Multi-path Summation for Decoding 2D Topological Codes

Ben Criger[1 2] and Imran Ashraf[1]

[1]QuTech, TU Delft

[2]Institute for Globally Distributed Open Research and Education (IGDORE)

October 16, 2018

Fault tolerance is a prerequisite for scalable quantum computing. Architectures based on 2D topological codes are effective for near-term implementations of fault tolerance. To obtain high performance with these architectures, we require a decoder which can adapt to the wide variety of error models present in experiments. The typical approach to the problem of decoding the surface code is to reduce it to minimum-weight perfect matching in a way that provides a suboptimal threshold error rate, and is specialized to correct a specific error model. Recently, optimal threshold error rates for a variety of error models have been obtained by methods which do not use minimum-weight perfect matching, showing that such thresholds can be achieved in polynomial time. It is an open question whether these results can also be achieved by minimum-weight perfect matching. In this work, we use belief propagation and a novel algorithm for producing edge weights to increase the utility of minimum-weight perfect matching for decoding surface codes. This allows us to correct depolarizing errors using the rotated surface code, obtaining a threshold of $17.76 \pm 0.02\%$. This is larger than the threshold achieved by previous matching-based decoders ($14.88 \pm 0.02\%$), though still below the known upper bound of $\sim 18.9\%$.

1 Introduction

Quantum information processing (QIP) provides a means of implementing certain algorithms with far lower memory and processing requirements than is possible in classical computing [1, 2]. The analog nature of quantum operations suggests that devices which perform QIP will have to function correctly in the presence of small deviations between ideal operations and those that are actually implemented, a property called *fault tolerance* [3, 4]. In addition, many external sources of error exist in current QIP devices, making fault tolerance a practical necessity as well as a theoretical one.

In order to design fault-tolerant QIP protocols, we first require a set of quantum states for which the effect of small deviations can be reversed, called a *quantum error-correcting code* [5], analogous to *classical* error-correcting codes [6]. One important difference between classical and quantum error correction is that quantum states are inherently continuous, whereas classical states are discrete. This suggests that noise processes on quantum computers will

Ben Criger: bcriger@gmail.com

also be continuous, and therefore difficult to correct. However, the action of a projective measurement can effectively discretise the noise [7, Chapter 2], bringing the study of quantum error correction closer to classical coding theory. Unfortunately, the action of a projective measurement also ensures that the qubits used in a quantum code cannot be measured directly in order to determine where errors have occurred, since such a measurement would destroy any superposition of code states, discretising the logical state as well as the noise. However, a large set of quantum error-correcting codes, called *stabiliser codes* [3], allow the measurement of multi-qubit operators which yield results depending only on which error has occurred, and not on the code state itself. These operators are the *stabilisers* of the code, and the result from their measurement is called the *syndrome*. This syndrome indicates which stabilisers anticommute with the error.

Typically, the stabilisers are taken to be *Pauli operators* on n qubits, tensor products of the operators $\hat{\mathbb{1}}$, X, Y, and Z, where

$$\hat{\mathbb{1}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \text{ and } Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{1}$$

Conveniently, noise discretisation implies that if a code can be used to correct the random application of these operators on a subset of the qubits, it can also correct arbitrary operators on the same subset, since the Pauli operators form a basis for the space of operators.

When measuring the stabilisers, it is possible for the measurement apparatus to return an incorrect eigenvalue, or for the measurement procedure to produce errors on the qubits. In order to obtain fault tolerance in such a scenario, it is necessary to repeat the stabiliser measurements many times. In this work, we restrict ourselves to a simpler model, in which Pauli errors occur at random on data qubits, and the measured eigenvalues are correct, and obtained without consequence.

In order to correct these Pauli errors, it is necessary to derive an error which reproduces the given syndrome, and is unlikely to alter the intended logical state if performed on the system. This process, called *decoding*, is non-trivial, since a code with n physical qubits will typically contain $\mathcal{O}(n)$ stabilisers, so the number of possible syndromes scales as 2^n, prohibiting the use of a simple lookup table. Also, many quantum codes are *degenerate*, having multiple errors corresponding to a given syndrome, further complicating the process of decoding.

The purpose of this work is to improve the accuracy of a frequently-used method for decoding 2D *homological codes* [8]. The prototype for these codes is the 2D *surface code*, which we review in the following section, along with the reduction of the decoding problem to minimum-weight perfect matching. In Section 3, we review prior enhancements to the decoding algorithm of interest, as well as alternative algorithms. In Sections 4 and 5, we introduce the methods we propose to enhance the accuracy of surface code decoding and obtain increased threshold error probabilities. We discuss the effect of changing boundary conditions, as well as the additional complexity of decoding using these methods, and conclude in Section 6.

2 The Surface Code

A surface code [9–11] is supported on a square array of qubits, with stabilisers defined on individual square tiles, see Figure 1. Specifically, we focus on the *rotated* surface code, with the boundary conditions from Figure 1. Earlier constructions involve tiling a surface with non-trivial topology, such as a torus [8], or using an alternate open boundary condition with *smooth* and *rough* boundaries [10]. We choose to study the rotated code, since it requires the fewest physical qubits per logical qubit.

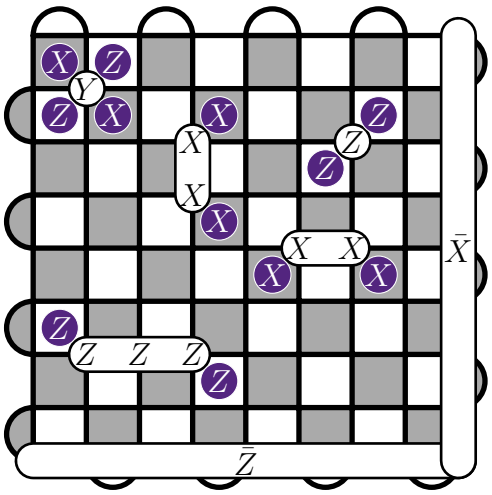

Figure 1: Surface code with distance $d = 9$. Data qubits are placed at the vertices of a d-by-d square tiling. Logical operators can be placed along the sides of the square. Stabilisers of the form $ZZ/ZZZZ$ are supported on grey tiles, and stabilisers of the form $XX/XXXX$ are supported on white tiles. An X or Z error will anticommute with a stabiliser if the stabiliser is at an endpoint of a path of Pauli operators which is part of the error. A Y error is detected by both types of stabilisers.

In any of these codes, pairs of stabilisers anticommute with chains of errors that connect the two tiles supporting the stabilisers, producing a set of locations where the syndrome is 1 (which we will call *vertices* for the remainder of this work). This mathematical structure reduces the decoding problem to that of finding a likely set of connecting errors which join the vertices in pairs (we will call these connecting errors *paths* for the remainder of this work, and refer to abstract connections between vertices as *edges*). To maximise the likelihood, we follow the derivation of [12, Section IV D]. We begin with the assumption that only one type of error (X or Z) can occur, for simplicity. The likelihood of an error E is then a function of its weight $w(E)$ (the size of its support), the total number of qubits n, and the underlying probability of error p:

$$p(E) = p^{w(E)}(1-p)^{n-w(E)} = (1-p)^n \left(\frac{p}{1-p}\right)^{w(E)} \tag{2}$$

We assume that, if two vertices are connected by a path, the length of that path is minimal. This implies that the weight of the corresponding edge is given by the rotated Manhattan distance between the vertices it connects, see Figure 2. Note that, due to the open boundary

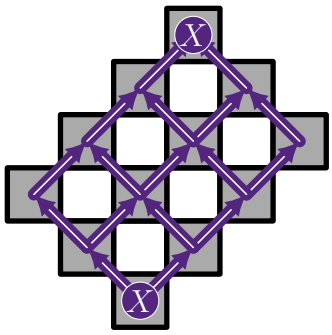

Figure 2: A pair of vertices separated by an edge whose minimum length is 5. A minimum-length path is composed of steps between neighbouring tiles, and any path which is consistent with the indicated direction will be minimum-length.

conditions, a single vertex can be connected to the boundary instead of another vertex. This is typically remedied by introducing *virtual vertices* which are connected to each other with weight-0 edges, and to a designated vertex with a weight equal to the minimum path length between that vertex and the nearest boundary [13].

The total weight of an error is then the sum of the weights of the edges, so the likelihood can be expressed as a product over the edge set:

$$p(E) = (1-p)^n \prod_{e \in \text{edges}(E)} \left(\frac{p}{1-p} \right)^{|e|} \tag{3}$$

To maximise the likelihood, it suffices to maximise any function proportional to the likelihood, so we can use a simplified cost function:

$$p(E) \propto \prod_{e \in \text{edges}(E)} \left(\frac{p}{1-p} \right)^{|e|} \tag{4}$$

$$\sim \sum_{e \in \text{edges}(E)} |e| \ln \left(\frac{p}{1-p} \right) \tag{5}$$

$$\propto - \sum_{e \in \text{edges}(E)} |e| \tag{6}$$

with the minus sign present since the *odds* of a physical error $p/1-p \ll 1$. This implies that, in order to maximise the likelihood of a prospective error, we should find a set of edges that connects the vertices in pairs (a *perfect matching* in graph theory) with minimal weight. Finding minimal-weight perfect matchings is a well-known combinatorial problem, solved using the *Blossom* algorithm [14, 15].

Decoding with this algorithm results in a threshold error rate of $\sim 10.3\%$ [16], compared with a threshold rate of $\sim 10.9\%$ estimated using a correspondence between the toric code decoding threshold and the phase transition in a spin model [17, 18]. From this, we surmise that minimum-weight perfect matching (MWPM) decoding performs well, though not optimally, in the event that errors of a single type are distributed with identical probability on each qubit. In addition, there remain many scenarios in which the independent identically-distributed (IID) error model does not apply. In the following section, we summarise the current efforts to enhance MWPM decoding with both IID and more realistic models in mind.

3 Prior Work

The derivation of the objective function in Equation (6) assumed that X and Z errors occur independently, with identical probability on each qubit. This implies that, if the probability of an X or Z error is $O(p)$, the probability of a Y error is $O\left(p^2\right)$, since $Y \sim XZ$. If the probabilities of X, Y, and Z errors differ from this, the performance of the decoder will be decreased. Consider the frequently used example of *depolarizing* noise, in which each type of error is applied with equal probability. A decoder which minimises the weight of X and Z errors separately may fail to minimise the relevant weight, resulting in a logical error, see Figure 3.

To minimise the appropriate weight function, we need to avoid 'double-counting' the weight of Y errors. To accomplish this, Delfosse [19] and Fowler [20] reduce the weight of X edges that cross Z edges, since individual Y errors result in crossed X and Z edges (see Figure 1, upper left). In order to determine where these crossings occur, the authors first calculate one

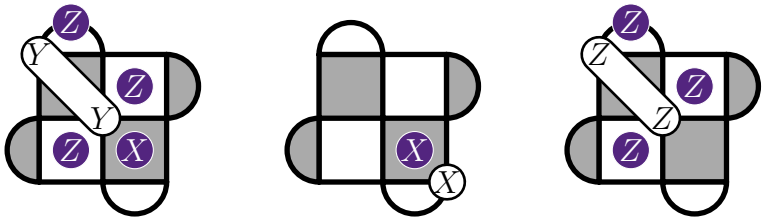

Figure 3: From left to right: a weight-2 Y error acting on a distance-3 surface code, and the independent minimum-weight X and Z corrections. The appropriate correction, $XX \cdot ZZ$, is not minimum-weight in the independent error model.

of the matchings, X or Z, then use the resulting edge set to determine edge costs for the other matching. In this way, Delfosse demonstrates that the threshold error rate against Z errors can be raised for a homological code which has a naturally larger tolerance to X errors.

One obstacle for this approach is that X and Z paths can intersect more than once, see Figure 4. It is not clear during decoding which error corresponding to a given edge should be

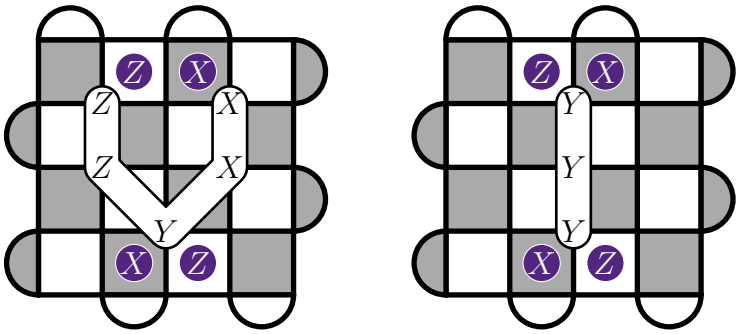

Figure 4: For a continuous chain of Y errors, different minimum-weight X and Z paths can be found with different crossing numbers for the same syndrome.

selected in order to maximise the crossing number, thus minimising the total weight. Furthermore, there can be more than one minimum-weight perfect matching corresponding to a vertex set, further hampering any effort to obtain an accurate correction to the depolarizing channel using MWPM decoding. Accounting for degeneracy caused by edge and matching multiplicity, it appears, will provide insight in this direction.

There have been multiple efforts to address these types of degeneracy. The first is due to Barrett, Stace and Doherty [21–23], and uses the IID error model. They derive a modified objective function, accounting for the fact that there are multiple error configurations with the same weight corresponding to a given matching, see Figure 5. If this number of minimum-weight configurations corresponding to the error E is labeled $\Omega(E)$, then the derivation beginning with Equation (3) instead begins with the likelihood function

$$\Omega(E)(1-p)^n \left(\frac{p}{1-p}\right)^{w(E)}. \tag{7}$$

In order to use an MWPM-based decoder, it is necessary to express the likelihood function as a product over the set of edges. This requires us to express $\Omega(E)$ as a product over the set of edges. This is possible, since the number of paths joining a pair of vertices depends only on the positions of those vertices, and not on any property of the graph not related to the edge under consideration. See, for example, Figure 5, in which the total number of weight-6 errors (16) is

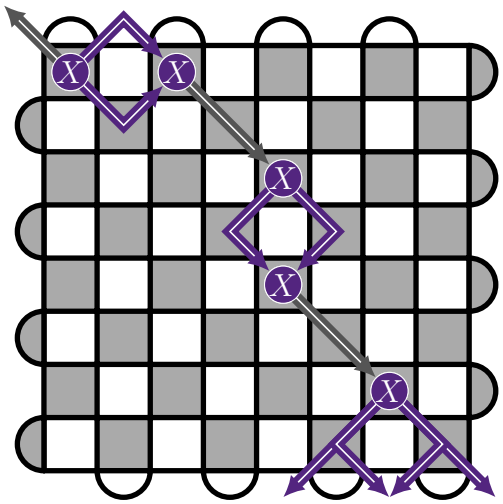

Figure 5: Syndrome set with a single minimum-weight matching (weight 5) and 16 matchings with a single unit of excess weight. If the odds of an error $p/1-p > 1/16$, then the higher-weight matching is more likely, see Equation (7).

a product of the number of paths joining the paired vertices (2, 2, and 4). The probability of a minimum-weight error equivalent to E can then be factored as before:

$$p(E) \propto \prod_{e \in \text{edges}(E)} \Omega(e) \left(\frac{p}{1-p}\right)^{|e|} \tag{8}$$

Taking the logarithm results in the final form of the objective function:

$$p(E) \propto -\sum_{e \in \text{edges}(E)} |e| - \frac{\ln \Omega(e)}{\ln {}^{1-p}/_{p}}. \tag{9}$$

In order to calculate $\Omega(e)$ for an edge between two real vertices, Barrett/Stace/Doherty note that there are $\binom{\Delta_y+\Delta_x}{\Delta_x}$ paths between points on a rotated square tiling of the torus separated by (Δ_x, Δ_y) (see also [24]).

Using this modification to the edge weights, MWPM decoding results in a threshold near 10.3% when correcting IID X/Z noise. This is similar to the increase in the threshold that can be obtained from solving an associated statistical physics model using matchings obtained from Metropolis sampling [25], which suggests a connection between the two methods. In addition, efforts have been made to improve the threshold of MWPM decoders by constraining the input graph to support errors from a specific coset, and by modifying the obtained solutions through Markov Chain Monte Carlo [26]. However, the thresholds from modified MWPM decoding are still suboptimal, suggesting that further improvements are required.

In a recent departure from MWPM-based decoding, Bravyi, et al. [27] have introduced polynomial-time maximum-likelihood decoders for surface codes which are based on efficient sub-theories of quantum mechanics. For IID X/Z noise, they begin by matching each vertex to an arbitrary boundary, producing a correction which eliminates the syndrome, but results in a logical error with high likelihood (also known as a *pure error* [28]). They then provide a reduction from the problem of determining the probability that the pure error is an accurate correction to the calculation of the output of a *matchgate circuit* [29, 30]. For arbitrary stochastic Pauli noise, likelihood estimation reduces instead to the problem of contracting a tensor network state [31], which can also be solved in polynomial time, reminiscent of earlier

decoders for topological codes which use the Renormalisation Group [32, 33]. Using these techniques, Bravyi et al. are able to obtain optimal thresholds against IID X/Z and depolarizing noise, opening a gap between MWPM and maximum-likelihood decoding.

This gap can be narrowed by introducing a generalised edge weight calculation, similar to the method of Barrett/Stace/Doherty, adapted to the case in which there are different probabilities of error on each qubit. We do this in the following section.

4 Odds Calculation

In order to more accurately approximate the odds of a path existing between two vertices, we first generalise the calculation of Barrett/Stace/Doherty to cases where a vertex is being matched to the boundary, or to where some paths cannot be realised, see Figure 6. To count the

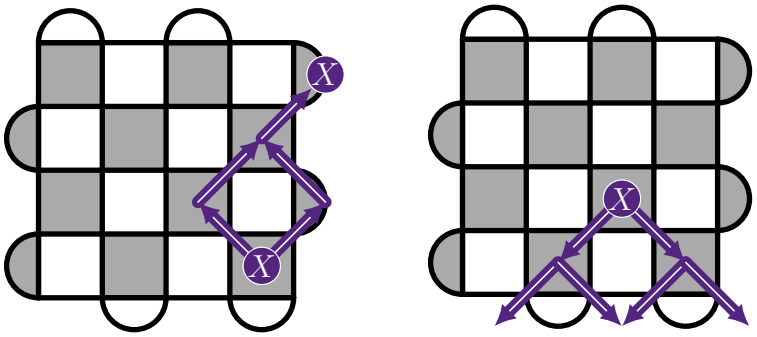

Figure 6: Two cases in which the number of paths is not $\binom{\Delta_y+\Delta_x}{\Delta_x}$, due to open boundary conditions. Left: one otherwise-possible path between two syndromes cannot be realised since two of the qubits on the path are outside the lattice. Right (also see Figure 5): Paths from a syndrome to the boundary can exit the lattice at multiple points, further preventing a simple calculation of Ω_e.

number of paths on *directed acyclic graphs* such as these, we can use the following function:

function NUM_PATHS(g, v_i, F) ▷ g directed acyclic graph, v_i initial vertex, F final vertex set;
    for all $v \in$ `vertices`(g) **do**
        `count`$[v] \leftarrow 0$
    end for
    `count`$[v_i] \leftarrow 1$
    while `true` **do**
        if `all`(`count`$[f] \neq 0$ `for` $f \in F$) **then return** `sum`(`count`$[f]$ `for` $f \in F$)
        end if
        for all $v \in$ `vertices`(g) **do**
            if `all`(`count`$[i] \neq 0$ `for` $i \in$ `incoming`(v)) **then**
                `count`$[v] \leftarrow$ `sum`(`count`$[i]$ `for` $i \in$ `incoming`(v))
            end if
        end for
    end while
end function

If this function is applied to a closed bounding box (see Figure 2), the result will be $\binom{\Delta_y+\Delta_x}{\Delta_x}$, which simplifies the calculation for periodic boundary conditions, for which all bounding boxes

are closed. For a generic directed acyclic graph, this function requires one operation for every edge of the input graph, and can be parallelized using *message passing* [34, Chapter 16], reducing the runtime to be proportional to the path length using one constant-size processor per vertex of the input graph. To be implemented serially, the input graph g should first be *topologically sorted* [35, Section 22.4], also requiring a number of operations at most linear in the graph's size. The key to the derivation of the path-counting algorithm is that the number of paths reaching a vertex w is the sum of the number of paths reaching the vertices with edges incoming to w. Using this function to calculate Ω_e, we can replicate the result of Barrett/Stace/Doherty using the rotated surface code, see Figure 7.

To generalise to the case where Y errors can occur with arbitrary probability, we first consider the role of detected Z errors in determining where X errors have occurred (see [19] for an in-depth discussion). We assume that the physical error model has a fixed probability of X, Y, or Z on each qubit, which is qubit-independent. To see the role that detected Z errors play, we imagine that we could determine with certainty the positions of errors that anticommute with the Z-detecting stabilisers. These errors must be either Z or Y, and the qubits which have not been affected by one of these errors must have been affected by an X error, if at all. To use this information in determining the positions of X errors, we use the *conditional* probabilities of an X error occurring, depending on the result of the Z decoding:

$$p\left(X \text{ or } Y | Y \text{ or } Z\right) = \frac{p_Y}{p_Y + p_Z}, \qquad p\left(X \text{ or } Y \middle| \hat{\mathbb{1}} \text{ or } X\right) = \frac{p_X}{p_X + p_{\hat{\mathbb{1}}}} \tag{10}$$

Thus, we see that attempting to account for X/Z correlations in the error model will require us to consider error models in which there are different probabilities of error on each qubit, even when the underlying error model is qubit-independent.

The method we use to approximate conditional probabilities of error in the absence of perfect detection is explained in the following section. For now, we generalise the derivation of the likelihood of an error beginning in Equation (3), with a few approximations and definitions:

- We only consider minimum-length paths, which are restricted to the set of qubits within bounding boxes of the type seen in Figures 2 and 6.
- We consider two possibilities, either the vertices being considered are joined by a single path, or none of the qubits in the bounding box are in error.

The probability $p(E)$ (now referring to the probability of any error equivalent to E under a change of minimum-length path) can then be expressed in terms of the odds of error on a given qubit, $o_q \equiv \frac{p_q}{1-p_q}$:

$$p(E) = \prod_q (1 - p_q) \times \prod_{e \in \text{edges}(E)} \sum_{p \in \text{paths}(e)} \prod_{q \in p} o_q \tag{11}$$

$$\sim \sum_{e \in \text{edges}(E)} \log \left(\sum_{p \in \text{paths}(e)} \prod_{q \in p} o_q \right) \tag{12}$$

We can divide the set paths(e) into disjoint subsets that pass through each of the vertices adjacent to the final vertex, assuming that there is only one (otherwise, we can divide the set paths(e) into disjoint subsets associated with individual final vertices, then proceed). The path from each of these *predecessor* vertices to the final vertex only traverses a single qubit q_p, so:

$$\sum_{p \in \text{paths}(e)} \prod_{q \in p} o_q = \sum_{q_p} o_{q_p} \sum_{p' \in \text{paths}(q_p)} \prod_{q \in p'} o_q, \tag{13}$$

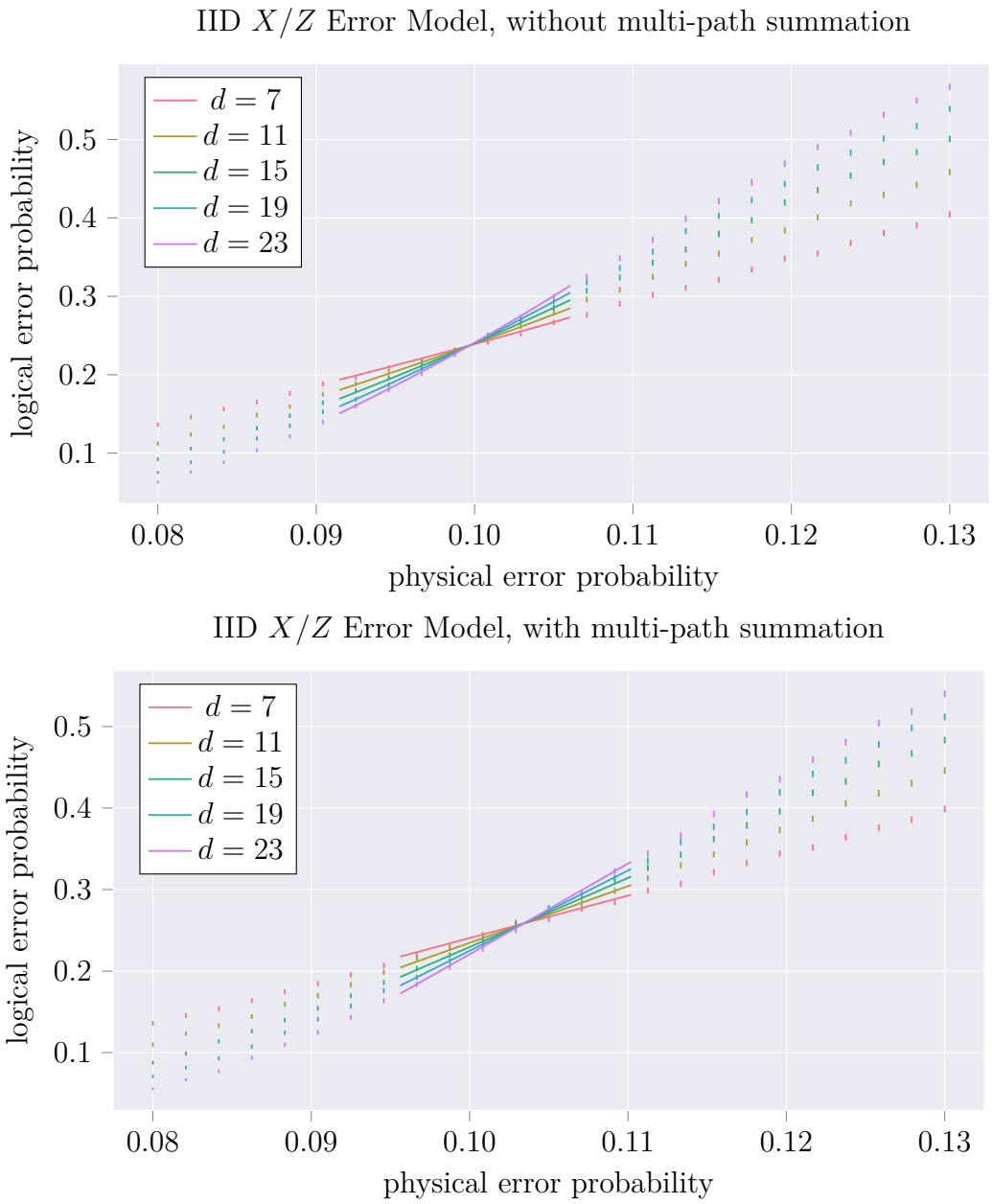

Figure 7: Threshold of the rotated surface code correcting IID X/Z errors. Above: Edge weights for MWPM are derived using the Manhattan distance, resulting threshold is $9.97 \pm 0.01\%$, lower than the value of 10.3% obtained using alternate boundary conditions in [36]. Below: Edge weights for MWPM are derived using path-counting, resulting in a threshold of $10.34 \pm 0.01\%$, compensating for the disadvantage imposed by the rotated boundary conditions. Error bars represent a 99% confidence interval. Solid lines are maximum-likelihood fits to data near the threshold.

where paths(q_p) is the set of paths leading from the initial vertex to the predecessor vertex q_p. This reduction is similar to the sum reduction used in path-counting, though now the sum is weighted by o_{q_p}. The path-counting function can be easily modified to evaluate this quantity:

function PATH_SUM(g, v_i, F, `o_q`) ▷ `o_q` o_q associated with edges (v, v') of g;
  for all $v \in$ `vertices`(g) **do**

```
        odds[v] ← 0
    end for
    odds[v_i] ← 1
    while true do
        if all(odds[f] ≠ 0 for f ∈ F) then return sum(odds[f] for f ∈ F)
        end if
        for all v ∈ vertices(g) do
            if all(odds[i] ≠ 0 for i ∈ incoming(v)) then
                odds[v] ← sum(o_q[(i, v)] * odds[i] for i ∈ incoming(v))
            end if
        end for
    end while
end function
```

This produces an approximate cost function, since we have neglected the possibility that there is an error inside the bounding box that does not cause a syndrome (a randomly applied stabiliser), and we have failed to account for the large number of less likely matchings that are equivalent to the likeliest matchings up to a stabiliser. Nevertheless, we will see that surface code decoding can be improved, once we have an accurate set of estimates for the marginal probability of each type of error on each qubit. In the following section, we review belief propagation, a method which can provide these estimates.

5 Belief Propagation

Belief propagation (BP) is a message-passing algorithm for calculating marginal probabilities, which has been applied to decoding quantum codes correcting Pauli noise [37]. To use BP for decoding, we first define the *Tanner graph* corresponding to a surface code, which contains a *qubit vertex* for every qubit in the code, and a *check vertex* for every stabiliser check. An edge exists between each check vertex and each qubit vertex in its support, see Figure 8.

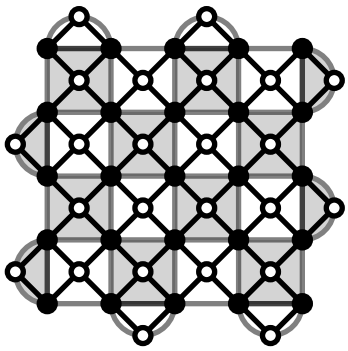

Figure 8: Tanner graph for the $d = 5$ surface code. Qubit vertices are shown in black, stabiliser vertices are shown in white. Edges exist between stabiliser vertices and the qubit vertices in the support of the corresponding stabiliser.

The algorithm begins with a probability distribution $p(E_q)$ (also called a *belief*), assigned to each qubit q, of the form $[1 - (p_Z + p_X + p_Y), p_Z, p_X, p_Y]$, equal to the *prior distribution* of error probabilities on each qubit (called 'the physical distribution' earlier). These beliefs are passed as messages to each of the neighbouring check vertices. Each check vertex c calculates

messages to be passed back to each of its neighbouring qubit vertices, one for each possible error E_q on the qubit:

$$m_{c\to q}(E_q) \propto \sum_{E_{q'},\, q'\in \mathrm{supp}(c)\setminus q} \left(\delta_{\mathrm{synd}_c, S_c\cdot E_c} \prod_{q'\in \mathrm{supp}(c)\setminus q} m_{q'\to c}(E_{q'}) \right). \tag{14}$$

Here, proportionality indicates that the messages are to be normalized after they are calculated. Each qubit vertex then calculates a set of messages to be passed back:

$$m_{q\to c}(E_q) \propto p(E_q) \prod_{c'\in \mathrm{supp}(q)\setminus c} m_{c'\to q}(E_q), \tag{15}$$

where proportionality again indicates that the messages are to be normalized.

Once the algorithm has converged, we calculate the updated beliefs (the conditional probabilities):

$$b_q(E_q) = p(E_q) \prod_{c\in \mathrm{supp}(q)} m_{c\to q}(E_q). \tag{16}$$

These beliefs can, in some instances, be used to provide an approximate maximum-likelihood error, but belief propagation is only guaranteed to provide an accurate estimate of likelihood on *trees* (acyclic graphs). In decoding, where Tanner graphs have many short cycles, the product of the likeliest errors on each qubit may not conform with the syndrome. For an example of this, consider the scenario in Figure 9 (also [37]). In this scenario, an X error has occurred on the upper-left corner, causing a single vertex to appear in the neighbouring square. With equal likelihood, this vertex could have been caused by an X error on the neighbouring qubit. In this case, belief propagation outputs a marginal probability $p_X + p_Y$ slightly below $1/2$ on each of these qubits. We refer to this scenario as a *split belief*. Taking the likeliest error on each qubit results in an error which does not conform with the syndrome, preventing the use of BP as a decoder. However, using a split belief in the approximate cost calculation in Section 4 still provides a final `odds` near 2, resulting in a small negative edge weight, and a good chance that the vertex in question will be matched to the boundary, as it should be.

Using d rounds of belief propagation and the cost calculation in Section 4, we obtain a threshold of $17.76\pm0.02\%$ when correcting depolarizing noise, see Figure 10. This is a significant improvement over normal MWPM decoding, though not optimal. In the following section, we discuss the complexity of the additional steps in the decoder, as well as potential enhancements and other applications.

6 Discussion/Conclusion

To our knowledge, little discussion exists in the literature regarding the effect of boundary conditions on the performance of surface code decoders. In addition, it is important to note the complexity of the additional steps used in the decoder studied in this work. In this section, we address these topics, and conclude by pointing out opportunities for future research.

6.1 Boundary Conditions

Intuition drawn from the statistical physics of spin models which are related to surface codes motivates the belief that the decoding threshold of a surface code is independent of 'finite-size' considerations such as boundary conditions and aspect ratio (if non-square lattices are used).

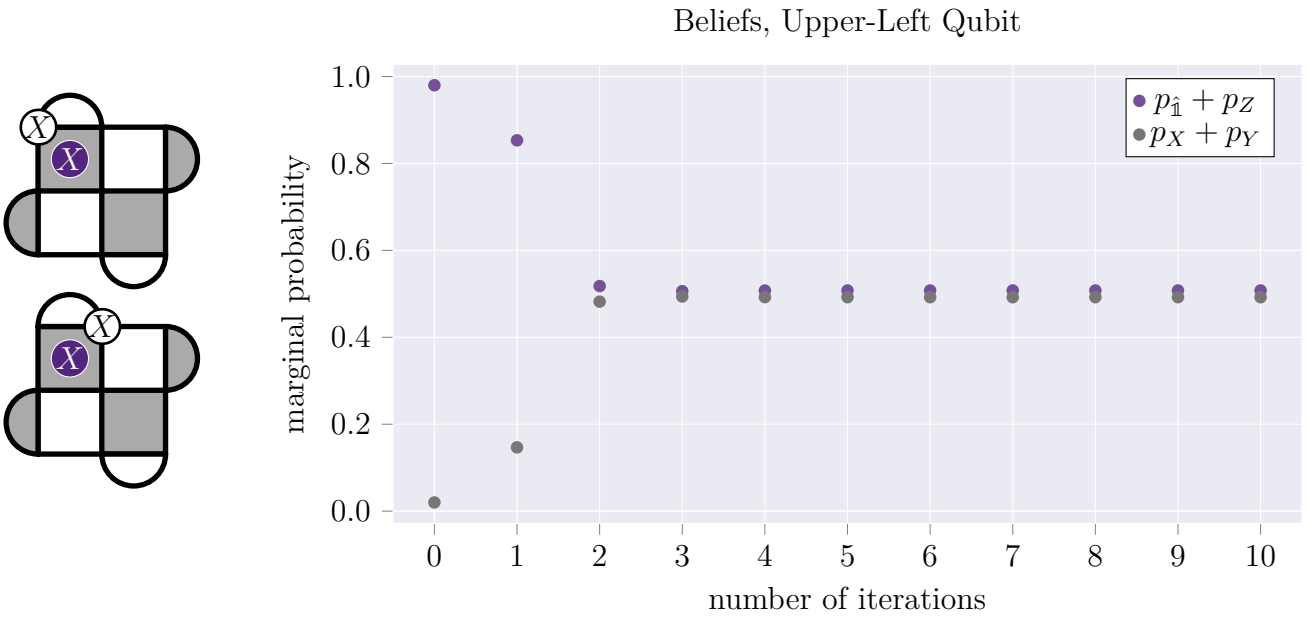

Figure 9: Left: Two possible errors which result in a split belief. Right: Convergence of beliefs when the true error is a weight-one X error on the upper-left qubit, the underlying error model has probabilities $[0.97, 0.01, 0.01, 0.01]$. A nearly-identical set of beliefs is supported on the neighbouring qubit.

Indeed, we see this belief to be approximately correct in the current work, though there are small deviations from decoding thresholds calculated elsewhere. To ensure that these deviations are related to boundary conditions, we re-evaluate the decoders we consider using smooth/rough boundary conditions in Figures 11 and 12, obtaining thresholds for comparison to Figures 7 and 10.

The fact that these thresholds exceed those for the rotated boundary conditions is confusing if the decoding threshold corresponds exactly with a thermodynamic property. However, this correspondence is only exact when using an *optimal* decoder [38]. The suboptimality of the matching-based decoders studied here thus allows for boundary conditions to influence decoder performance. Nonetheless, it is surprising that the change in threshold caused by the change in boundary condition should be discernible, and that it should be similar to the change in threshold caused by the change in decoding algorithm.

While we cannot explain this discrepancy completely, it can be explained partially by comparing the sets of minimum-length paths which traverse the two lattices, shown in Figure 13. When using smooth/rough boundary conditions, each vertex on the graph to be traversed only has one neighbouring edge which can be part of a minimum-length directed path from top to bottom, the edge immediately below. A minimum-length logical operator must, therefore, traverse the lattice directly from top to bottom. Therefore, there are exactly d minimum-length paths across the lattice. When using rotated boundary conditions, however, there are two edges neighbouring each vertex which can be part of a minimum-length path. The total number of minimum-length paths is therefore much larger, being bounded from below by $\binom{d}{\lfloor d/2 \rfloor}$, since the graph being traversed contains a $(\lfloor d/2 \rfloor)$-by-$(\lceil d/2 \rceil)$ bounding box as a subgraph. The number of minimum-length paths on the rotated lattice can also be calculated exactly in linear time by path-counting, as explained in Section 4.

In order for a minimum-weight decoder to produce a logical error when a small number of (for example, X) errors occur on the lattice at random, it must be possible to produce a minimum-length path by traversing at least $(\lceil d/2 \rceil)$ edges corresponding to qubits on which

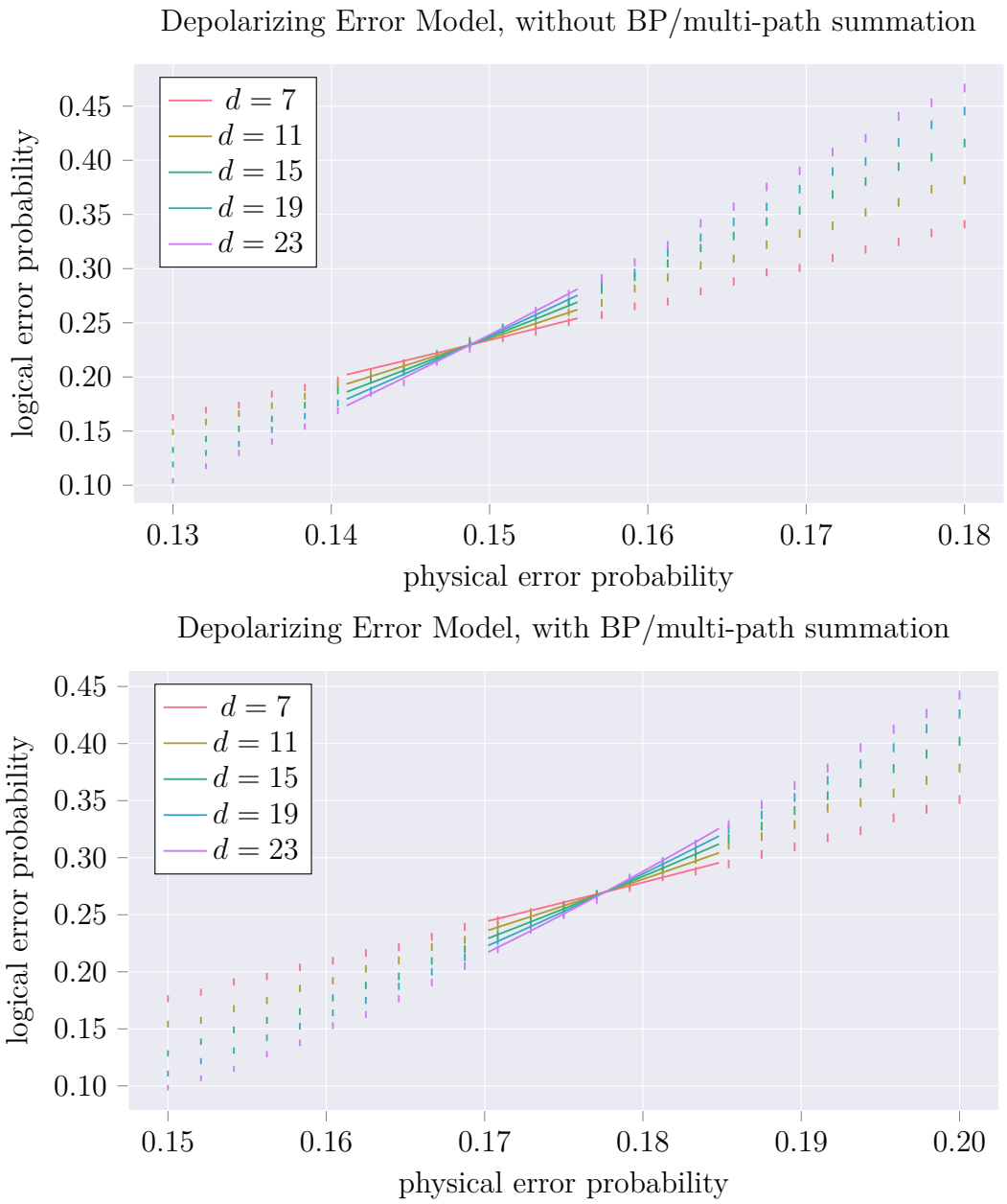

Figure 10: Threshold for the rotated surface code correcting depolarizing errors. Above: Using uniform edge weights, the threshold is $14.88 \pm 0.02\%$, less than the value of $15.5 \pm 0.5\%$ obtained using smooth/rough boundary conditions in [13]. Below: When using multi-path odds summation as discussed in Section 4, the threshold improves to $17.76 \pm 0.02\%$. Error bars are 99% confidence intervals. Solid lines are maximum-likelihood fits to data points near the threshold.

errors have occurred. When the number of available minimum-length paths is much larger, intuition would say that should occur with higher probability. This qualitative difference sheds some light on the origin of the difference in threshold, though we cannot explain it fully.

We note, in addition, that this difference in threshold has also been observed in predictions of logical error rates [39] and when using highly-correlated error models [40]. It is interesting to note that, for the two error models considered here, the difference in threshold between code families with different boundary conditions is smaller when the decoder is closer to optimality, though this is by no means conclusive. Another confounding factor, which prevents us from attributing these deviations solely to boundary conditions, is the behaviour of the Blossom V

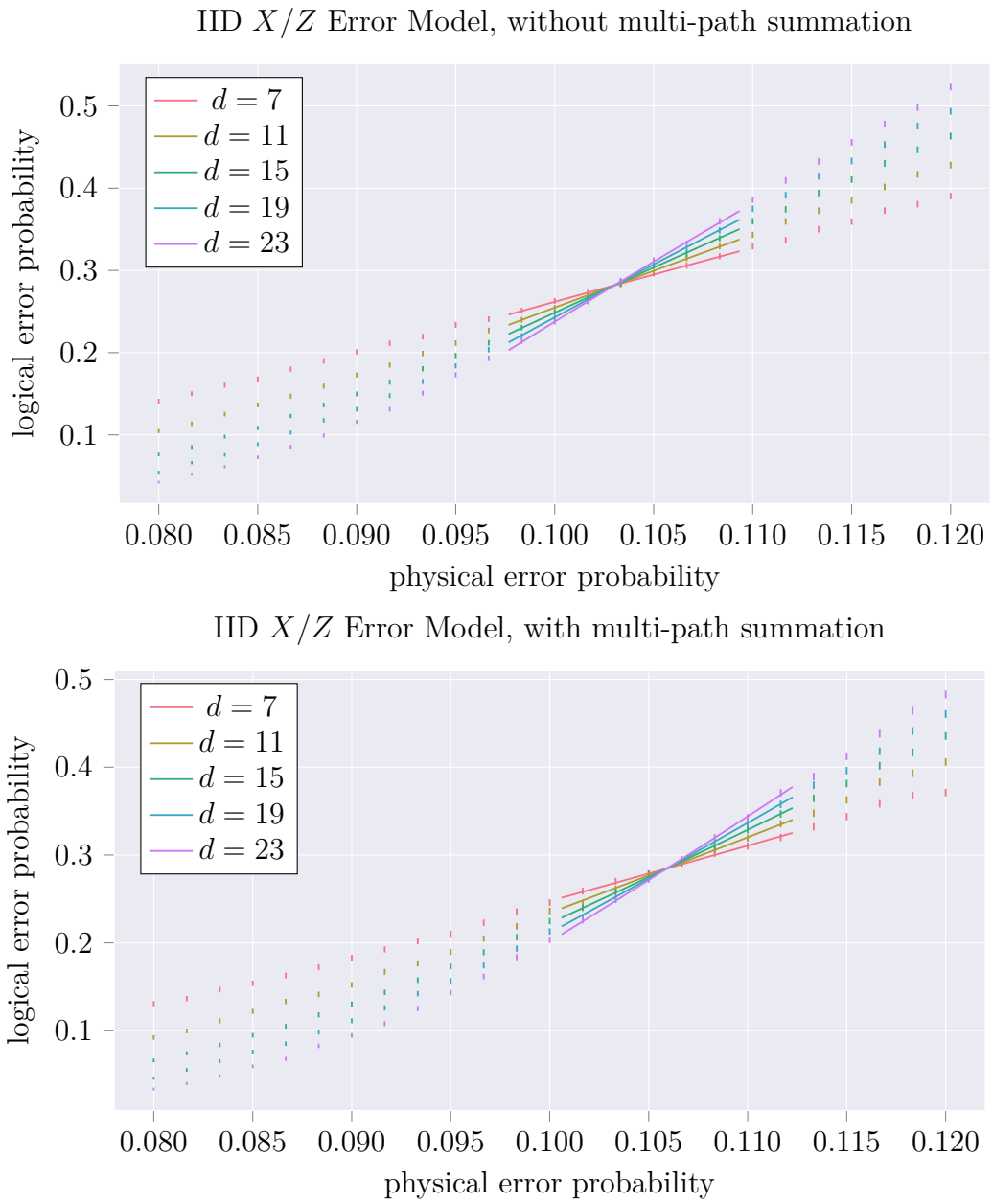

Figure 11: Replication of previous results using smooth/rough boundary conditions. The threshold error rate of $10.30 \pm 0.01\%$ is in accordance with earlier results. This threshold increases to $10.59 \pm 0.01\%$ using path-counting, comparable to the $\sim 10.6\%$ observed by Barrett/Stace/Doherty (with periodic boundary conditions) [21, 23]. Error bars are 99% confidence intervals. Solid lines are maximum-likelihood fits near the threshold.

MWPM implementation on graphs with non-unique minimum-weight perfect matchings, also noted by [22], which may also play a role.

6.2 Complexity

The decoder studied in this work does not require any modification to the Blossom algorithm, but adds pre-calculation steps which determine the edge weights. The additional overhead from belief propagation is $O(d^3)$, since it acts for d rounds and performs $O(d^2)$ operations in each round. The operations in each round of belief propagation can be carried out independently of

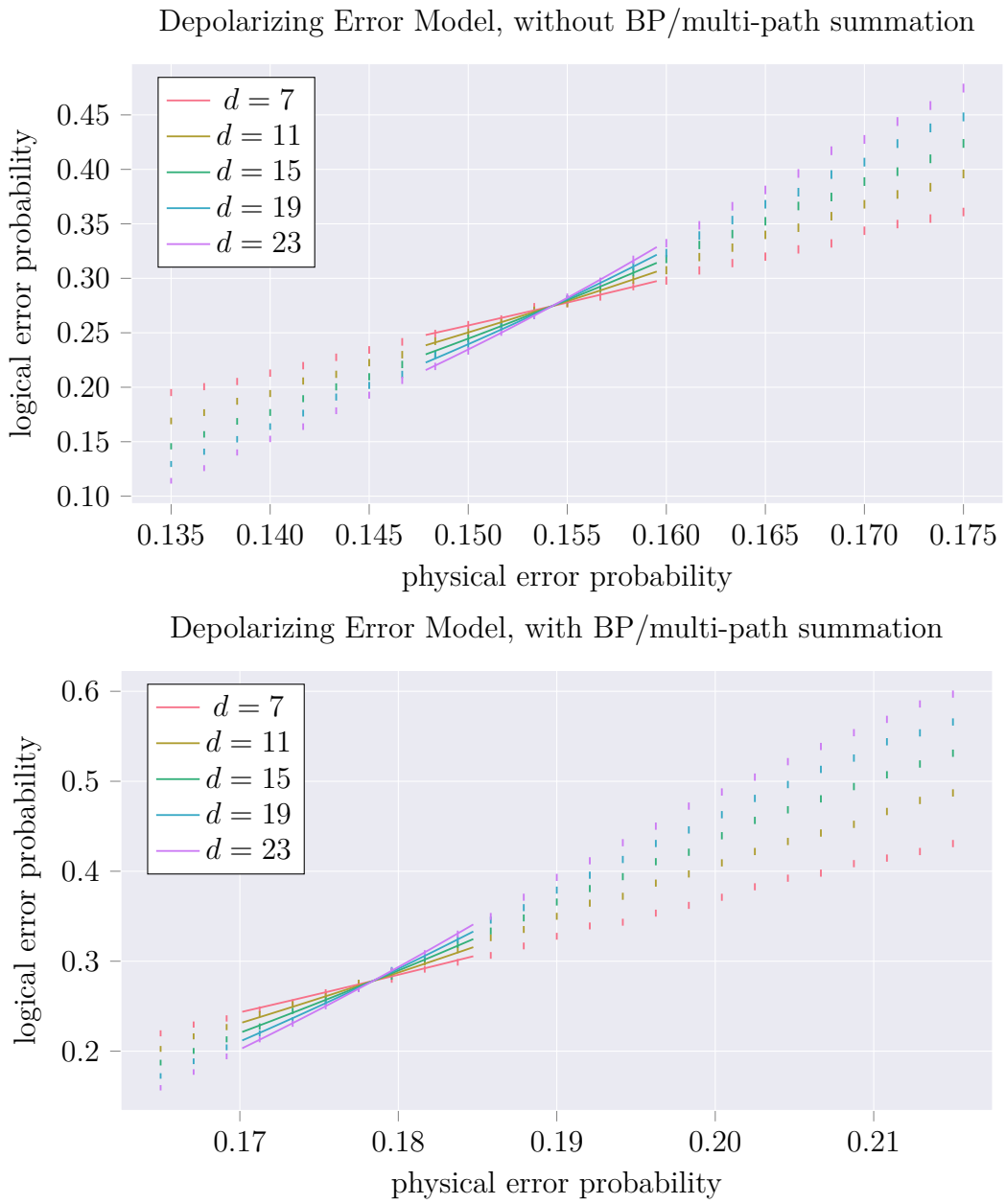

Figure 12: Application of belief propagation/multi-path summation to the decoding of surface codes with smooth/rough boundaries which are subjected to depolarising noise. A similar increase in threshold error rates is observed using these boundary conditions, with a threshold of $15.42 \pm 0.01\%$ obtained using a standard MWPM decoder, compatible with the observation of $15.5 \pm 0.5\%$ by [13]. This increases to $17.84 \pm 0.01\%$ when belief propagation and multi-path summation are incorporated. Error bars are 99% confidence intervals. Solid lines are maximum-likelihood fits near the threshold.

one another, allowing parallelization to $O(d)$ time using $O(d^2)$ processors. The path-summation function is more complex. If all path-sums are calculated, $O(d^4)$ operations are required, since there are $O(d^4)$ ancilla pairs, each one requiring a constant number of operations for calculation, assuming that they are performed in the appropriate order. A naïve parallel implementation of this algorithm would require one processor per pair of stabilisers ($O(d^4)$ processors) and $O(d)$ time, since the longest minimum-length paths are of $O(d)$ length. It is likely possible to reduce the number of processors, since each processor in the naïve parallelization only performs a calculation during one of the $O(d)$ timesteps of the algorithm. It is also likely possible to

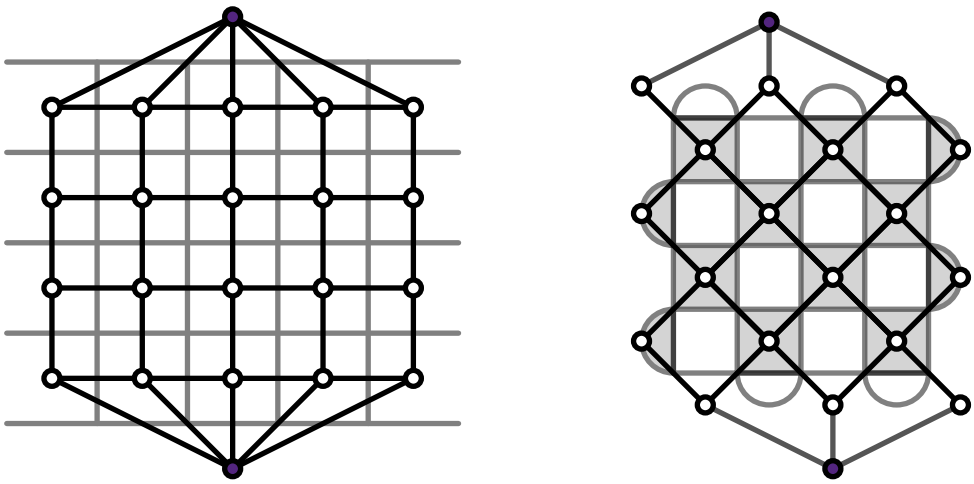

Figure 13: Graphs for which paths between the two coloured vertices correspond to logical Paulis on distance-5 surface codes. Left: Smooth/rough boundary conditions, right: rotated boundary conditions, with grey edges costing nothing to traverse. With smooth/rough boundary conditions, horizontal edges cannot be used in a minimum-length path, resulting in a number of minimum-length paths equal to d. With rotated boundary conditions, every edge can be part of a minimum-length path, resulting in a number of minimum-length paths which can be calculated by path-counting as explained in Section 4 (in this instance, there are 52 such paths), and lower-bounded by $\binom{d}{\lfloor d/2 \rfloor}$ (see text).

reduce the overall complexity of decoding by combining the proposed edge weight evaluation techniques with methods for pruning the syndrome graph [41], which is input to Blossom.

6.3 Future Work

To address the sub-optimality of the decoder, it is necessary to determine which of the approximations made in its definition are limiting accuracy. The first of these approximations is the set of marginal probabilities from belief propagation. The split belief in Figure 9 produces an `odds` near 2, where it should be much higher in principle (in the absence of other nearby syndromes, or a strong prior). Efforts to decrease this inaccuracy by altering the Tanner graph or prior used in BP have failed. Immediate future work will likely focus on finding a modification to or substitute for the BP algorithm which remedies this inaccuracy.

In any event, it will be interesting to apply multi-path odds summation to the $(2+1)$-dimensional problem of decoding the surface code when incorporating multiple rounds of measurement to combat errors in the syndrome measurement circuit, as well as to the problem of decoding 2D colour codes, which can be reduced to multiple correlated instances of surface code decoding [42].

Acknowledgements/Remarks

The authors would like to thank Tom O'Brien and Brian Tarasinski for the motivation to study this problem, as well as Barbara Terhal, Kasper Duivenvoorden, Christophe Vuillot, Nikolas Breuckmann and Ben Brown for useful discussions. Open-source software libraries used in this work include the Python libraries `numpy` [43], `networkx` [44], `scipy` [45], `matplotlib` [46], `seaborn` [47], and `emcee` [48], as well as Blossom V [15] in C++. Additional software produced for this work is available on Github [49].

References

[1] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997. DOI: 10.1137/S0097539795293172. URL http://dx.doi.org/10.1137/S0097539795293172.

[2] I. M. Georgescu, S. Ashhab, and Franco Nori. Quantum simulation. *Rev. Mod. Phys.*, 86:153–185, Mar 2014. DOI: 10.1103/RevModPhys.86.153. URL https://doi.org/10.1103/revmodphys.86.153.

[3] Daniel Gottesman. *Stabilizer Codes and Quantum Error Correction*. Caltech Ph.D. Thesis, 1997. DOI: 10.7907/rzr7-dt72. URL https://doi.org/10.7907/rzr7-dt72.

[4] A. M. Steane. Active stabilization, quantum computation, and quantum state synthesis. *Phys. Rev. Lett.*, 78:2252–2255, Mar 1997. DOI: 10.1103/PhysRevLett.78.2252. URL https://doi.org/10.1103/physrevlett.78.2252.

[5] Emanuel Knill and Raymond Laflamme. Theory of quantum error-correcting codes. *Phys. Rev. A*, 55:900–911, Feb 1997. DOI: 10.1103/PhysRevA.55.900. URL https://doi.org/10.1103/physreva.55.900.

[6] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes*. North-Holland Publishing Company, 2nd edition, 1978. ISBN 9780080570877. URL https://elsevier.com/books/the-theory-of-error-correcting-codes/macwilliams/978-0-444-85193-2.

[7] Daniel A. Lidar and Todd A. Brun. *Quantum Error Correction*. Cambridge University Press, 2013. DOI: 10.1017/CBO9781139034807. URL https://doi.org/10.1017/CBO9781139034807.

[8] A.Yu. Kitaev. Fault-tolerant quantum computation by anyons. *Annals of Physics*, 303 (1):2 – 30, 2003. ISSN 0003-4916. DOI: 10.1016/s0003-4916(02)00018-0. URL https://doi.org/10.1016/s0003-4916(02)00018-0.

[9] Michael H. Freedman and David A. Meyer. Projective plane and planar quantum codes. *Foundations of Computational Mathematics*, 1(3):325–332, jul 2001. DOI: 10.1007/s102080010013. URL https://doi.org/10.1007%2Fs102080010013.

[10] Sergey B. Bravyi and A. Yu. Kitaev. Quantum codes on a lattice with boundary. *arXiv preprint quant-ph/9811052*, 1998. URL https://arxiv.org/abs/quant-ph/9811052.

[11] H. Bombin and M. A. Martin-Delgado. Optimal resources for topological two-dimensional stabilizer codes: Comparative study. *Physical Review A*, 76(1), jul 2007. DOI: 10.1103/physreva.76.012305. URL https://doi.org/10.1103%2Fphysreva.76.012305.

[12] Eric Dennis, Alexei Kitaev, Andrew Landahl, and John Preskill. Topological quantum memory. *Journal of Mathematical Physics*, 43(9):4452–4505, 2002. DOI: 10.1063/1.1499754. URL http://dx.doi.org/10.1063/1.1499754.

[13] D.S. Wang, A.G. Fowler, A.M. Stephens, and L.C.L. Hollenberg. Threshold error rates for the toric and planar codes. *Quantum Information & Computation*, 10(5):456–469, 2010. DOI: 10.26421/QIC10.5. URL https://doi.org/10.26421/QIC10.5.

[14] Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17: 449–467, 1965. DOI: 10.4153/CJM-1965-045-4. URL http://dx.doi.org/10.4153/CJM-1965-045-4.

[15] Vladimir Kolmogorov. Blossom V: a new implementation of a minimum cost perfect matching algorithm. *Mathematical Programming Computation*, 1:43–67, 2009. DOI: 10.1007/s12532-009-0002-8. URL http://dx.doi.org/10.1007/s12532-009-0002-8.

[16] Chenyang Wang, Jim Harrington, and John Preskill. Confinement-higgs transition in a disordered gauge theory and the accuracy threshold for quantum memory. *Annals of*

Physics, 303(1):31–58, jan 2003. DOI: 10.1016/s0003-4916(02)00019-2. URL https://doi.org/10.1016%2Fs0003-4916%2802%2900019-2.

[17] F. Merz and J. T. Chalker. Two-dimensional random-bond ising model, free fermions, and the network model. *Physical Review B*, 65(5), jan 2002. DOI: 10.1103/physrevb.65.054425. URL https://doi.org/10.1103%2Fphysrevb.65.054425.

[18] A. Honecker, M. Picco, and P. Pujol. Universality class of the nishimori point in the 2d±JRandom-bond ising model. *Physical Review Letters*, 87(4), jul 2001. DOI: 10.1103/physrevlett.87.047201. URL https://doi.org/10.1103%2Fphysrevlett.87.047201.

[19] Nicolas Delfosse and Jean-Pierre Tillich. A decoding algorithm for CSS codes using the X/Z correlations. In *2014 IEEE International Symposium on Information Theory*. IEEE, jun 2014. DOI: 10.1109/isit.2014.6874997. URL https://doi.org/10.1109%2Fisit.2014.6874997.

[20] Austin G. Fowler. Optimal complexity correction of correlated errors in the surface code. *arXiv preprint arXiv:1310.0863*, 2013. URL https://arxiv.org/abs/1310.0863.

[21] Thomas M. Stace, Sean D. Barrett, and Andrew C. Doherty. Thresholds for topological codes in the presence of loss. *Physical Review Letters*, 102(20), may 2009. DOI: 10.1103/physrevlett.102.200501. URL https://doi.org/10.1103%2Fphysrevlett.102.200501.

[22] Sean D. Barrett and Thomas M. Stace. Fault tolerant quantum computation with very high threshold for loss errors. *Physical Review Letters*, 105(20), nov 2010. DOI: 10.1103/physrevlett.105.200502. URL https://doi.org/10.1103%2Fphysrevlett.105.200502.

[23] Thomas M. Stace and Sean D. Barrett. Error correction and degeneracy in surface codes suffering loss. *Physical Review A*, 81(2), feb 2010. DOI: 10.1103/physreva.81.022317. URL https://doi.org/10.1103%2Fphysreva.81.022317.

[24] Project Euler: Problem 15. https://projecteuler.net/problem=15. Accessed: 2017-08-07.

[25] H Freund and P Grassberger. The ground state of the + or − J spin glass from a heuristic matching algorithm. *Journal of Physics A: Mathematical and General*, 22(18):4045–4059, sep 1989. DOI: 10.1088/0305-4470/22/18/036. URL https://doi.org/10.1088%2F0305-4470%2F22%2F18%2F036.

[26] Adrian Hutter, James R. Wootton, and Daniel Loss. Efficient markov chain monte carlo algorithm for the surface code. *Physical Review A*, 89(2):022326, feb 2014. DOI: 10.1103/physreva.89.022326. URL https://doi.org/10.1103%2Fphysreva.89.022326.

[27] Sergey Bravyi, Martin Suchara, and Alexander Vargo. Efficient algorithms for maximum likelihood decoding in the surface code. *Physical Review A*, 90(3), sep 2014. DOI: 10.1103/physreva.90.032326. URL https://doi.org/10.1103%2Fphysreva.90.032326.

[28] Guillaume Duclos-Cianci and David Poulin. A renormalization group decoding algorithm for topological quantum codes. In *2010 IEEE Information Theory Workshop*. IEEE, aug 2010. DOI: 10.1109/cig.2010.5592866. URL https://doi.org/10.1109%2Fcig.2010.5592866.

[29] Leslie G. Valiant. Quantum circuits that can be simulated classically in polynomial time. *SIAM Journal on Computing*, 31(4):1229–1254, jan 2002. DOI: 10.1137/s0097539700377025. URL https://doi.org/10.1137%2Fs0097539700377025.

[30] Sergey Bravyi. Lagrangian representation for fermionic linear optics. *Quantum Information & Computation*, 5(3):216–238, 2005. DOI: 10.26421/QIC5.3. URL https://doi.org/10.26421/QIC5.3.

[31] Guifré Vidal. Efficient classical simulation of slightly entangled quantum computations. *Physical Review Letters*, 91(14), oct 2003. DOI: 10.1103/physrevlett.91.147902. URL https://doi.org/10.1103%2Fphysrevlett.91.147902.

[32] Guillaume Duclos-Cianci and David Poulin. A renormalization group decoding algorithm for topological quantum codes. *Information Theory Workshop (ITW), IEEE*, pages 1 – 5, 2010. DOI: 10.1109/CIG.2010.5592866. URL http://dx.doi.org/10.1109/CIG.2010.5592866.

[33] Adrian Hutter, Daniel Loss, and James R Wootton. Improved HDRG decoders for qudit and non-abelian quantum error correction. *New Journal of Physics*, 17(3):035017, mar 2015. DOI: 10.1088/1367-2630/17/3/035017. URL https://doi.org/10.1088%2F1367-2630%2F17%2F3%2F035017.

[34] D.J.C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003. ISBN 9780521642989. URL https://books.google.nl/books?id=AKuMj4PN_EMC.

[35] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms*. Computer science. MIT Press, 2009. ISBN 9780262533058. URL https://books.google.nl/books?id=aefUBQAAQBAJ.

[36] Ashley M. Stephens. Fault-tolerant thresholds for quantum error correction with the surface code. *Physical Review A*, 89(2), feb 2014. DOI: 10.1103/physreva.89.022321. URL https://doi.org/10.1103%2Fphysreva.89.022321.

[37] David Poulin and Yeojin Chung. On the iterative decoding of sparse quantum codes. *Quantum Information & Computation*, 8(10):987–1000, 2008. DOI: 10.26421/QIC8.10. URL https://doi.org/10.26421/QIC8.10.

[38] S. Flammia. personal communication, 2017.

[39] Austin G. Fowler. Accurate simulations of planar topological codes cannot use cyclic boundaries. *Physical Review A*, 87(6), jun 2013. DOI: 10.1103/physreva.87.062320. URL https://doi.org/10.1103%2Fphysreva.87.062320.

[40] David K. Tuckett, Stephen D. Bartlett, and Steven T. Flammia. Ultrahigh error threshold for surface codes with biased noise. *Physical Review Letters*, 120(5), jan 2018. DOI: 10.1103/physrevlett.120.050505. URL https://doi.org/10.1103%2Fphysrevlett.120.050505.

[41] Austin G. Fowler, Adam C. Whiteside, and Lloyd C. L. Hollenberg. Towards practical classical processing for the surface code. *Physical Review Letters*, 108(18), may 2012. DOI: 10.1103/physrevlett.108.180501. URL https://doi.org/10.1103%2Fphysrevlett.108.180501.

[42] Nicolas Delfosse. Decoding color codes by projection onto surface codes. *Physical Review A*, 89(1), jan 2014. DOI: 10.1103/physreva.89.012317. URL https://doi.org/10.1103%2Fphysreva.89.012317.

[43] Stéfan van der Walt, S Chris Colbert, and Gaël Varoquaux. The NumPy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, mar 2011. DOI: 10.1109/mcse.2011.37. URL https://doi.org/10.1109%2Fmcse.2011.37.

[44] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008. URL https://networkx.github.io/.

[45] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. URL https://scipy.org/. [Online; accessed 2017-09-01].

[46] John D. Hunter. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. DOI: 10.1109/mcse.2007.55. URL https://doi.org/10.1109%2Fmcse.2007.55.

[47] Seaborn: statistical data visualization. https://seaborn.pydata.org/. Accessed: 2017-09-01.

[48] Daniel Foreman-Mackey, David W. Hogg, Dustin Lang, and Jonathan Goodman. emcee: The MCMC hammer. *Publications of the Astronomical Society of the Pacific*, 125(925): 306–312, mar 2013. DOI: 10.1086/670067. URL https://doi.org/10.1086%2F670067.

[49] `sc_decoding`. https://github.com/bcriger/sc_decoding.