many-body entangled state of physical quantum systems which are subject to some physical noise channel. Logical information is encoded in a carefully chosen subspace of the Hilbert space of the system. The subspace is chosen in such a way that the information encoded in this space is not easily affected by the local noise which attacks the entangled physical systems. Protection against noise is then achieved by performing syndrome measurements on the Hilbert space where logical information is not encoded. The classical information learned from performing syndrome measurements is then used to learn what errors are likely to have occurred on the physical systems of the error correcting code. A particularly exciting class of error correcting codes are known as topological codes [28, 50], which are very closely related to the Hamiltonian models we discuss in this Thesis.

A simple classical analog of a quantum error correcting code is known as the repetition code. If we wish to store a classical bit of information using physical bits which may be subject to noise, the classical information can
be duplicated over many physical bits. The information can be read out by making a majority vote of the state of all the physical bits. Provided the noise the physical bits are subject to is below a certain *threshold*, we can recover the information encoded over the physical bits with arbitrary precision by increasing the number of physical bits we use to encode the information. We note that we cannot build a quantum repetition code, as this is forbidden by the no-cloning theorem. Fortunately, there exist quantum error correcting codes which are proven to have a threshold, where arbitrarily precise quantum computation can be achieved by increasing the size of the quantum error correcting code, provided the noise the physical quantum systems are subject to is below a certain threshold [1].

Here we consider two-dimensional topological codes [50]. Topological codes are particularly elegant codes; logical information is encoded over homologically non-trivial cycles of physical systems embedded in a high genus manifolds. The high genus manifold we consider is the torus, shown in Figure 4.1. The code is locally embedded in the manifold. For this reason, topologically codes are physically feasible to achieve, as syndrome measurements are also performed locally which provides experimental amenability with certain architectures in the laboratory.

Upon performing syndrome measurements on the encoded lattice which has suffered errors, we seek to find an operation we can perform on the lattice which reverses the incurred noise to recover the initially encoded state. To deal with the obtained syndrome information and calculate a suitable
operation to retrieve the encoded state, we require some classical algorithm. Such an algorithm is known as a decoding algorithm [14, 16, 28, 30, 31, 43, 65, 82]. It is important to find good decoding algorithms which process this syndrome information quickly, and with high success rates. Speed is important as we do not wish for the quantum error correcting code to suffer further errors while the decoding algorithm takes place, and we require high success rates to achieve high performance quantum computation with a low overhead of physical resources. Furthermore, the threshold of an error correcting code is highly dependent on the decoding algorithm used to recover its encoded state.

In this Chapter we consider two different high-speed decoding algorithms based on renormalisation-group techniques which are capable of decoding the qudit toric code. Namely, the hard-decision renormalisation-group (HDRG) decoder [16, 42] and the soft-decision renormalisation-group (SDRG) decoder [30, 65]. In particular, we study how the performance of these decoding algorithms improve as we change the dimension, $d$, of the physical qudits over which logical qudits is encoded. We remark that the SDRG decoder used here is a variant of that used in Reference [33], which is a generalisation of the decoder originally described in [30, 31]. We note that we developed the decoder used here independently of that used in Reference [30].

We remark that the decoders considered here use only perfect measurement information. A more realistic physical system will receive incorrect measurement results. Generalisations of these decoders which deal with incorrect syndrome measurements can be produced [32, 71], but in this Section we are interested only in the different methods of dealing with syndrome information. For this purpose it is sufficient to consider only the perfect measurement case.

We note also that algorithms developed in this Chapter, namely the HDRG decoder, is also used the following Chapter is used to examine and compare the performance of quantum memories with defect lines that we subject to thermal noise. A quantum memory is a Hamiltonian realisation of the quantum error correcting codes discussed in this Chapter, which rely on energy barriers introduced by the Hamiltonian interactions to suppress dynamical errors to information encoded within their degenerate ground states. In particular, the decoding algorithms developed here enable us to show numerically that introducing defect lines to the qudit toric code Hamil-
tonian introduce significant qualitative improvements to the coherence times of topological quantum memories.

In this Chapter we review the qudit toric code. We discuss the different methods the two considered decoders use to process the syndrome information and return a recovery operation. We then review the methods we use to find the threshold points for the different decoders at different $d$. In the final Section we discuss the different thresholds we obtain using these methods, and analyse why we should expect the obtained performance.

### 4.1. The Qudit Toric Code

In this Section we review the qudit toric code [50]. The qudit toric code is a stabilizer code. A stabilizer code protects a logical Hilbert space encoded in a larger physical Hilbert space of $d$-level physical systems, qudits. The encoded logical Hilbert space is known as the codespace. The codespace of a stabilizer code is described by the stabilizer group. States encoded in the codespace are the common +1 eigenstates of elements of the stabilizer group.

The stabilizer group is an Abelian subgroup of the $n$-qudit generalised Pauli group $P_d^n$ which is the $n$-fold tensor product of the one-qudit generalised Pauli group. The one-qudit generalised Pauli group is generated by the operators $X$ and $Z$ where

$$
X | j \rangle = |(j + 1) \text{mod} d \rangle \quad Z | j \rangle = \omega^j | j \rangle
$$

(4.1)
as well as the phase $\omega$ for odd $d$ and $\sqrt{\omega}$ for even $d$, where $\omega = e^{2\pi i/d}$. The operators $X$ and $Z$ are such that $ZX = \omega XZ$. The $n$-qudit generalised Pauli group then is $P_d^n \equiv P_d \otimes_n$.

#### Stabilizers of the Qudit Toric Code

The qudit toric code is described on a square lattice of $L \times L$ vertices where qudits are placed on the edges of the lattice, shown by white points in Figure 4.2. The model has periodic boundary conditions to achieve the desired topology shown in Figure 4.1. The generators of the stabilizer group of the qudit toric code are vertex operators $A_v = X^\dagger \otimes X \otimes X \otimes X^\dagger$ and plaquette operators $B_p = Z \otimes Z \otimes Z^\dagger \otimes Z^\dagger$, shown in blue and red respectively in Figure 4.2. We have one vertex operator for each vertex of the lattice $v$.
4.1 The Qudit Toric Code

Figure 4.2.: An $A_v$ and a $B_p$ operator on the qudit toric code lattice, shown in red and blue respectively.

Figure 4.3.: (a) Logical operators $\bar{X}_2$ and $Z_1$ are shown in red and blue, respectively. (b) Logical operators $\bar{X}_1$ and $Z_2$ are shown in red and blue, respectively.

and one plaquette operator for each face of the square lattice $p$.

**Logical Operations on the Qudit Toric Code**

The qudit toric code encodes two logical qudits on the surface of the torus. Logical information on the torus is manipulated with string operators of generalised Pauli matrices which wrap around homologically non-trivial cycles of the lattice, as shown in Figure 4.3. Importantly these operators commute with all the syndrome measurement operators. It is for this reason that we are able to perform stabilizer measurements without collapsing the quantum state encoded in the lattice.
Syndrome Measurements on the Qudit Toric Code

Error correcting codes are designed to tolerate noise up to a certain threshold. We perform syndrome measurements on the code to try to learn the locations of physical errors on the code. We perform projective syndrome measurements $P_v(a)$ and $Q_p(a)$ on the code, where we write projectors as polynomial functions of vertex and plaquette operators

$$P_v(a) = \frac{1}{d} \sum_{k=1}^{d} e^{2ak\pi i/d} A_v^k, \quad Q_p(a) = \frac{1}{d} \sum_{k=1}^{d} e^{2ak\pi i/d} B_p^k. \quad (4.2)$$

We perform these measurements for all $v$, $p$ and $a = 0, 1, \ldots, d - 1$. These measurements project the encoded state onto an eigenstate of the stabilizers of the code, although not necessarily into a state in the codespace of the code. Specifically, these operators will project the lattice into a state where the encoded state has suffered an error $E$, where $E$ is an operator of the form of a tensor product of $X_e^j$ and $Z_e^j$ operators over edges $e$ of the lattice. The measurements will also return classical information, which we call syndromes, that we use to determine the positions of errors the lattice have suffered [66]. Specifically, each syndrome measurement $P_v(a)$ and $Q_p(a)$ will return an integer which takes values $0, 1, \ldots, d - 1$ for each vertex $v$ and each plaquette $p$. A decoder uses this classical information to attempt to find the correct recovery operator such that we are able to recover the encoded
4.1 The Qudit Toric Code

state in the code. The remainder of this Section will discuss incident errors on the code, and the syndromes they generate.

We first consider an error incident to the lattice in Figure 4.4. Here we see a lattice in a state which has been taken out of the code space by an error \( E \) of \( X \)-type and \( Z \)-type operators, shown by a red string and a blue string. Performing the syndrome measurements over all the vertices \( v \) and faces \( p \) will return syndrome information of where errors have occurred on the faces and plaquettes of the lattice. We also show the non-zero syndrome outputs on the lattice.

The toric code is a CSS code [25, 67]. A CSS code is such that stabilizer generators can be written either only \( Z \)-type Pauli operators or only \( X \)-type Pauli operators. This enables us to consider \( X \)-type errors and \( Z \)-type errors independently of one another. This is because \( X \)-type noise will commute with the vertex operators, and hence the \( P_v(a) \) syndrome measurements, and \( Z \)-type noise will commute with the plaquette operators of the code and the \( Q_p(a) \) syndrome measurements. Indeed, in Figure 4.4, and subsequent diagrams in this Chapter, syndrome information about \( Z \)-type errors is written on the vertices of the lattice, and syndrome information about \( X \)-type errors is written on the faces of the lattice. As we are able to consider the different errors equally and independently, we restrict our attention to only the errors of type \( X_j^e \) for the remainder of this Chapter.

We now consider what we can learn about the error from the syndrome information. First of all, as we have mentioned, the code is local, we see this by the stabilizers which occupy a small local space in Figure 4.2. Due to this, non-trivial syndrome data, by which we mean non-zero measurement outcomes, will only be caused by errors local to the syndrome.

More generally, Pauli errors form string-like objects on the lattice. Strings in general commute with stabilizers on the lattice. For instance, both errors in Figure 4.4 are string-like, and they each have a common support with two stabilizers which detect their respective type of error, but do not produce any syndrome information. In fact, only the end points of string-like errors produce non-trivial stabilizer information. In some sense this is a limitation of the code and is, in fact, a fundamental limitation of the topological error correcting code we consider here [21].

A second important observation is that the star and plaquette operators satisfy the property \( \prod_v A_v = 1 \) and \( \prod_p B_p = 1 \). It follows from this that
the sum of all the syndromes on the faces of the lattice, and the sum of the syndrome measurements on the vertices of the lattice, must both sum up to $0 \mod d$. This constrains significantly the different types of errors that can occur on the lattice. Due to the locality of the stabilizer operators of the code, it follows from this observation, if an error remains small and local compared to the size of the lattice, the sum of the syndrome measurements within a local patch will also sum to $0 \mod d$. We can use these observation to correct for a large class of small local errors.

These observations also give some intuition for why increasing the qudit dimension will enable the decoder to improve its performance. If the syndromes generated by a small local error must sum to $0 \mod d$, we expect for higher $d$ a single local patch of errors will generate more syndrome information over more faces. A good decoder will be able to exploit this information to achieve more impressive thresholds with higher $d$.

**Correcting Errors on the Lattice using Syndrome Information**

Upon learning the syndrome information, and assuming that errors are sufficiently small and local, we use the information to find a correction operator, $C$, which we expect will recover the initial state. We are not allowed to learn individual positions of errors that caused the syndrome, only the syndrome information itself. However, in the consideration of this Section, we allow ourselves to know the error $E$ which has generated the syndrome. We consider the X-type error shown in blue in Figure 4.4. We consider two different potential correction operators we might generate upon learning the syndrome information, one correction operator will recover the initial state of the lattice successfully, $C'$, shown in Figure 4.5 (a), and the other correction operator, $C''$, will correct the lattice with a logical operation having been performed on the initial state in the code space, shown in Figure 4.5 (b).

The important difference is that the operator $C'E$ forms a homologically trivial loop over the torus. By homologically trivial, we mean that we can contract the loop into a single point on the torus. Another way to see this is we can construct the operator $EC'$ entirely out of stabilizer operators. The operator $EC''$ on the other hand is homologically non-trivial, and cannot be contracted to a point as $EC'$ can. As we mentioned earlier in this Section, logical information is encoded over homologically non-trivial cycles on
4.1 The Qudit Toric Code

![Diagram](image)

Figure 4.5: (a) and (b) show the operators $C'E$ and $C''E$ respectively. The correction operator $C'$ and $C''$ are shown by Pauli matrices in red circles. The error operator $E$ is shown with Pauli operators in black circles in both diagrams.

the torus, and the logical information is affected by operators acting over homologically non-trivial cycles of the code.

Another way to understand the correction operators is to look at the commutation relations between $EC$ with the stabilizers and the logical operations. Importantly we see the operators $EC'$ and $EC''$ commute with all the $A_v$ and $B_p$ operators. This shows both correction operators return us to the code space. However, only $EC'$ also commutes with all the logical operators, which means that only correction $C'$ will leave the logically encoded information invariant. The other correction operator, $C''$ does recover the encoded state, as it will not commute with the logical operator $Z_2$, shown in Figure 4.3 (b).

The difference between $C'E$ and $C''E$ is that the operator $C'E$ forms a homologically trivial cycle, or in other words, this operator can be contracted to a point, which in this case is the identity operator by applying only stabilizer operators $A_v$. This is not true for the operator $C''E$. This operator forms a non-trivial cycle around the torus, such as the one shown in Figure 4.1 which affects states in the code space of the lattice. Cycles such as this cannot be written as a product of stabilizer operators.

The types of errors that are correctable are those where the probability of an error occurring is low. In this regime, the most likely correction operator is that with the lowest weight. We therefore operate under the hypothesis that the best correction operator is the correction operator with
the lowest weight. In the examples we discuss above, the correction operator which recovers the encoded state is the correction operator with the lowest weight. In general it is difficult to find the lowest weight correction operator. The decoders we consider efficiently estimate the most probable error configuration.

The remainder of this Chapter looks at two different algorithms that use syndrome information and knowledge of the noise model to attempt to find the optimal correction operator. Both decoders we consider are optimised for high speed decoding and use renormalisation techniques to process the syndrome data. The first decoder is a hard-decision renormalisation group (HDRG) decoder and the other is is a soft-decision renormalisation group (SDRG) decoder. The names are chosen [32, 65] because, as the lattice is renormalised, the HDRG decoder will decide on correction operators to apply throughout the algorithm, whereas the SDRG decoder will only assign probabilities to different potential correction operators as the algorithm progresses, and finally decide what correction operator to apply when the algorithm is complete. The main focus of the study is a comparison in the thresholds of the decoders as dimension \(d\) increases. Although one would expect the SDRG decoder to perform better than the HDRG decoder, due to the computational simplicity of the HDRG decoder, which we explain in detail in the following Sections, it is remarkable that the HDRG decoder can perform comparably well and in some cases slightly better than the SDRG decoder. The HDRG decoder is a particularly interesting point of study, due to its simplicity. Indeed, this decoder is the only decoder which is proved to have a non-zero threshold analytically. Other decoders must suffice at having only numerical data demonstrating a threshold [16].

### 4.2. HDRG Decoder

The HDRG decoder is a very simple algorithm, and relies very much on the assumption we discussed in the previous Section that errors will be low weight, spatially small and local. It considers all the syndromes on the lattice, and it attempts to cluster them together over different levels of renormalisation until all the clusters can contain a correction operator. The algorithm is easily explained using an example syndrome, such as the one shown in Figure 4.6. The decoder is not able to consider the error that
4.2 HDRG Decoder

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d - 1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$d - 2$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$d - 2$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.6.: A lattice with an example syndrome. We omit physical qudits from the diagrams in this Section, as the decoder does not have access to error information on the edges.

may have caused such a syndrome, and so we omit it from the diagram. Nevertheless, the there is an obvious error which most likely caused the syndrome on the lattice.

4.2.1. Clustering a Syndrome

The decoder begins by putting each of the syndromes in a cluster, or box, as shown over the example syndrome in Figure 4.7(a). This is level-0 clustering. At the end of each level of clustering the decoder will attempt to find a correction operator that will neutralise all the syndromes within the cluster where the correction operator has support only inside the cluster. It is easy to determine if the syndrome can be neutralised inside the cluster, as the sum of all the syndromes mod $d$ will be 0. Obviously, at the level-0 clustering, no clusters can be neutralised.

Figures 4.7 (b) and (c) show level-1 and -3 clustering. In the diagrams in Figure 4.7, neutralised clusters are drawn in blue boxes, and otherwise they are drawn in red boxes.

At each level the clusters are increased in size at each level by taking each syndrome within each of the boxes, and then searching within a fixed radius of its face. If a syndrome finds another non-trivial syndrome inside
Figure 4.7: (a), (b) and (c) show the broom decoder at clustering the example syndrome at levels 0, 1 and 3 respectively.
4.2 HDRG Decoder

Figure 4.8.: (a) A lattice with a difficult syndrome to find a correcting operator by considering only the weights of different correction operators. (b) Two different low-weight correction operators which are homologically equivalent. (c) A correction operator with no degeneracy.

another cluster, the clusters are combined. Once all the syndromes of the unneutralised clusters have attempted to find syndromes within other clusters within the fixed radius of the given level, the new clusters are checked if they can be neutralised. Once clusters have been neutralised, a correction operator is found which has support only inside a the smallest square box which contains the cluster.

Clusters which remain unneutralised are left for higher levels. Figure 4.7 (b) shows a syndrome at level-1 clustering where the cluster in the top-left corner has been neutralised, whereas the other clusters remain active. It is not until level-3 clustering that all the clusters are neutralised. The remainder of this Section will describe how we choose to grow the cluster radius with level, and how correction operators can be found, which completes the operation of the decoder.
4.2.2. Optimising Decoding by Degeneracy

For certain error configurations it is beneficial to use decoders that consider characteristics of correction operators other than their weight. For instance, sometimes there are many low-weight correction operators which correct the same error. In this instance, the correction operator has a high degeneracy. Optimising over degeneracy, as well as the weight of a correction operator will optimise the performance of a decoder. Again, we consider degeneracy by example. Consider the syndrome of Figure 4.8 (a) on the $d = 2$ qudit toric code, where syndrome measurements are either 1 or trivial. We consider two possible correction operators for this syndrome, that shown in blue in Figure 4.8 (b), and that shown in Figure 4.8 (c).

If we consider only the weight of the correction operator to determine the best correction operator, then the correction shown in Figures 4.8 (b) and (c) are indistinguishable. However, the best correction operator to apply is that shown in Figure 4.8 (b). This is because there are more low-weight error configurations that can be corrected by the correction in Figure 4.8 (b) than there are in Figure 4.8 (c). In other words, the correction shown in Figure 4.8 (b) has a greater degeneracy than that shown in Figure 4.8 (c).

An example of such an alternative error configuration is shown in green in Figure 4.8 (b). In fact, it is easily seen that there are four low weight-four error configurations that can be successfully corrected for by the operator shown in blue in Figure 4.8 (b) whereas there is only one weight-four error that will be corrected by the correction operator of Figure 4.8 (c). It is therefore optimal to decode by considering degeneracy, as well as the weight of a correction operator.

4.2.3. Cluster Levels

We use a metric to ensure that at each clustering level, we search within the appropriate radius. A carefully chosen metric can optimise the clustering levels to deal with degeneracy.

Each syndrome which is still within an unneutralised cluster will search within a certain distance $r$ of the syndrome. On a square lattice we can use a Manhattan distance, where $r = r_1 + r_2$, where $r_1$ and $r_2$ are measured in lattice units. It will be convenient to use the convention $r_2 \leq r_1$, as shown in Figure 4.9 (a).
4.2 HDRG Decoder

We could use a metric that searches distance \( r \) from each syndrome at each clustering level. However, we can also refine this search to deal with degeneracy. Consider two points \( a \) and \( b \) separated by Manhattan distance \( r \). We see that for two equal \( r \), the \( r \) with a greater \( r^2 \) value will have more degenerate paths of length \( r \) connecting the two points. We can therefore subdivide levels that search a distance \( r \) by considering larger \( r^2 \) values first. This is the method we use in this Chapter. We show the first four levels using this metric in Figure 4.9 (b).

Using this metric, we see in Figure 4.7 (c), that the example syndrome we use in Figure 4.6 is annihilated at the 3-level clustering.

4.2.4. Neutralising a Cluster

Once all the syndromes have attempted to find local clusters within their fixed range at a given level, each cluster is then checked to see if it can be neutralised. This is easily found by summing the values of all the syndromes within a cluster, if the sum of the syndromes is \( 0 \mod d \), then the cluster can be neutralised. We use the assumption that the error will be small and local compared with the cluster. The decoder finds the smallest rectangular box that contains the entire neutralised cluster, and finds a correction operator with support enclosed entirely within the box. The correction operator for the cluster is then found trivially by, for instance, applying the string.
operator that will move every syndrome of the cluster into the bottom-right corner of the box, as shown in Figure 4.10. As the cell is neutral, this operator will necessarily neutralise all the syndromes within the cluster.

We note once more that it does not matter if the correction operator is not the inverse of the error that caused the syndrome, provided EC creates a homologically trivial loop. It is therefore not a problem if the box we have created does not enclose the physical error if the error rate is below threshold, the correction operator we suggest will still with high probability correct the error without affecting the logical subspace of the error correcting code.

4.3. SDRG Decoder

In this section we study the SDRG decoder introduced by Duclos-Cianci and Poulin in [31, 30]. This decoder differs from the HDRG decoder, the SDRG decoder does not directly find a correction path. Instead, for a given syndrome and set of prior probability distribution of local physical errors, priors, it estimates probabilities that the syndrome was generated by an error chain from a particular homology class. In this context, we define the homology class of an error configuration with respect to different measure-
ment outcomes of logical operators with fixed support over the lattice. The calculated probabilities can then be used to produce a correction operator from the appropriate homology class to attempt to return the lattice to its initial state.

In a topological code, it is not important to know precisely what error caused a syndrome, but only the homology class to which the error configuration belongs. A well informed decoder will consider the probabilities of many likely error configurations, instead of seeking the single most likely error configuration. Following this idea, a good strategy for decoding is to evaluate probabilities of many error configurations within a homology class to decide the best correction operator to return.

We note that this decoder has been adapted to tolerate faulty syndrome measurements in reference [32] and to decode the qudit toric code in [33]. The SDRG decoder used here, developed independently of that used by Duclos-Cianci and Poulin in Reference [33], differs in that we optimise the decoder for very high speed decoding at the expense of a reduced threshold. This enables us to probe thresholds up to very large $d$. In this section we broadly review the techniques used in the SDRG decoder. Next, we introduce the specific implementation of the SDRG decoder we use. Finally, we discuss the thresholds obtained by this decoder.

### 4.3.1. Exhaustive Decoding

Although it is not computationally efficient, it will be instructive to first consider exhaustive decoding. Here, we evaluate the probabilities of all the error configurations from each homology class on the lattice using known probabilities from the error model individual physical qudits have sustained errors. Given an error configuration consistent with the syndrome from one particular homology class, we can then find all the other error configurations from that homology class by modifying the configuration by stabilizer deformations. Summing over the probabilities of all of these configurations then will give the exact probability that the syndrome was caused by an error from a particular homology class. This method of decoding is not adopted because it is not efficient with system size. The group of stabilizer deformations on the qudit toric code is generated by $O(L^2)$ operators.

There are therefore an exponential number $O(d^{L^2})$ of error configurations.
to consider within a homology class.

4.3.2. Efficient Decoding which Approximates Exhaustive Decoding

It is not necessary to consider all the error configurations within a homology class. Instead, we can consider probabilities of many ‘sensible’ error configurations which are likely to have occurred. The SDRG decoder uses renormalisation group methods to efficiently consider the probabilities of many sensible error configurations. The SDRG decoder coarse grains syndromes and priors over multiple scales using Bayesian inference methods. We label different length scales with an integer $\lambda$. The decoder coarse grains over $\sim O(\log L)$ levels, until it reaches the final coarse graining level which we label $\lambda_0$. The priors at level $\lambda_0$ correspond to approximate probabilities of the error configuration on the original lattice having come from particular homology classes.

We denote a lattice which contains both syndromes and priors at different scales by $\mathcal{L}(\lambda)$. To efficiently coarse grain $\mathcal{L}(\lambda)$, the SDRG dissects $\mathcal{L}(\lambda)$ into small fixed cells of constant size. Each cell occupies a local connected area of $\mathcal{L}(\lambda)$. Examples of three cells, $\alpha$, $\beta$ and $\gamma$ in green, red and blue respectively in Figure 4.11. This cellular decomposition is then used to coarse grain $\mathcal{L}(\lambda)$ to $\mathcal{L}(\lambda + 1)$, shown on the right of Figure 4.11. Syndromes of the coarse grained lattice $\mathcal{L}(\lambda + 1)$ are evaluated by summing the syndromes of each cell, and the priors of $\mathcal{L}(\lambda + 1)$ correspond to probabilities that the syndrome of the cell is generated by an error chain from different homology classes of the cell. Each cell is decoded exhaustively. As the size of each cell is constant, and small, the time to decode a single cell is constant, and fast. The cells of $\mathcal{L}(\lambda)$ are decoded in $O(L^2)$ time, with the capacity to be parallelised to constant time. After coarse graining to scale $\lambda_0 \sim O(\log L)$, we arrive at $\mathcal{L}(\lambda_0)$ whose syndrome is necessarily vacuum and whose edge priors contain the probabilities that the syndromes were generated by an error configuration from different homology classes.

4.3.3. Decoder Implementation

Coarse graining $\mathcal{L}(\lambda)$ by exhaustively decoding individual cells will only give approximate priors for $\mathcal{L}(\lambda + 1)$, as each cell only has access to restricted
4.3 SDRG Decoder

local information from the local region of the cell of $\mathcal{L}(\lambda)$. In particular, at the boundaries where cells dissect the lattice, the approximation used is very poor. To overcome this, the SDRG decoder employs belief propagation to share information between neighbouring cells before renormalisation takes place. The cells are chosen such that they contain overlapping edges with neighbouring cells, as in Figure 4.11. Before the cells are renormalised, they pass marginal messages to other neighbouring cells. The messages take the form of a probability distribution, and describe the beliefs of a cell of what physical errors may have occurred on edges shared, given its syndrome information. In a similar spirit to exhaustively decoding each small cell, the marginal messages are also evaluated exhaustively over the cell in a constant time. Messages received from nearby cells are used to find better priors when the cells are coarse grained. In general, many messages can be shared between cells, where new messages are generated iteratively using previous messages. Multiple iterations of this step significantly enhance the performance of the SDRG decoder.

In the following sections, we describe how renormalisation step of the decoder works using messages that we assume have already been exchanged. We then explain how the messages are generated and passed, and we finally discuss the performance of this decoder.

The decoder will coarse grain the lattice $\mathcal{L}(\lambda)$, to a lattice of fewer edges $\mathcal{L}(\lambda + 1)$. For the decoder implementation used here, for even $\lambda$ we use cells of
2×1 vertices, and for odd \( \lambda \) we use cells of 1×2 vertices. We describe in detail the coarse graining and belief propagation stages for a 2×1 cell as shown in Fig. 4.12, but the cell decomposition for odd or even \( \lambda \) are equivalent up to a transposition. We note that the cells used here are the smallest possible cells that can be used in such a decoder, which optimise the speed of the algorithm. In choosing this cell size, it is necessary to use different cell shapes at odd and even \( \lambda \). A further detail of the message passing stage in the implementation used here, cells pass messages only to left and right neighbouring cells for even \( \lambda \), and to above and below neighbouring cells for odd \( \lambda \). In a general implementation however, messages can be passed in all directions at all levels. Cells evaluate probabilities of their own homology classes, which become priors on the coarse grained lattice. The decoder uses many cells at every \( \lambda \). However, the action of a single cell of each \( \mathcal{L}(\lambda) \) is identical up to its input. In the following subsection we describe in detail the action of a single cell, and its two nearest neighbours, which is repeated over the entire lattice \( \mathcal{L}(\lambda) \) for all \( \lambda \).

### 4.3.4. Renormalisation Cells

Each cell contains five edges, two syndrome measurements, \( a \) and \( b \), and two messages which arrive from the left, \( l \), and the right, \( r \), which are shown at the left of Fig. 4.12. The homology classes of the cell are defined with respect to their logical operators \( \bar{Z}_1 = Z_1 \) and \( \bar{Z}_2 = Z_2Z_5 \). Each edge, \( e \),
of $\mathcal{L}(\lambda)$ contains a prior vector $p_e = (p_e(I), p_e(X), p_e(X^2), \ldots, p_e(X^{d-1}))$ where $p_e(A)$ is the estimated probability that error $A$ has occurred at $e$. The initial lattice $\mathcal{L}(0)$ contains the original lattice syndrome and takes its priors from the error model the decoder deals with (4.7). Each message, $Q_e = (Q_e(I), Q_e(X), Q_e(X^2), \ldots, Q_e(X^{d-1}))$, encodes beliefs calculated by neighbouring cells which share edges 2 and 3. Each cell will find a syndrome for $\mathcal{L}(\lambda + 1)$, $a \oplus b$, and calculates two prior vectors, $P'_e$, of $\mathcal{L}(\lambda + 1)$. For consistency, we use notation where priors of $\mathcal{L}(\lambda)$ are always labeled with lower case $p$ symbols and prior probabilities of $\mathcal{L}(\lambda + 1)$ are given upper case $P$ probabilities.

A cell will evaluate $P_1'$ and $P_2'$ by considering the probabilities of individual error configurations over its edges for different homology classes of the cell. We describe error configurations, $E$, by the notation $E = (E_1, E_2, E_3, E_4, E_5)$, where the error $X^{E_e}$ has occurred on edge $e$ of the cell. If $E_e = 0$, then no error has occurred on edge $e$. This notation is adopted for brevity, such that we can write the function

$$F[E] = p_1(X^{E_1})Q(I)(X^{E_2})Q_r(X^{E_3})p_4(X^{E_4})p_5(X^{E_5}),$$

which evaluates the probability that an error configuration has occurred using priors $p_e$ and messages $Q_e$. The cell must consider error configurations over all of its homology classes, with respect to $\mathcal{Z}_1$ and $\mathcal{Z}_2$. The homology class is changed by applying logical operators $X_1 = X_1X_4$, and $X_2 = X_5$. We label homology classes, $\mathcal{H}_{H_1, H_2}$ where $H_1, H_2 \in 0, 1, \ldots, d - 1$. If we begin with an error configuration $E$ which is in the trivial homology class $E \in \mathcal{H}_{0,0}$, we change the homology class of the error configuration to one of homology class $\mathcal{H}_{H_1, H_2}$ by adding the error configuration $L(H_1, H_2) = (H_1 \times L_1) \oplus (H_2 \times L_2)$ where $L_1 = (1, 0, 0, 1, 0)$ and $L_2 = (0, 0, 0, 0, 1)$. The trivial homology class is that where error configurations measured by operators $\mathcal{Z}_1$ and $\mathcal{Z}_2$ will return a measurement 0.

We now explain how to evaluate the probability that a cell lies in the trivial homology class, $\mathcal{H}_{0,0}$, $P_1'(I) \times P_2'(I)$. The decoder will adapt this calculation to find the probabilities of all the homology classes. The cell only considers error configurations consistent with the syndrome. The error configuration of the trivial homology class is such that the syndrome of
a is created by an error on edge 4 of Figure 4.12. This corresponds to error configuration $T(a) = (0, 0, 0, a, 0)$. This correction is of the trivial homology class as it does not have a common support with either $\mathbb{Z}_1$ or $\mathbb{Z}_2$. We ignore the syndrome on cell $b$, the total charge of the cell $a \oplus b$ is carried to $L(\lambda + 1)$. To evaluate $P_1(I) \times P_2(I)$, we sum over all error configurations of $H_{0,0}$. We find other paths of the same homology class using the group $S$ of error configurations of trivial paths. The group is generated by error configurations $S_1 = (0, 1, 0, 1, d - 1)$ and $S_2 = (0, 0, 1, d - 1, 0)$. These configurations freely modify the error configurations as they only trivially affect the homology class of the cell. We are then able to evaluate $P_1(I) \times P_2(I) = \sum_S F(T(a) \oplus S)$, where we sum over all group elements $S$.

In general we are then able to find the priors $P_e$ of $L(\lambda + 1)$ by evaluating

$$P_I(X^{H_1}) \times P_2(X^{H_2}) = \sum_S F[L(H_1, H_2) \oplus T(a) \oplus S],$$

for all $H_{1,2}$, where $F[\cdot]$ is the function in Eqn. (4.3).

### 4.3.5. Belief Propagation

To enhance the performance of the decoder, each cell is supplied with marginal messages from neighbouring cells. The messages correspond to the beliefs of a cell that physical errors have occurred on particular edges. The messages are calculated before each level of coarse graining. We label one cell $\beta$, and its left and right neighbours are labeled $\alpha$ and $\gamma$, as shown in Fig. 4.11. Each $\beta$ prepares two messages which are the believed error distributions over the shared edges, 2 and 3 of Fig. 4.12, between neighbouring cells $\alpha$ and $\gamma$ using the syndrome information of the cell. One message $L = (L(I), L(X), L(X^2), \ldots, L(X^{d-1}))$ is passed left to become $Q_l$ of cell $\alpha$ and the other, $R = (R(I), R(X), R(X^2), \ldots, R(X^{d-1}))$, is passed right to become its $Q_l$ of cell $\gamma$. At the same time $\alpha$ and $\gamma$ will respectively prepare messages $Q_l$ and $Q_r$ respectively for cell $\beta$. These messages are then exchanged for later message passing rounds or for use in coarse graining.

At the beginning of any level $\lambda$, all the messages $Q_{l/r}$ are initialised to the uniform distribution. The messages are then evaluated with the formulas
\begin{align}
L(X^a) &= \sum_{L,S_2} G[L \oplus S_1^a \oplus S_2 \oplus T(a)], \quad (4.5) \\
R(X^a) &= \sum_{L,S_1} H[L \oplus S_1 \oplus S_2^a \oplus T(a)], \quad (4.6)
\end{align}

over all $a$, where we sum over all homology classes $L(H_1, H_2)$ and we use the function

\[ G[E] = p_1(X^{E_1})p_2(X^{E_2})Q_l(X^{E_3})p_4(X^{E_4})p_5(X^{E_5}), \]

and

\[ H[E] = p_1(X^{E_1})Q_l(X^{E_2})p_3(X^{E_3})p_4(X^{E_4})p_5(X^{E_5}). \]

Conveniently, the only error configurations that act on edges 2 and 3 are stabilizers $S_1$ and $S_2$, respectively. This is why we are able to change $a$ of $Q_l(X^a)$ ($Q_r(X^a)$) simply by adding the error configuration $S_1^a$ ($S_2^a$).

One may notice from $G$ and $H$ that the messages being passed left (right) do not evaluate new messages using messages received from the left (right). This is to avoid feedback, where messages are created using messages that have previously been sent.

It is easily seen that the computational complexity of evaluating a round of messages is the same as performing one coarse graining step. However, the improvement in threshold by applying belief propagation significantly enhances the threshold of the decoder, so it pays to spend a few rounds evaluating messages [30]. Further, short cuts can be found to evaluate future messages, after the first round of messages have been evaluated by simply performing updates on the previous messages, rather than evaluating new messages from scratch. This significantly speed up evaluation of messages. Typically, between two and four message passing rounds are used. In the implementation used here, we use five rounds of message passing at each stage before performing a renormalisation step. After a few rounds of message passing, messages tend to converge, and they are used to coarse grain the lattice in a renormalisation step.
4.4. The Noise Model and Threshold Estimation

In this section we discuss the noise model the lattice is subjected to, and how we estimate the thresholds of each of the decoders.

As stated previously, we will assume that syndrome measurements are performed without faults. Moreover, both decoders will decode the $X$-type and $Z$-type errors independently. For this reason we restrict the discussion to only $X$-type errors whose syndromes lie on the faces of the lattice. In the noise model every qudit which lies on edge, $e$, suffers an error $A$ with the probability $P(A)$ with

$$P(I) = 1 - p, \quad P(X^j) = p/(d - 1),$$

for all non-zero $j \in \mathbb{Z}_d$ and $0 \leq p \leq 1$ is the physical error rate.

We estimate the error correction threshold numerically via a Monte-Carlo simulation. For a single Monte Carlo sample, we initiate a lattice in a pure eigenstate of $Z_1$ and $Z_2$ in the code space $|\psi\rangle$. An error is generated using the above noise model for some $p$. The syndromes of the error configuration are then measured and fed to the decoder. The decoding algorithm will returns correction chain $C$ which will return $E |\psi\rangle$ to the code space. We then evaluate $\langle \psi | CE |\psi\rangle$. In the event we obtain outcome 1, we consider the decoder successful, otherwise the decoder fails. We repeat this simulation $N = 10^4$ times for a given $p$, and we evaluate the success probability $p_s$ as the fraction of times the decoder succeeds. The standard deviation of the success probability is found with Benoulli’s formula $\sigma = \sqrt{p_s(1 - p_s)/N}$.

To determine the threshold, we plot $p$ versus $p_s$ for different lattice sizes as shown, for example, by Figures 4.13 and 4.14 for the HDRG decoder and SDRG decoder at $d = 5$ and $d = 3$ respectively. The threshold $p_c$ is defined to be the point at which the success probability curves intersect in the limit $L \to \infty$. In other words, the threshold represents the point below which arbitrarily high $p_s$ can be achieved provided that the lattice large enough.

To account for finite system size effects in the SDRG decoder, we estimate $p_c$ by using the fitting proposed by Harrington et. al. [42, 71]. In this fitting, data points are fitted to the hypothesis

$$A + Bx + Cx^2 + FL^{-1/\mu},$$

(4.8)
4.4 The Noise Model and Threshold Estimation

Figure 4.13: Threshold estimation for the HDRG decoder for $d = 5$. Points close to the crossing are used in the hypothesis (4.8), shown in the inset.

Figure 4.14: Threshold estimation for the SDRG decoder for $d = 3$. Points close to the crossing are used in the hypothesis (4.9), shown in the inset.
Figure 4.15: Shows thresholds for different prime $d$ for different decoder types. The SDRG decoder thresholds are shown in blue and the HDRG decoder thresholds are shown in purple. Points a constant factor 0.68 from 4.10 are shown in red. The threshold calculations where these data points are obtained for the HDRG and SDRG decoder are shown in Appendices B and C, respectively. We omit the finite-size effect term of fitting (4.9), as otherwise the recursive parameter fitting fails to converge to suitable values.

where $x = (p - p_c) L^{1/\mu}$, as shown, for example, in the inset in the plot in Figure 4.14. In particular, the last term in the fitting, $FL^{-1/\mu}$, accounts for the the finite system size effects. We can see that, in the limit of $L \to \infty$, this term tends to 0, where $\mu > 0$.

For the HDRG decoder, it becomes necessary to introduce a cubic term to the fitting. This is due to the crossing point falling very low for large $d$, this is an effect of syndrome percolation which we discuss in the analysis, for this reason we use the fitting

$$A + Bx + Cx^2 + Dx^3 + FL^{-1/\mu}, \quad (4.9)$$

In the following Section we evaluate thresholds for all prime $d$ smaller than 20.
4.5. Results and Analysis

We compare the thresholds of the both decoders as \( d \) increases, shown in Figure 4.15. We see that for both decoders, the threshold improves as \( d \) increases. We expect this because as \( d \) increases, we have more syndrome information available. Here we analyse what happens as \( d \) becomes large, and the limitations of both decoders.

We see that, while for low \( d \), both decoders perform comparably well, as \( d \) increases the SDRG decoder begins to outperform the HDRG decoder. This is perhaps not surprising, as the SDRG decoder uses more information to find a suitable correction operator. We consider first the HDRG decoder.

The limitation of the HDRG decoder is that it is constrained by vertex percolation, which we now explain. Consider a square lattice of a certain size. With an identically and independently chosen probability \( P \), each vertex is marked. We say that the lattice percolates if there exists a line can be drawn which spans the lattice. A spanning line begins on one edge of the lattice and finishes at the opposite edge, but can only be drawn on edges whose adjacent vertices are marked. There exists a threshold \( P \), the vertex percolation threshold, above which the probability that the lattice percolates approaches 1 as the lattice grows infinitely large. This threshold \( P = 0.593 \).

Consider once more the HDRG decoder. The decoder is not equipped to deal with clusters whose syndromes percolate. This syndrome percolation is intimately related to vertex percolation. We can give a heuristic argument based on vertex percolation for an upper bound of \( p_c \) for the HDRG in the limit of large \( d \). As mentioned in the previous section, edges suffer some type of error with probability \( p \). In the limit of large \( d \), the probability that a given face has a non-trivial syndrome on a face is \( P \sim 1 - (1 - p)^4 \), as the probability that no errors occur on any of the four edges surrounding a face is \( (1 - p)^4 \). Solving for \( 1 - (1 - p)^4 = 0.593 \) we find an upper bound for the HDRG threshold as \( p \sim 0.2 \). We expect that there is a smaller upper bound than that obtained by these arguments, as the above argument does not correlate different percolating faces. However, the numerics show that the obtained bound is reasonable, as it is seen in Figure 4.15 that the threshold for the HDRG decoder appears to converge close to this bound. We check the limit of \( d \to \infty \) by calculating the threshold of the HDRG decoder for
$d = 7709195422^1$, and find a threshold $p_c \sim 0.178$. We are able to do this because the HDRG decoder does not take additional time to decode lattices of higher $d$. This differs from the SDRG decoder whose run time scales like $O(d^4)$, for this reason we cannot calculate thresholds greater than $d = 19$.

We next consider the SDRG decoder. This decoder efficiently approximates exhaustive decoding. For this reason it is sensible to compare the decoder to optimal decoding. The optimal threshold for a CSS quantum error correcting code of $d$-dimensional physical systems is believed to coincide with the qudit generalisation of the Hashing bound, which is an inequality from quantum Shannon theory which gives a lower bound for the capacity of a quantum channel [60, 68]

$$1/2 > H_d(p),$$

where $H_d(p) = -(1 - p) \log_d(1 - p) - p \log_d(p/(d-1))$ in the case that the number of encoded qudits is constant in the number of physical qudits [33].

It is interesting to see that the thresholds of the SDRG decoder follow quite closely a constant factor $\sim 0.68$ of Equation (4.10). These data points are shown in red in Figure 4.15. The constant factor is a function of the level of approximation used by the decoder which approximates optimal decoding. The results in [33] follow a constant factor $\sim 0.84$ of the optimal threshold.

4.6. Conclusions

In this Chapter we have considered two efficient high-speed decoding algorithms, a hard-decision renormalisation group decoder and a soft-decision renormalisation group decoder. We evaluate the thresholds of both of these decoders as we vary $d$. We see that the improvement of threshold for the HDRG decoder is fundamentally restricted by syndrome percolation in the limit of large $d$. The SDRG decoder considers the probabilities of different error configurations, and therefore approximates the behaviour we expect from an exhaustive decoder. In fact, we see that the decoder follows approximately a constant factor of around 0.68 of the threshold bound we obtain for optimal decoding. The limitation of the SDRG decoder is its increasing

\[^1\text{How's my thesis? UK extension (+44).}\]
complexity in $d$. The complexity is polynomial, and therefore efficient in $d$. However, its analysis is restricted by available computational resources.

Curiously, as $d$ increases, we see that the thresholds of the SDRG decoder begin to tend away from the constant factor of the bound set in Equation (4.10). It is interesting to ask if this is an effect of the degeneracy which the decoder is able to exploit. Certainly, it can be partially explained by small system size effects. Given more computational power, it would be interesting to probe this trend for larger $d$.

Remarkably, for low $d$, the HDRG decoder performs comparably well to the SDRG decoder. In the following Chapter we use the HDRG to probe the performance of quantum memory Hamiltonians with defect lines subject to thermal noise at low $d$. The impressive low $d$ performance, coupled with its versatility make this HDRG decoder the optimal decoder for such a task. Indeed, as mentioned, the decoder was designed for its relatively simplicity, rather than to obtain impressive thresholds. Curiously, we find that the SDRG decoder does not perform as well in the presence of thermal noise. This can be explained by the information the decoder requires to perform error correction; it requires an input of the noise model as well. It is easy to encode an independently distributed noise model. In fact, the optimal strategy for encoding the probabilities of errors in a correlated noise model, such as the thermal noise model we consider in the following Section, is not obvious. We leave this problem as a difficult exercise for the reader.
5. Two-Dimensional Quantum Memories with Line Defects

An important open problem that spans both quantum information and condensed matter physics is how one might realise a realistic thermally stable quantum memory [6, 16, 27, 28, 56, 63, 79]. A quantum memory is a many-body locally interacting quantum system, capable of retaining coherent quantum states for an arbitrarily long time given suitable environmental conditions at a sufficiently large system size. It is known that there exists a thermally stable quantum memory in four dimensions [3], whose coherence time scales exponentially in its system size provided the system is below some critical temperature. This however is not a realistic system as we cannot embed a four-dimensional system locally in three or fewer spatial dimensions.

The models we typically consider for quantum memories are topologically ordered. These models provide a promising avenue of study for the realisation of a stable quantum memory due to their degenerate ground spaces which are locally indistinguishable. This property protects information encoded in the ground state of a topologically ordered system. This is because information encoded in the ground space of a topologically ordered system may only be manipulated by global operations. As environmental noise will only act locally on the system, in general we can reduce the probability that logically encoded information will be corrupted by increasing the size of the system. This will make it very difficult for a local noise model to perform a non-trivial operation over the logical space of the memory. Of course, once we begin to consider the dynamics of a memory in some realistic environment, given a sufficiently long time the probability of a local noise model producing an error which can corrupt a memory will still tend to one. The topic of this Chapter is to consider methods of preventing this decay from occurring for as long as possible in realistic two-dimensional systems.
In three dimensions there have been promising results towards the realisation of a thermally stable quantum memory [16, 26]. While there have not been any results showing thermal stability, partial self correction has been conjectured and numerically verified in the cubic code model [16]. The model described in this Reference shows polynomial coherence time improvements with system size are achieved up to a critical system size, at which point the system size improvements cease. The critical system size, and the degree of the polynomial of the system size improvements, are both dependent on the inverse temperature of the environment. Given this, one can achieve arbitrarily long coherence times by increasing the system size as the temperature of the heat bath is lowered. Other results have also been presented that show that better memories may be achievable by breaking translational invariance in three-dimensional systems [56].

All the models in three and four dimensions that demonstrate robustness against dynamical noise are due to energy barriers which grow macroscopically as an environment attempts to introduce errors into the system. Imposing a high energy cost on large errors to a topologically ordered system will prevent small commonly occurring errors from nucleating and creating large errors whose action has a non-trivial overlap with the logical space of the memory. In the three- and four-dimensional models we have mentioned thus far, energy barriers are invariably imposed by the Hamiltonian terms of the system. Information encoded in the ground space of models with energy barriers will have a coherence time that can scale with the size of the system provided a suitably cold environment.

It is well understood that a topologically ordered system embedded on a two-dimensional manifold, interacting via a local Hamiltonian, cannot support an energy barrier that grows macroscopically with the size of an error [21, 51, 61]. This is due to the string-like logical operators which manipulate the ground space of a two-dimensional model. This necessarily gives rise to a model with string-like errors on the lattice, onto which a local Hamiltonian can only impose a constant energy cost to the end points of the string-like errors. This leads to a system whose coherence time cannot scale with the size of the system in the presence of a thermal bath.

It is well motivated to look for other mechanisms that can prevent the dissipation of noise spreading through a two-dimensional system. Low-dimensional systems are considerably easier to realise in a laboratory as
individual spins are considerably easier to address. In this Chapter we explore the feasibility of proliferating the energy of growing excitations in two-dimensional memories using other elements of the extended system. In particular we use the heat bath to entropically encourage propagating errors to find high energy configurations, thus preventing their dissipation. Specifically we introduce defect lines to the considered lattice models to encourage this behaviour.

Here we consider an experimentally feasible scenario of topologically ordered lattice models with different anyonic excitations whose masses will in general differ \[80\]. Such mass differences are expected to occur in superconducting realisations of such two-dimensional memories \[39, 80\]. The most common errors of such a model in a thermal environment will be low energy excitations. As we will typically expect low energy excitations to corrupt information encoded in the system, we aim to guard the lattice against these types of excitations. We achieve such protection by introducing defect lines to the system which map low energy excitations onto higher energy excitations, and visa versa. This will provide the memory some protection against low energy excitations which attempt to propagate across the lattice, while high energy excitations are still able to travel along the lattice unimpeded. To prevent the propagation of high energy excitations we require the additional condition that they are able to decay into low energy excitations via a fusion channel of the anyon model of the system. With suitable excitation masses we can entropically encourage such an event to occur, thus preventing the rapid propagation of high-mass excitations. There are many lattice models which give rise to Abelian anyons that satisfy this property. The model we choose is Kitaev’s quantum double model of the cyclic group of five elements.

We remark that the memories we discuss here where excitations carry different masses do not perform as well as memories where all the excitations have a uniform high mass. We emphasise that the case we consider is experimentally well motivated with respect to superconducting architectures, and that the defect lines we introduce here will benefit architectures such as these. The key insight we expose are the novel dynamics in the present model, which we hope to be of fundamental interest.

In the considered system we find an interesting regime in parameter space where the introduction of defect lines is particularly effective at suppressing
harmful noise from affecting logical information encoded in the memory. We study this regime extensively. Namely, at low temperatures we expect for the population of high-mass excitations to be strongly suppressed by the gap of the Hamiltonian, and the density of low-mass excitations to be particularly sparse. In this regime the noise that will corrupt a homogeneous lattice with no defect lines will be caused by few low-mass excitations which propagate a very long distance over the lattice. The introduction of defect lines will cause the common low-mass excitations to increase their mass as they propagate across a defect line, and subsequently split with high probability before reaching the next defect line, further impeding their propagation. This approach to the suppression of propagating noise is particularly novel because, alone, the Hamiltonian does not offer a macroscopic energetic penalty for excitations that propagate long distances. However, when coupled with the heat bath, the studied Hamiltonian is capable of entropically encouraging propagating excitations into high energy configurations with high probability. The topic of this Chapter is to study the effectiveness of the entropic suppression of propagating excitations.

Importantly, the parameter regime where we witness the effects we describe does not extend to the zero-temperature limit. We investigate a low temperature regime. This is such that the density of excitations over the lattice is very low, and the encoded quantum information is decohered by a small number of excitations travelling long distances across the system. However, the dynamics we describe, where we use defect lines to encouraging the splitting, will not extend to the zero-temperature limit. This is because excitations require a nominal amount of energy to find high energy configurations. This energy comes from the heat bath. If the heat bath becomes too cool, then excitations are not easily able to find high energy configurations, and instead look for low energy paths to decohere the lattice where proliferation is not a dominant effect.

We begin this Chapter by providing a simple numerical study of a homogeneous lattice model with excitations with imbalanced masses. In particular, we study the rate of creation of particle types of different masses, as well as the scaling of the coherence times of a lattice as we vary the temperature of the heat bath. This provides a good understanding of the behaviour of local two-dimensional Hamiltonian models we consider, and shows that in a low-temperature regime we have the excitation dynamics we have described.
We argue that the introduction of defect lines will suppress the propagation of commonly occurring low-mass excitations, thus protecting quantum information encoded in the ground state of the model. In this Chapter we give an exposition of the new mechanisms we expect defect lines to introduce to excitations diffusing across the lattice. In particular, we expect low-mass excitations crossing the lattice to propagate over defect lines, thus modifying their particle type to one of high mass. We expect this high mass excitation to subsequently decay into pairs of low mass excitations. We go on to show that the new dynamics occur in the considered model by numerically simulating a single excitation as it diffuses across the lattice. We measure the mass and velocity of its evolution to demonstrate that it behaves qualitatively differently to an excitation which occurs on a lattice with no defect lines. In particular we see that an energy penalty frequently occurs, and the consequences are that the expansion of commonly occurring errors dramatically reduced.

Finally, we provide an extended numerical study where we consider the coherence times of the introduced model. We begin by comparing the model to a hypothesis where coherence times grow with the activation energy. We find to a first order approximation that this provides a good description of the behaviour of the presented model. In the regime of temperatures we study, we also identify weak super-exponential coherence time scaling with inverse temperature. We probe the super-exponential temperature scaling by comparing the behaviour of the system to a model where excitations grow logarithmically in mass as they propagate. We find that in the temperature regime that we consider, the super-exponential scaling of the coherence time can be explained using this model, where excitations very weakly proliferate as they propagate, agreeing with the study of a single excitation. We emphasise that, while we present evidence supporting that defects we introduce to the model inspire the dynamics we describe, the proliferating effects we introduce are very weak and do not enhance the model far beyond an activation energy hypothesis. We conclude the Chapter by offering suggestions on how we might look to enhance the entropic effects we present as we approach the zero-temperature limit.

This Chapter is structured as follows, we begin by introducing the qudit toric code model, and we provide a detailed numerical study of its thermal properties. We go on to introduce charge-modifying defect lines which we
5.1. The Qudit Toric Code

We study now the qudit generalisation of the toric code model introduced in Section 3.1, which we use in this Chapter. We note that the model we describe in this Section is the Hamiltonian realisation of the error correcting code used in Chapter 4.

5.1.1. The Hamiltonian

We take a two-dimensional $L \times L$ lattice of green primal faces with periodic boundary conditions, as shown in Figure 5.1. We place $d$-level quantum spins on vertices where the primal faces touch. Dual faces fit in between the primal faces.

The spins of the lattice interact by four-body local projectors $P_v^a$ and $Q_p^a$ around each of the primal faces, $v$, and dual faces, $p$, respectively, where...
5.1 The Qudit Toric Code

\[ a \in 1, 2, \ldots, d - 1. \] These projectors are given by

\[ P_v^a = \frac{1}{d} \sum_{k=1}^{d} e^{2\pi i a k / d} A_v^k, \quad Q_p^a = \frac{1}{d} \sum_{k=1}^{d} e^{2\pi i a k / d} B_p^k, \quad (5.1) \]

where \( A_v = X^\dagger \otimes X \otimes X \otimes X^\dagger \) are the primal and \( B_p = Z \otimes Z \otimes Z^\dagger \otimes Z^\dagger \) the dual operators, as shown in blue and green in Figure 5.1 respectively. The operators \( X \) and \( Z \) are generalised Pauli matrices, where

\[ X|k\rangle = |k + 1(\text{mod } d)\rangle, \quad Z|k\rangle = e^{2\pi i k / d}|k\rangle. \quad (5.2) \]

From (5.2) it follows that

\[ ZX = e^{2\pi i / d} XZ. \quad (5.3) \]

We use the Hamiltonian

\[ H = \sum_v \mathbf{J} \cdot \mathbf{P}_v + \sum_p \mathbf{J} \cdot \mathbf{Q}_p, \quad (5.4) \]

in a condensed notation, where

\[ \mathbf{J} = (J_1, J_2, \ldots, J_{d-1}), \quad (5.5) \]

are interaction strengths, and

\[ \mathbf{P}_v = (P_v^1, P_v^2, \ldots, P_v^{d-1}), \quad \mathbf{Q}_p = (Q_p^1, Q_p^2, \ldots, Q_p^{N-1}). \quad (5.6) \]

The choice of different \( \mathbf{J} \) will become important later in this Chapter where we consider the experimentally motivated case that different excitations carry different masses.

5.1.2. Excitations of the Qudit Toric Code Model

The excitations of the qudit toric code are anyons. There are two different types of excitations, electric anyons and magnetic anyons, labeled \( e_k \) and \( m_k \) respectively. In the qudit generalisation of the toric code model, the excitations carry charges, \( k \), which can take values \( k = 1, \ldots, d - 1 \). The \( e_k \) and \( m_k \) excitations live exclusively on the primal and dual faces, respectively.
5.1 The Qudit Toric Code

Figure 5.2: A green string operator on a qudit toric code which creates an $e_1$ and an $e_{d-1}$ charge at its end points. Also shown in blue is a string operator which creates two $m_1$ charges and an $m_{d-2}$ charge. The $m_1$ charges are spatially separated, this splitting is allowed as the total charge of the lattice is preserved.

Contrary to the qubit toric code, the anyons of the qudit toric code are not their own anti particles. Instead, an $e_k$ or $m_k$ excitation, carrying charge $k$ will be annihilated by the anyon $e_{d-k}$ and $m_{d-k}$ respectively. In these models we therefore draw a single charge $e_1$ and $m_1$ on a diagram as a circle, and an anti charge, an $e_{d-1}$ or $m_{d-1}$ as a diamond symbol, shown in Figure 5.2. Multiple charges or anti charges can be diagrammatically represented by multiple circles or diamonds, respectively.

In general, these anyons have different masses depending on the choice of $J$. The eigenvalues $E = \sum_k (n_{m_k} + n_{e_k})J_k$ of the Hamiltonian increase with the number of anyons $n_a$ of type $a$ that emerge as localised excitations. Examples of string operators that create anyons at their end points are shown in Figure 5.2. String operators in this model for $d > 2$ can split, as shown by the blue string operator shown in Figure 5.2. This is dictated by the fusion rules that preserve the total charge of the all the $e_k$ and $m_k$ anyons on the lattice, modulo $d$ [50].
Figure 5.3.: (a), (b), (c) and (d) show logical operators $Z_1$, $X_1$, $Z_2$ and $X_2$, respectively.
5.1.3. Ground State Degeneracy of the Qudit Toric Code

We finally look at the ground state degeneracy of the qudit toric code model. The model has a \( d^2 \)-fold ground state degeneracy, which consists of two independent qudits associated to different homologically non-trivial cycles around the torus, shown in the previous Chapter in Figure 4.1. and it can be manipulated by the logical operators shown in Figure 5.3 where \( X_1 \) and \( Z_1 \) manipulate one qudit, and \( X_2 \) and \( Z_2 \) manipulate the other qudit in the ground space.

5.2. Simulating the Toric Code Hamiltonian in a Thermal Bath

In this Section we consider the time evolution of the lattice when it is subject to a thermal bath and discuss the methods we use to numerically simulate it.

We simulate the desired dynamics using a rate equation derived from an Ohmic heat bath \([2, 27]\) of the spin-boson noise model obtained from the Davies weak coupling limit. The rate equation takes the form

\[
\gamma(\omega) = \omega/(1 - e^{-\beta\omega}),
\]

where \( \beta = 1/T \) is the inverse temperature of the model and \( \omega \) is the energy change due to the error operation. As this rate satisfies the detailed balance equation

\[
\gamma(-\omega) = e^{-\beta\omega}\gamma(\omega),
\]

it eventually leads to a thermal state \( \rho \sim e^{-\beta H} \). Importantly, this noise model will exponentially penalise the rate of actions with an energy cost with \( \beta \). However, actions that do not incur an energy penalty happen much more rapidly, at a rate \( \sim 1/\beta = T \), which can be seen by taking Equation (5.7) to the limit of \( \omega \to 0 \). We note also that the environment will only act locally, and will only be capable of affecting a single spin at a time.

In the remainder of this Section we describe in detail how we simulate this noise model. We first point out that we are able to simplify the study of the model by simulating only the \( X \)-type errors acting on the lattice.
5.2 Simulating the Toric Code Hamiltonian in a Thermal Bath

One would expect a noise model to introduce both phase noise (Z-type errors) and dit-flip noise (X-type errors) to the lattice. However, as the primal interaction terms of the Hamiltonian of the lattice commute with X-type errors, and the dual interaction terms of the Hamiltonian commute with the Z-type noise. This enables us to consider the two types of noise independently of one another, where excitations will occur only on either the primal or the dual faces. This significantly reduces the computational resources necessary to simulate the dynamical noise we introduce.

Further, phase noise and dit-flip noise act equally on the primal and dual faces, respectively. For this reason we model only $X^a_k$ type errors for $a = 1, 2, \ldots, d - 1$ which generate excitations only on the dual faces. The results we find by considering this noise model will be equivalent to those where we consider only the phase noise on the system. For this reason the simulations we describe here form a good approximation of the entire system.

We now describe the individual steps we follow to simulate thermal noise acting over the the lattice interacting via Hamiltonian (5.4). We consider only $d = 5$, and for each simulation we fix the system size $L$, the coupling coefficients of Hamiltonian (5.4), $J$, the inverse temperature of the system $\beta$, and the amount of time for which we simulate the dynamics of the system $t_{\text{max}}$.

The simulation will continually update dynamical variables as time $t$ progresses. We keep track of the errors that have occurred on the lattice $L$ of $2L^2$ spins, as well as the different excitations on the $L^2$ of the lattice of primal faces $S$. We keep track of the rates at which each possible error $X^a_j$ will occur on each edge $j$ with the terms $\gamma(\omega, X^a_j)$. It is also convenient to record the total rate $R$, which is the sum of all rates $\gamma(\omega, X^a_j)$ for all $j$ and $a$. Each time steps follows the following stages:

1. The rates $\gamma(\omega, X^a_j)$ are evaluated for all $j$ and $a$ using $\beta$, $J$ and $S$ which describes the energy of the system. The total rate is evaluated as

   $$R = \sum_{j,a} \gamma(\omega, X^a_j).$$

2. One of the possible errors $X^a_j$ is chosen, and $L$ is updated accordingly.

   The probability that an error is chosen is weighted by the rate equa-
tions $\gamma(\omega, X^a_j)/R$. Once the error is added to $\mathcal{L}$, $S$ is updated to be consistent with the change to $\mathcal{L}$.

3. The time $t$ is increased by a number chosen randomly from a Poissonian distribution, weighted by the reciprocal of the total rate, $1/R$.

4. If $t$ exceeds $t_{\text{MAX}}$, the noise simulation ends. Otherwise, the described stages are repeated.

Once the noise is generated, we are then able to analyse the noise data by, for instance, measuring the population of excitations that are on the lattice, or by feeding the $S$ to a decoding algorithm, such as those described in the previous Chapter. This enables us to study the rate encoded information decoheres.

5.3. The Behaviour of the Qudit Toric Code Coupled to a Thermal Bath

In this Section we study the behaviour of the homogeneous toric code model interacting via Hamiltonian (5.4) where we have an imbalance in the interaction strengths $J$. In particular we study this system by monitoring how the population of different types of defects vary with temperature at a time long before equilibration occurs, as well as coherence times of information encoded in the ground state of the considered model. In this Section we present extensive numerics of excitation populations which demonstrate that low-mass excitations are generated most commonly in the limit of low temperatures. We use this data to argue that these excitations are those we should be most concerned about in the low temperature regime we examine. We study the system further by considering the coherence times of the qudit toric code in a low temperature regime. This data uncovers a discrepancy between the rate that particles are created, and the rate at which information encoded in the ground state of the qudit toric code achieves a mixed state. We argue that this discrepancy provides evidence that in the low-temperature limit, decoherence of encoded quantum information is caused by a very low density of excitations which have propagated long distances, and that we will be able to dramatically enhance the coherence times of the considered lattice model by confining excitations travelling on
5.3 The Behaviour of the Qudit Toric Code Coupled to a Thermal Bath

Under this lattice model by imposing an energy penalty on their transport. The arguments presented in this Section motivates the introduction of defect lines to minimise the propagation of low-mass excitations.

We begin the study of the thermal dynamics of the qudit toric code by considering the different excitation types we expect that appear on the lattice after a single unit of time. For the temperatures we consider, the population measurements are taken long before the system equilibrates with the thermal bath. The populations of excitations are still growing at the instant we measure the populations, which means creation is a dominant process compared with annihilation. Also the information encoded in the ground state still has a high fidelity with the state encoded initially in the ground state of the system, with respect to the considered decoder.

The presented numerical data shows that in a low temperature regime the most commonly created excitations are those with a small mass cost. We observe with good agreement that the populations of these particles follow a Boltzmann distribution as we vary inverse temperature. We plot the populations of high-mass excitations and the populations of low-mass excitations per unit area after a single unit of time in Figure 5.4. We plot this data on an lattice of modest size, $L = 48$, interacting via Hamiltonian (5.4) for $d = 5$ where $J_1 = J_4 = J_L = 0.38$ and $J_2 = J_3 = J_H = 1$. The Hamiltonian is

Figure 5.4.: Exponential decay of low-mass populations, shown in blue, and high-mass populations, shown in purple, after one unit of time. Low-mass excitations decay like $e^{-0.74\beta+1.4}$ and the high-mass excitation populations decay as $e^{-1.8\beta+2.6}$. 

such that the environment can create pairs of low-mass particle anti-particle excitation pairs, $e_1 (m_1)$ and $e_4 (m_4)$, or high-mass particle anti-particle excitation pairs, $e_2 (m_2)$ and $e_3 (m_3)$. Once particles are created, they are then able to split and fuse, provided their charges are conserved. The choice of $J_L$ seems arbitrary at this stage, but we point out that this choice of $J_L$ will enhance an effect we wish to demonstrate later in the Chapter. We point out also that for the temperatures we consider, a single time unit occurs long before the coherence time of the lattice.

Figure 5.5 shows us that particles are being created at a rate we expect, the gradient of the rate of creation of low-mass excitations as a function of inverse temperature is $\sim -0.74$. This agrees very closely with the gap of the Hamiltonian $\Delta$, and the cost of pair creation of two low-mass excitations, $0.76$. We see also that the rate of creation of high-mass excitations also decays with inverse temperature at a rate $\sim -1.8$. This rate is slightly lower than the energy cost of creating a pair of high-mass excitations. We do not present an explanation for this behaviour, as it can be caused by many microscopic effects which appear in the discussed simulation. We suggest that this discrepancy could be caused by considering the decay of high-mass excitations into pairs of low-mass excitations, which in the case we consider will reduce the lattice to an energetically more-stable configuration.

Importantly in Figure 5.4 we observe that the rate of creation of high-mass
excitation decays much more rapidly than the rate of creation of low-mass excitations, such that in the limit of low temperatures we will expect only low-mass excitations to occur on the lattice. This is highlighted in Figure 5.5, which shows the ratio of the population of high-mass excitations and the population of low-mass excitations. We see this quantity decaying exponentially with inverse temperature. This motivates further the study of introducing defect lines which may inhibit the diffusion of low-mass excitations as these will be the most problematic type of excitations.

We continue the study the qudit toric code model further with the mentioned choice of parameters by calculating coherence times, \( \tau \), of the model on a large lattice of \( L = 72 \). We begin by defining the coherence time. We measure coherence times using the hard-decision renormalisation-group (HDRG) decoding algorithm presented in the previous Chapter. We initialise the lattice in a particular ground state, and then subject it to a thermal environment at some temperature \( \beta \) for a fixed amount of time \( t_{\text{max}} \). Once the noise has been added to the lattice we attempt to recover the ground state of the lattice using the decoding algorithm. We measure the probability of success of recovering the ground state using Monte Carlo sampling evaluated using 40000 samples. We define the coherence time to
be the smallest $t_{\text{max}}$ such that the success rate of recovering the encoded ground state falls below 99%. We modify the HDRG decoder described in the previous Chapter such that it will report failure if the clusters become larger than $L/3$. This minimises finite size effects in later Sections where we consider smaller system sizes. This modification does not affect the threshold of the decoder in the presence of identically independently distributed noise, which we evaluated to be $\sim 13.1\%$, where the calculation is shown in Figure 4.13.

We study the coherence times of the qudit toric code model as a function of inverse temperature. We plot the results in Figure 5.6. The data we present shows that decoherence in the low temperature limit is largely caused by the long range propagation of very few excitations, which we now explain. We begin by applying a linear fit to the logarithmic plot in Figure 5.6, and we obtain a gradient $\sim 0.53$. Were the decoherence determined entirely by the independent creation of excitations we would expect the coherence time to scale with the gap, and the rate of creation of low-mass excitations, $\sim 0.74$, calculated in Figure 5.4. In fact, the gradient is appreciably smaller than this. Moreover, a careful examination of the residual plot shows a slight sub-exponential trend. This behaviour can be explained by considering that the errors sufficient to decay the encoded ground state into a mixed state are generated by excitations with are created and then subsequently propagate across the lattice. The creation of excitations is exponentially suppressed by the Hamiltonian gap $e^{-\Delta \beta}$, as we have already demonstrated. However, once excitations are created, they are free to propagate at a rate $\sim 1/\beta$ which is considerably faster than the rate of creation in the high $\beta$ limit. When considering that excitations can propagate rapidly once they have been created, we should not expect for the coherence times to continue to scale exponentially with the gap of the Hamiltonian, but instead to be affected by sub-exponential terms. The analysis of these sub-exponential terms in the coherence time go beyond the scope of this Thesis, as the primary subject of this Chapter is to study the effects of introducing defect lines to the considered model. Instead, for the analysis we consider in this Chapter in the low temperature and large system-size limit we assume a picture where some density of excitations are quickly generated with a density $\sim e^{-\Delta \beta}$ which rapidly decohere the lattice by diffusing across
5.4 Charge Modifying Defect Lines

In this Section we introduce charge $M$-modifying defect lines to the qudit toric code which modify the charge of an anyonic excitation as they cross, as shown in Figure 5.7. We modify the Hamiltonian interaction terms to realise such defect lines in the following way. We draw defect lines in between the spins of the lattice, with an orientation towards one side of the line. We call this the modified side of the line. Along these lines we modify the $A_v$ and $B_p$ operators as follows: if a defect line lies on a primal face $v$, then we raise the support of $A_v$ on the modified side to the power $M$. We modify the $B_p$ operators on the dual faces $p$ along the defect line in the same way, except a distance

$$R \sim e^{\Delta \beta/2}.$$  \hfill (5.9)

The factor of 2, arrises due to the dimensionality of the model. Assuming this picture where decoherence is largely caused by the transport of excitations, we are motivated to explore the effects of adding defect lines to the considered model to inhibit the propagation of low-mass excitations along the lattice, which is the topic of the remainder of this Chapter.

5.4. Charge Modifying Defect Lines

In this Section we introduce charge $M$-modifying defect lines to the qudit toric code which modify the charge of an anyonic excitation as they cross, as shown in Figure 5.7. We modify the Hamiltonian interaction terms to realise such defect lines in the following way. We draw defect lines in between the spins of the lattice, with an orientation towards one side of the line. We call this the modified side of the line. Along these lines we modify the $A_v$ and $B_p$ operators as follows: if a defect line lies on a primal face $v$, then we raise the support of $A_v$ on the modified side to the power $M$. We modify the $B_p$ operators on the dual faces $p$ along the defect line in the same way, except
the support of the operator which is not on the modified side of the defect line is raised to the power $M$. This is such that they all commute with the modified $A_v$ operators. The modified terms of the $A_v$ and $B_p$ operators are shown in red circles in Figure 5.8. The $M$-modifying defect lines affect the $e_k$ ($m_k$) anyons crossing the line in the negative (positive) direction by multiplying their charge by $M \mod d$, as shown in Figure 5.7, respectively, for $M = 2$. The inverse operation occurs if an anyon crosses in the opposite direction.

In the explicit case that we consider later we will add $M = 2$ charge modifying defect lines to the qudit toric code Hamiltonian (5.4) for $d = 5$. It is easily shown that charge 2-modifying defect lines introduced to this model satisfy condition (2.9) and are therefore transparent. This is checked by by considering all the braiding statistics of all the anyonic excitations of the Hamiltonian, and comparing them to the braiding statistics of the mapped anyons once they cross the defect line.

5.5. Modified Excitation Dynamics in the Presence of Defect Lines

In Section 5.3 we have presented numerical evidence arguing that the propagation of low-mass excitations is an important limiting factor in the coherence time of the two-dimensional lattice model we consider. In this
5.5 Modified Excitation Dynamics in the Presence of Defect Lines

Figure 5.9.: Dynamics of a low-mass excitation on a lattice with defect lines.
Section we consider the new behaviour we expect to observe for propagating low-mass excitations in the qudit toric code model to experience after we introduce defect lines. Specifically, we explicitly describe the dynamical mechanisms we expect to frequently occur that will limit the propagation of excitations. We also consider the limitations of the introduction of line defects, and discuss how to reduce such limitations. For concreteness, we introduce a grid of charge 2-modifying defects lines in a regular grid of separation $\lambda$ to the homogeneous lattice, as shown in Figure 5.10. This choice is such that low mass excitations crossing a defect line will always map onto high mass excitations and visa versa. We remind the reader that we consider Hamiltonian (5.4) for $d = 5$ where $J_1 = J_4 \equiv J_L < J_H/2$ and $J_2 = J_3 \equiv J_H = 1$. This choice will cause high-mass excitations in the considered model to decay into pairs of low mass excitations with high probability via allowed fusion channels.

In the exposition of this Section we need only consider a single excitation on the lattice, as shown in Figure 5.9 (a). The single excitation will attempt to cross from the left-hand side of the page to the right-hand side of the page. Such an evolution will cause significant decoherence to information encoded in the ground state of the lattice.

In order for the excitation to cross the defect line, it must increase in energy of the system from $J_L$ to $J_H$ to become a high-mass excitation of two charges, displayed in Figure 5.9 (b). In order for this to occur, the particle must find energy from the environment, thus impeding its advance. We expect this to be advantageous in protecting the memory from decoherence by low-mass excitations as these defect lines may deflect propagating excitations to follow a less harmful trajectory to the information encoded in the lattice, as shown in Figure 5.9 (c), thus enhancing the coherence time of the system. Indeed, given a newly created pair of excitations confined in a pen, we might expect excitations to find a trajectory that allows them to easily annihilate with their anti-particle pair such that the ground state is not affected.

This is not the full extent to which we expect defect lines to improve the behaviour of the memory. Indeed, low-mass excitations will inevitably escape confining defect lines over a long enough time scale. However, once this occurs there is a fusion channel of the anyon model of the lattice which allows high-mass excitations to decay into pairs of low-mass excitations, as
shown in Figure 5.9(d). Provided $J_L \leq J_H/2$, this will happen with high probability, as pairs of low-mass excitations exist in a more stable state than a single high-mass excitation in the setup we describe. These new low-mass excitations are also confined by defect lines. This means that subsequent propagation of these low-mass excitations will only continue at an additional energy cost, further suppressing the propagation of these excitations. We show such a high energy excitation trajectory in Figure 5.9(e).

We remark that it is not necessary for the energy of a growing error to proliferate in this way. It is possible that a high-mass excitation can propagate between two defect lines without decaying into low-mass excitations, as shown in Figure 5.9(f). Such a mechanism can be suppressed by separating defects by a sufficiently large distance. Given a large separation, the probability that a high-mass excitation walks in between two defect lines without decaying into two low mass excitations will quickly tend to zero.

Excitingly enough, in a low temperature regime, where long range propagation of excitations is common within the coherence time, if we are able to suppress events occurring such as that in Figure 5.9(f) such that propagation by splitting is very common, then we encourage a high energy cost for excitations that propagate long distances. Such behaviour is reminiscent of that shown in three dimensions in Reference [16]. In the remainder of the Chapter, we numerically study the effects of introducing defect lines to a lattice. We argue that in certain regimes, the high energy processes we describe occur appreciably frequently during evolutions of the system.

5.6. A Single Excitation on a Lattice with Defect Lines

Thus far in this Chapter we have considered the homogeneous qudit toric code model, and studied how it behaves in a finite temperature environment. We have observed that in a low temperature regime that the dominant type of excitations will be those of low mass, as the ratio of high-mass excitations and low-mass excitations will decay exponentially with inverse temperature. Moreover, in this low temperature regime we have made the considered assumption that the density of low-mass excitations will be very low at the coherence time of the lattice, and that decoherence is caused dominantly by
Figure 5.10: A single excitation which has propagated over many defect lines. Excitations have readily split upon crossing defect lines. The resulting configuration carries a very large mass.
Figure 5.11.: The time evolution of an initial single pair of excitations subject to a thermal environment. We modify the environment so that it only propagates and fractures excitations, but does not cause pair creation. The average total mass (solid lines) and average spread (dashed lines) of the evolution are depicted without (purple) and with (blue) a defect grid until thermalisation. In the absence of the grid the number of excitations remains constant, while they spread rapidly. In the presence of the grid the propagation of excitations is accompanied by a continuous increase in mass, signifying fractal-like diffusive behaviour. At the same time the excitations spread with a slower velocity. The depicted thermalisation times, $\tau_{\text{no grid}}$ and $\tau_{\text{grid}}$, correspond to the complete thermal evolution of the system. The evolutions are obtained for $\lambda = 2$, $\beta = 8$, $J_L = 0.4$ on an $L = 48$ lattice. We average over one thousand samples.
a small number of sparsely separated excitations which are able to rapidly propagate long distances over the lattice. It is therefore well motivated to study the behaviour of individual excitations occurring on the lattice to achieve a better understanding of the studied system. It is also telling to observe how the behaviour of propagating excitations is modified once we introduce defect lines to the lattice. In this Section we provide such a comparison.

We argued in the previous Section that the dynamics we inspire when we introduce defect lines to the system to low-mass excitations could suppress the propagation of errors diffusing across a lattice. In particular we describe an energy cost which proliferates as excitations diffuse. In this Section we show numerically that individual excitations diffusing across the lattice with defect lines will arrive at high energy configurations before decoherence, such as that depicted in Figure 5.10. We compare this to the lattice model where there are no defect lines introduced, and we see in this same temperature regime that the energy will not proliferate in the same way. We study also the velocity of a diffusing pair of excitations. We observe that this energy cost significantly reduces the rate of diffusion which will ultimately protect encoded quantum information in the physically realistic setting we are interested in.

To identify the diffusive behaviour of the excitations, we numerically simulate the propagation of two low-mass excitations introduced by a single error on the lattice, subject to a low temperature bath. We modify the noise model such that the lattice will never return to the vacuum state, and such that no new excitations can be created far away from the existing excitations. Diffusion and splitting will still occur as normal. This will provide a realistic simulation of the dynamics of an individual excitation. We simulate the propagation until the lattice approaches equilibrium at the coherence time.

We begin the numerical study of individual excitations by measuring how the mass of a pair of diffusing low-mass excitations will change over time. We observe that the diffusion of a pair of excitations on a lattice with a defect grid results in linear increase of the mass of a single excitation pair with time, unlike the no-grid case which remains almost constant. This behaviour is shown in Figure 5.11. This data serves as evidence that low mass excitations must increase to a high mass excitations upon diffusing over a defect line,
which we expect to rapidly decay before reaching the next defect line. Hence, by inserting the defect grid we introduce an increasing energy penalty for an excitation to diffuse across the lattice. As a result excitations propagate significantly more slowly, which we also check numerically.

To numerically study the velocity of excitations diffusing across a lattice, we measure the spread of the excitations over time. We define the spread to be average distance the excitation masses have diffused from the centre of mass of all the excitations on the lattice. The spread of the individual excitations is shown by the dashed lines in Figure 5.11. We see that in both cases the spread of excitations grows linearly with time. This is expected in a model moving with random motion. More significantly, we see that the excitations are slowed by the presence of defect lines. This slowing can be attributed to the energy penalty we have introduced to the propagating excitations by adding defect lines to the lattice. We remark also that both grid and no-grid evolutions thermalise when the spread has reached the same distance of about 1.5. This serves as evidence that decoherence in a low temperature regime is caused by excitations which have propagated long distances. We remind the reader that a spread is the average distance the excitations have traveled away from the centre of mass of all the excitations. This means that the average length of an error is \( \sim 3 \) lattice units, which is larger than the separation of the defect lines, indicating that a pair of excitations which dissipate typically in this parameter regime must cross at least one defect line.

Having observed a qualitative difference in the dynamics of individual excitations on a lattice where we have introduced defect lines, we spend the remainder of this Chapter studying the effects the altered excitation dynamics have on the coherence times of the system.

5.7. Coherence Time Improvements on a Lattice with Defect Lines

In this Section we study how introducing defect lines affects the coherence times of the qudit toric code model with excitations with varying masses. We first study how the coherence time scales with inverse temperature. We first compare coherence time scaling with Arrhenius scaling with the
activation energy $\sim 1$, and we find this model to provide a good fitting, showing that any novel effects our model may exhibit do not substantially enhance coherence times beyond scaling we might expect by increasing the interaction strength of the Hamiltonian coupling terms. However, within the temperature regime we consider, we also witness weak super-exponential inverse temperature scaling. We investigate this scaling using a model where excitations grow logarithmically with energy as they propagate. We extend this investigation by considering how coherence times scale as we increase the size of the system. We find using this model that we have good agreement between the results we present here, and the single excitation study we considered in the previous Section, indicating that the super-exponential scaling within the considered temperature regime serves as evidence of the proliferating dynamics we have described.

We conduct the numerical study in this Section using a lattice where the defect line separation alternates between 1 and 2 along the lattice, as shown in Figure 5.12. This is not the optimal defect line configuration we can use, as there is little separation between defect lines for excitations to split. However, we wish to probe small system sizes, and we have to add four defect lines at a time to keep the ground state degeneracy constant. For this reason we fit four defect lines along six lattice spacings, such that later in this analysis we are able to probe how the memory time improves for many small system sizes. In particular we probe system sizes $L = 12, 18, 24$ and 30. We discard the results of $L = 6$, as this system size is subject to small system size effects.

Having introduced an energy cost for an excitation to propagate, we now consider the memory times for the defect grid model. We numerically simulate a large lattice with $L = 72$ and determine its thermalisation as a function of $\beta$. Figure 5.13 shows that the scaling of the memory time has a super-exponential dependence on the inverse temperature in the interval $\beta = 6$ to $\beta = 9$. This behaviour should be contrasted to the model where we do not introduce defect lines shown in Figure 5.6.

### 5.7.1. Arrhenius Scaling

We begin the study of the coherence time of the model by examining its scaling against inverse temperature. We first consider that the coherence
Figure 5.12: A single $6 \times 6$ lattice unit used to probe coherence times at small system sizes. While the tightly packed defect lines are not the ideal configuration, we must add four defect lines to the lattice at a time to keep the ground state degeneracy constant. Packing the defect lines tightly as in the diagram enables us to probe many small system sizes.
5.7 Coherence Time Improvements on a Lattice with Defect Lines

Figure 5.13.: The coherence time, $\tau$, of the system with defect grid as a function of $\beta$. The errors in the coherence times are conservatively calculated as the earliest time where the error bar of the Monte Carlo sample we use to find the success rate overlaps the 99% fidelity line. Explicitly, the error bar lies at the time where fidelity $\sim 99.05\%$, as we take $N = 40\,000$. We take $L = 72$ and $J_L = 0.38$, while $\lambda$ alternates between 1 and 2 along the lattice in both directions so that a variety of system sizes can be probed. Inset depicts the residual logarithmic plot of $\tau$. The purple points correspond to the linear fitting and the blue to the quadratic fitting. A super-exponential behaviour is obtained with fitting $\tau \sim e^{0.028\beta^2 + 0.54\beta^{-2.5}}$ to be compared with (5.12). We compare this with a linear fitting which shows $\tau \sim e^{0.96\beta - 4.0}$.

time follows an Arrhenius scaling law, and indeed, we find a very good fit. We present the data in Figure 5.13. We see that the coherence time scales $\sim e^{0.96\beta - 4.0}$. We expect a gradient $\sim 1$ as this is the activation energy of the model. Once pairs of low mass excitations are created, at the energy cost of 0.76, the system must then find an additional unit of energy to escape a defect line, thus corrupting information encoded in the ground state of the Hamiltonian. Therefore, we conclude that the dynamics we introduce to the system do not significantly improve the coherence times of the system far beyond that which we expect to a first order approximation. However, we still witness a weak super exponential tendency in the regime of $\beta = 6$.  

113
to $\beta = 9$. We argue that this observation serves as evidence that excitations demonstrate a slight proliferation before the memory decoheres. In the following Subsection we explore this by considering a model where the mass of a propagating excitation grows logarithmically with its expansion.

### 5.7.2. Weak Super-Exponential Inverse Temperature Scaling

We have seen that within the parameter regime we consider here that we do not exceed an inverse-temperature coherence-time scaling whose gradient significantly exceeds Arrhenius behaviour. However, the data we present also demonstrates weak super-exponential scaling. To interpret the weak super-exponential behaviour we entertain the hypothesis that the number of excitations $n$ grows logarithmically with the size of an error diffusing over the lattice, where we denote the size of an error $\xi$. Such a hypothesis is motivated by the study of a single excitation shown in Fig. 5.11, where we presented numerical evidence that the energy of a single excitation on the lattice, and thus the number of excitations $n$, increases as $\xi$ increases. In the presence of thermal fusion and pair creation throughout the lattice we assign to an excitation an energy cost while propagating that scales as

$$\xi = \kappa^n,$$  

where $\kappa$ is some constant [26]. It follows that the energy of an excitation increases as $\Delta \ln \xi / \ln \kappa$. To reach decoherence, the excitations need to travel a distance $\xi \sim R$, so we obtain the condition

$$\tau \sim e^{\Delta n \beta / 2}.$$  

Combining these two expressions we arrive at a coherence time

$$\tau \sim e^{\Delta^2 \beta^2 / (2 \ln \kappa)},$$  

which reproduces the super-exponential behaviour we observe in Fig. 5.13. Hence, the numerical data we have presented provides evidence that the propagation of excitations along the lattice follow the proliferating behaviour determined by $\kappa$.

We now consider the behaviour of the model when the size of the system is varied. We consider low temperatures such that $R \sim L$. In this limit the
5.7 Coherence Time Improvements on a Lattice with Defect Lines

Figure 5.14: The plot of the coherence time as a function of $L$, using the same parameters as in Fig. 5.13. Fitting lines are displayed for $\beta = 7.6, 7.8, \ldots, 9.0$ ordered from bottom to top. Inset depicts the gradients, $G$, of the fitting for different temperatures, giving a polynomial degree that increases linearly with $\beta$. Hence, the coherence time is given by $\tau \sim L^{0.11\beta-0.15}$ to be compared with (5.13).

The lattice must find energy $\Delta \ln L/\ln \kappa$ to decohere, which gives decoherence times

$$\tau \sim L^{\Delta \beta/\ln \kappa}. \tag{5.13}$$

Fig 5.14 shows numerical simulations for the coherence times of small $L$. It is clear that they are growing polynomially with $L$. Moreover, the degree of the polynomial increases linearly with $\beta$. This system size dependence holds up to a critical size, $L^*$, a result that is tightly connected to the super-exponential behaviour. Indeed, the polynomial dependence (5.13) breaks down when the system size is larger than $R$, a condition that gives $L^* \sim e^{\Delta \beta/2}$ [17]. From Fig. 5.14 we see that for the range of $\beta$ accessible to us it is $24 < L^* < 30$.

From Figs. 5.13 and 5.14 we can obtain the effective values of $\Delta$ and $\kappa$ as employed in (5.12) and (5.13) that correspond to our model. A direct fitting gives $\Delta \sim 0.5$ and $\ln \kappa \sim 4$. Hence, the effective energy gap $\Delta$ is similar to the lower mass $J_L$. The large value of the fractal constant $\kappa$ corresponds to sparse generation of excitations during diffusion, due to the probabilistic
nature of the splitting. A lower value of $\kappa$ will lead to more impressive coherence time scalings.

As a final remark, we mention that while in the temperature regime we consider we observe super-exponential inverse temperature scaling and polynomial system-size scaling for small system sizes, we do not expect this behaviour to persist in the zero-temperature limit. Certainly, this is a fundamental difference between the three-dimensional models which demonstrate partial self correction by means of energy barriers and the two-dimensional proposal we present in this Chapter which achieves protection using entropic effects. We observe this difference by considering the following situation in the studied two-dimensional model: we have two low-mass excitations enclosed within a set of defect lines. The two particles arrived due to a single low-mass excitation propagating across a defect line and subsequently decaying into two low-mass excitations, as shown in Figure 5.9 (d). For the energy proliferating effects to persist, we require that one of the two excitations find energy enough from the environment to escape over another defect line. However, at a lower energy cost, the two excitations are able to recombine close to a defect line and escape together. In the limit that $\beta$ goes to infinity, this recombination event will occur much more commonly than proliferation, thus limiting the effects we describe in this Chapter to a particular temperature regime. In the concluding Section we discuss how we might remedy such a problem in future work.

5.8. Conclusions

In this Chapter we consider an experimentally feasible lattice model where different excitations carry different masses. Certain excitations of the model are readily able to split into many-particle configurations via allow fusion channels of the underlying anyon model. We introduce defect lines to the model which modify excitations to those of different masses which are in turn entropically encouraged to split into pairs of excitations of different masses. The effects of introducing defect lines enhances the coherence times of information encoded in the ground state as the probabilistic generation of high energy error configurations significantly reduces the rate of diffusion of errors. We demonstrate the proliferation in the energy cost of diffusing errors as well as the reduced velocity of diffusion by simulating individual
excitations travelling across a lattice. We present further evidence of the behav-
ior we describe by showing that for an extended regime of temperatures the coherence times of encoded information will scale super exponentially with the inverse temperature of the model, and polynomially in the system size of the model, up to some critical system size.

We have demonstrated novel dynamics in a two-dimensional model supported by extensive numerical evidence that could potentially positive results that may help to lead to an efficient quantum memory in two dimensions. No-go results have suggested that we cannot impose an energy barrier using a local Hamiltonian in a two-dimensional lattice model which have led research to consider higher dimensional models. We have shown that in certain temperature regimes, that there exists a model which will entropically achieve high-energy excitation configurations with high probability before decohering. Although the dynamics we demonstrate do not exceed coherence times we expect with the activation energy of a simple Hamiltonian model, we hope the dynamics we have demonstrated will open new exciting directions of research, where we can find alternative mechanisms that proliferate excitations in two-dimensional models in the zero-temperature limit. The first question we may wish to ask is how can we optimise an entropic barrier. In the considered example we have seen that we require a source of entropy, in this case, the source of entropy is the heat bath which causes the noise. Indeed, this means that the entropic protection we achieve will not extend to the low temperature limit, as we require a finite amount of temperature to generate the entropically proliferating excitations we describe. We ask then, are there other sources of entropy we may use to enhance the performance of a two-dimensional memory? For instance it will be interesting to compare how the effects we demonstrate here will compare to a model with randomised Hamiltonian interactions. It may even be possible that further optimisation of parameters of our model will enhance its coherence times at very low temperatures. Certainly it would be exciting to optimise the parameters of the considered model such that we can achieve inverse-temperature coherence-time scaling which cannot be explained by the gap of the Hamiltonian, to learn if it is possible to achieve entropic protection using dynamical effects.
6. Summary and Outlook

The objective of this Thesis is to study defects in topologically ordered lattice models. These objects are of particular interest as they demonstrate rich physics with the potential of achieving universal fault-tolerant quantum computation using models that so promise of experimental accessibility. In particular, we have looked at methods of classifying topological defects, as well as finding exciting new applications of lattice defects. The course of the Thesis has also seen a tangential detour to a discussion of decoding algorithms for the qudit toric code, to develop methods for studying the thermal stability of topologically ordered Hamiltonians with lattice defects. In this Chapter we summarise the results we obtained, and discuss their potential applications.

In Chapter 3 we use topological entanglement entropy calculations, which we know can be used to evaluate data characterising anyonic quasi particles, to probe topological data of twists, point-like topological defects, which is typically associated to anyons. The twists of the considered model admits the results we expect from Ising anyons, extending an analogy drawn in Reference [12]. Importantly, this is the first example of a calculation which shows that topological data of twists can be extracted by measuring the entanglement of their supporting state. This provides an important theoretical tool in the future study topological defects, and the analysis of their computational power. An important extension for the study of twists is a rigorous study of their braiding behaviour [8, 13, 69, 83], as the analogy between twists and anyons breaks down here [9, 12]. Topological quantum computation relies on braiding operations. An additional useful resource in the study of topological defects will be extensions of the tools we present here to entropically analyse the braiding behaviour of twists.

In Chapter 4 we consider two decoding algorithms which use classical information measured from topologically ordered models to attempt to calculate an operation that recovers the encoded information. We compare
two different decoding algorithms for decoding the qudit toric code model. To compare the decoders we evaluate their thresholds. The threshold is the largest rate of noise the lattice can suffer before increasing the size of the lattice will not increase the rate the decoder is able to successfully recover the encoded information. In particular, we study how the threshold is affected as the local quantum dimension $d$ changes. We compare two decoders, the hard-decision renormalisation-group (HDRG) decoder, which uses very simple clustering methods to find the most likely error which the lattice may have suffered, with the soft-decision renormalisation group (SDRG) decoder, which uses more sophisticated Bayesian inference techniques to evaluate probabilities of different error configurations having occurred on the model. We notice that for low $d$, the HDRG decoder performs comparably well to the SDRG decoder, and for $d = 2$ and $3$, it shows more impressive thresholds. This is quite a remarkable observation given the relative simplicity of the HDRG decoder. The thresholds of both decoders improve in $d$ for the considered noise model. We see that as $d$ increases, the SDRG decoder outperforms the HDRG decoder, as the HDRG decoder becomes subject to percolation effects. The SDRG avoids these problems by considering the probabilities of individual errors. This however restricts the SDRG decoder as its complexity has dependence on $d$, which restricts the thresholds we are able to calculate. When designing quantum codes, certain architectures might be more amenable to generating error correcting codes using $d$-level physical systems. The results obtained here provide bounds for error rates and coherence times of such architectures which are constructed from $d$-dimensional systems.

Finally, in Chapter 5 we show that the introduction of a grid of defect lines will introduce novel thermodynamics to two-dimensional lattice models where excitations carry different masses within a particular temperature regime. In the presence of a thermal bath, the defect lines encourage excitations of the model to dynamically achieve high energy configurations with high probability via allowed fusion channels of the anyon model. These dynamics demonstrate weak super-exponential coherence-time scaling with inverse temperature over a range of low temperatures which do not extend to the zero-temperature limit. No-go theorems prove it impossible to develop thermally stable topologically ordered quantum memories. It is therefore well motivated to study the dynamics of individual excitations on the lat-
tice, and to look for entropic methods of suppressing their diffusion. We hope that the results we present here will further motivate the study of entropically protected quantum memories.

The positive results presented in this Thesis show impressive applicability and versatility of topological defects. We have provided new tools to aid in their study, and we have demonstrated novel new applications. Certainly, these ideas show strong promise that the study of topological defects provides a promising avenue of research towards the realisation of a fault-tolerant universal quantum computer. In particular, the results we show in Chapter 5 offer a promising new steps towards the preservation of encoded quantum information in an experimentally-viable laboratory setting.
A. Stabilizers

A certain class of states can be efficiently described using the stabilizer formalism [59]. In this Appendix we review stabilizer formalism. A stabilizer state of \( n \) qubits, \(|\sigma_n\rangle\), can be described uniquely by the operators \( S_j \) that stabilize the state, where the stabilizer state is the common +1 eigenstate of all the stabilizers, such that

\[
S_j |\sigma_n\rangle = |\sigma_n\rangle \quad \forall j,
\]

where \( S_j S_k = S_k S_j \forall j, k \). The stabilizers \( S_j \) form an Abelian group, the stabilizer group, \( S \). The stabilizer group can be generated by \( n \) independent operators in the generating set, \( \tilde{S} \), such that

\[
\tilde{S} = (S_1, S_2, \ldots, S_n).
\]

The generating set is independent provided \( S_j S_k \neq S_l \forall j, k, l \in \tilde{S} \). The generating set is not unique, we can replace an element of the generating set, \( S_j \) with any other element of the stabilizer group, \( S'_j \) provided the generating set still generates the same group. This will always be true provided we choose \( S'_j \) such that \( S'_j = S_j \prod_{k \neq j} S_k^{n_k} \) where \( n_k \in \{0, 1\} \) and we take the product over all the generators of the generating set \( k = 1, \ldots, n \) but not including \( j \).

The stabilizer group \( S \) is a subgroup of the Pauli group \( \mathcal{P}_n \), where

\[
\mathcal{P}_n = \{ p \in \mathcal{P}_n | p = p_1 \otimes p_2 \otimes \ldots \otimes p_n, p_j \in \mathcal{P}_1 \},
\]

where \( \mathcal{P}_1 = \{ \pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ \} \) is the Pauli group for a single qubit and \( I, X, Y \) and \( Z \) are the standard Pauli matrices.
B. HDRG Decoder Threshold Curves

This Appendix shows the data used to find the threshold crossing points for the HDRG decoder. All thresholds use system sizes $L = 64, 96$ and $128$ with a cubic fitting hypothesis. The finite system size term of Equation (4.9) is omitted from the fitting in the $d = 11$ and $d = 19$ curves, we were unable to converge on a fitting using the parameter in these cases.
Figure B.1.: Qudit toric code threshold calculation for $d = 2$. Fitting gives threshold $0.0830 \pm 0.0008$.

Figure B.2.: Qudit toric code threshold calculation for $d = 3$. Fitting gives threshold $0.1114 \pm 0.001$. 
Figure B.3.: Qudit toric code threshold calculation for $d = 5$. Fitting gives threshold $0.1312 \pm 0.0005$.

Figure B.4.: Qudit toric code threshold calculation for $d = 7$. Fitting gives threshold $0.1419 \pm 0.0005$. 
Figure B.5.: Qudit toric code threshold calculation for $d = 11$. Fitting gives threshold $0.1488 \pm 0.0006$. We omit the finite-size effect term of Equation (4.9) by setting $F = 0$, as the recursive fitting method to converge to a suitable fitting.

Figure B.6.: Qudit toric code threshold calculation for $d = 13$. Fitting gives threshold $0.1504 \pm 0.0005$. 
Figure B.7.: Qudit toric code threshold calculation for $d = 17$. Fitting gives threshold $0.1531 \pm 0.0006$.

Figure B.8.: Qudit toric code threshold calculation for $d = 19$. Fitting gives threshold $0.1537 \pm 0.0008$. We omit the finite-size effect term of Equation (4.9) by setting $F = 0$, as the recursive fitting method to converge to a suitable fitting.
This Appendix shows graphically the data used to find the thresholds for varying \( d \) using the SDRG decoder. The system sizes decrease with \( d \), to compensate for the increasing computational work required to deal with higher \( d \) for this particular decoder. Using the SDRG decoder it is suitable to use a quadratic fit for all the threshold calculations with a small system size parameter.
Figure C.1.: Qudit toric code threshold calculation for $d = 2$. Fitting gives threshold $0.07559 \pm 0.0001$ for system sizes $L = 512, 1024$ and $2048$.

Figure C.2.: Qudit toric code threshold calculation for $d = 3$. Fitting gives threshold $0.1091 \pm 0.0002$ for system sizes $L = 512, 1024$ and $2048$. 
Figure C.3.: Qudit toric code threshold calculation for $d = 5$. Fitting gives threshold $0.14383 \pm 0.00006$ for system sizes $L = 256, 512$ and 1024.

Figure C.4.: Qudit toric code threshold calculation for $d = 7$. Fitting gives threshold $0.1636 \pm 0.0004$ for system sizes $L = 128, 256$ and 512.
Figure C.5.: Qudit toric code threshold calculation for $d = 11$. Fitting gives threshold $0.1860 \pm 0.0005$ for system sizes $L = 64, 128$ and 256.

Figure C.6.: Qudit toric code threshold calculation for $d = 13$. Fitting gives threshold $0.1943 \pm 0.0002$ for system sizes $L = 32, 64$ and 128.
Figure C.7.: Qudit toric code threshold calculation for $d = 17$. Fitting gives threshold $0.2059 \pm 0.0008$ for system sizes $L = 16, 32$ and $64$.

Figure C.8.: Qudit toric code threshold calculation for $d = 19$. Fitting gives threshold $0.2103 \pm 0.0008$ for system sizes $L = 16, 32$ and $64$. 

131
Bibliography


[70] Daniel C. Tsui, Horst L. Störmer, and Arthur C. Gossard. Two-
dimensional magnetotransport in the extreme quantum limit. Phys.

[71] Chenyang Wang, Jim Harrington, and John Preskill. Confinement-
higgs transition in a disordered gauge theory and the accuracy thresh-

[72] Xiao-Gang Wen. Vacuum degeneracy of chiral spin states in compact-


[74] Xiao-Gang Wen. Topological orders and edge excitations in fractional


[76] Xiao-Gang Wen and Qian Niu. Ground-state degeneracy of the frac-
tional quantum Hall states in the presence of a random potential and

[77] Edward Witten. Quantum field theory and the Jones polynomial. Com-

[78] Michael M. Wolf, Frank Verstraete, Matthew B. Hastings, and J. Ig-
nacio Cirac. Area laws in quantum systems: Mutual information and

[79] James R. Wootton. Novel topological phases and self-correcting mem-

[80] James R. Wootton, Ville Lahtinen, Benoit Douçot, and Jiannis K. Pa-
chos. Engineering complex topological memories from simple Abelian

[81] James R. Wootton, Ville Lahtinen, Zhenghan Wang, and Jiannis K.
78:161102(R), 2008.
