

Budapest University of Technology and Economics

DIPLOMA THESIS

Optimizing quantum error-correcting codes with theoretical and numerical tools

Author:

Áron Márton

MSc student in physics
BME Faculty of Natural Sciences

Supervisor:

János Asbóth

Associate Professor
BME Faculty of Natural Sciences
Department of Theoretical Physics

November 29, 2024



Diplomamunka feladat a Fizikus mesterképzési szak hallgatói számára

| | |
|--|--|
| A hallgató neve: Márton Áron | specializációja: Fizikus MSc - kutatófizikus |
| A záróvizsgát szervező tanszék neve: BME TTK Elméleti Fizika Tanszék | |

| |
|---|
| A témavezető neve: Asbóth János - tanszéke: BME TTK Elméleti Fizika Tanszék - beosztása: egyetemi docens - email címe: asboth.janos@ttk.bme.hu |
|---|

| |
|--|
| A kidolgozandó feladat címe: Kvantumos hibajavító kódok elméleti és numerikus optimalizációja |
| A téma rövid leírása, a megoldandó legfontosabb feladatok felsorolása: A kvantumos hibajavító kódok szükségesek ahhoz, hogy zajos elemi alkatrészekkel is megbízható kvantumszámítást lehessen végezni. A kódok működését numerikusan sokszor Heisenberg képben, Clifford szimulációval végzik, de a termodinamikai határesetben viselkedésük statisztikus fizikai modellekre való leképezésekkel érhető meg - a legegyszerűbb hibamodellekre (véletlen Pauli-hibák). Általánosabb hibák esetén (pl. koherens hibák) a leképezések és a szimulációs eszközök is csak megszorításokkal használhatók. A kutatási feladat ezen eszközök fejlesztésével, ill. a Majorana-fermionok használatával (FLO szimuláció) vizsgálni bizonyos kvantumos hibajavító kódokat. Koherens hibák esetén milyen statisztikus fizikai modellekre való leképezés létezik? Az ún. felületi kód esetére a kísérletihez közelebb álló zajmodellek mellett milyen dekódoló algoritmust érdemes használni? |

| |
|--------------------------------|
| A záróvizsga kijelölt tételei: |
|--------------------------------|

| |
|--------|
| Dátum: |
|--------|

| | | | |
|--------------------|-----------------------|------------------------------|--|
| Hallgató aláírása: | Témavezető aláírása*: | Tanszéki konzulens aláírása: | A témakiirását jóváhagyom (tanszékvezető aláírása): |
|--------------------|-----------------------|------------------------------|--|

*A témavezető jelen feladatkiírás aláírásával tudomásul veszi, hogy a BME TVSZ 145. és 146. § alapján az egyetem a képzési célok megvalósulása érdekében a szakdolgozatok, illetve diplomamunkák nyilvánosságát tartja elsődlegesnek. A hozzáférés korlátozása csak kivételes esetben, a dékán előzetes hozzájárulásával lehetséges.

Declaration of originality

Undersigned **Áron Márton** physics MSc student of the Budapest University of Technology and Economics, declare that I wrote this thesis without the use of any unauthorized assistance, under the guidance of my supervisor. I have cited every source I used. For any part that I have borrowed verbatim, or paraphrased while preserving its original meaning, I have marked its source of origin.

Identicality of digital and printed versions

Undersigned **Áron Márton** physics MSc student of the Budapest University of Technology and Economics, declare that the digital and printed copies of the thesis are identical in all respects.

Abstract

To implement practically useful quantum algorithms it is essential to suppress noise during quantum computation. Fault-tolerant quantum computation achieves this by storing the quantum information in error-corrected logical qubits and protecting logical operations from environmental noise. In this work, I investigate a fault-tolerant lattice-surgery-based logical CNOT operation between two logical qubits encoded into surface code patches. I fully characterize the two-qubit logical error channel of the CNOT operation under both phenomenological and circuit-level noise models. I show that a symmetry of the logical CNOT protocol appears in the logical error channel, guaranteeing a symmetry in the noise structure. Numerical simulations show that the symmetric noise structure is robust against the weak symmetry breaking of realistic circuit-level noise. Additionally, I demonstrate that X and Z errors become uncorrelated at the logical level under minimum weight perfect matching decoding. Consequently, the logical noise channel can be described with three logical error parameters with high numerical precision. Finally, I find that the CNOT protocol shows a threshold behaviour. I calculate numerically the threshold values for two phenomenological and one circuit-level noise models, finding that they match the threshold values observed in memory experiments.

1 Acknowledgements

First of all, I would like to thank my supervisor, János Asbóth, for his guidance, useful insights, and the help he provided throughout my Master's studies. I also want to thank Bálint Domokos, with whom I worked on a significant part of this project, resulting in a manuscript [1]. Without the countless discussions and working hours with Bálint, this thesis could not have been completed.

This research was supported by the Ministry of Culture and Innovation and the National Research, Development and Innovation Office within the Quantum Information National Laboratory of Hungary (Grant No. 2022-2.1.1-NL-2022-00004). Supported by the Horizon Europe research and innovation programme of the European Union through the HORIZON-CL4-2022-QUANTUM01-SGA project 101113946 OpenSuperQPlus100 of the EU Flagship on Quantum Technologies. I acknowledge the support of the Wigner Research Centre for Physics' trainee program.

Contents

| | |
|--|-----------|
| 1 Acknowledgements | 5 |
| 2 Introduction | 7 |
| 3 Quantum error correction with the surface code | 8 |
| 3.1 Correcting Pauli errors | 10 |
| 3.2 Correcting phenomenological readout errors | 12 |
| 3.3 Correcting circuit-level noise | 14 |
| 3.4 Minimum weight perfect matching decoder | 16 |
| 4 Surface code-based fault-tolerant logical operations | 17 |
| 4.1 A possible route towards fault-tolerant quantum computation | 17 |
| 4.2 Logical identity channel and memory experiments. | 18 |
| 4.3 Spacetime diagrams and logical errors | 19 |
| 5 Fault-tolerant CNOT gate | 22 |
| 5.1 Lattice surgery | 23 |
| 5.2 Spacetime diagram and logical error classes of the CNOT protocol | 25 |
| 6 Simulation methods | 27 |
| 6.1 Phenomenological errors and Pauli frame simulation | 27 |
| 6.2 Circuit-level noise and process tomography | 28 |
| 7 The structure of the logical noise channel | 31 |
| 7.1 Symmetry transformation of the CNOT protocol | 31 |
| 7.2 Numerical results for the logical noise channel | 33 |
| 7.3 Threshold behaviour | 36 |
| 8 Conclusion | 37 |

2 Introduction

In order to run useful quantum algorithms on real quantum computers sufficiently low error rates are required [2]. To reach low enough error rates fault-tolerant quantum computation with error-corrected logical qubits is needed. One of the most promising quantum error-correcting code, which is capable of performing universal fault-tolerant quantum logic, is the so-called surface code [3–5]. The popularity of this scheme is based on its planar connectivity, scalability and the high error threshold [5–7]. Recent experiments have realized surface code patches as quantum memories [8–11] and also the first experimental demonstrations of fault-tolerant logical operations have been done [12–15].

Several innovative ideas have been developed over the years in order to achieve universal fault-tolerant quantum computation with the surface code [16–19]. Among them is the so-called lattice surgery protocol [17, 20], which allows for the measurement of multi-qubit logical Pauli operators, therefore, the implementation of so-called Clifford-operations. These operations transform Pauli operators into other Pauli operators. An example for a multi-qubit Clifford operation is the CNOT gate. A natural question arises regarding the performance of surface code-based fault-tolerant quantum computation: what kind of logical noise emerges from the errors at the physical level?

In this work I answer this question for the logical error channel describing the noise of a lattice-surgery-based CNOT operation. I investigate physical error models at both the phenomenological and circuit levels. I show that a symmetry of the logical CNOT operation appears in the logical error channel, thus guaranteeing a special noise structure. This finding still holds (up to good numerical precision) for the case when the circuit-level errors weakly break the symmetry of the operation. Moreover, I show that the logical error probabilities factorize into X and Z parts under minimum weight perfect matching decoding, thus the two-qubit logical Pauli error channel of the CNOT can be described with just three logical error parameters. This also holds (up to good numerical precision) in the case when the underlying physical X and Z errors are correlated.

An important concept in quantum error correction is the error threshold. This is a critical physical error rate below which the logical error rates can be suppressed by scaling up the parameters (number of qubits, measurement rounds) of the protocol. I calculate numerically the threshold of the logical CNOT gate for two phenomenological and one circuit-level noise models, and show that these values agree with the quantum memory thresholds. These findings contribute to a deeper understanding of the noise structure

in surface code-based logical operations, and are of interest for experimental implementations of lattice surgery protocols.

The rest of the thesis is structured as follows: In Sec. 3 I introduce the surface code and show how error correction works in the case of random Pauli, phenomenological readout and circuit-level errors. In Sec. 4 I summarize a possible implementation of surface code-based universal fault-tolerant quantum computation, and introduce spacetime diagrams through the examples of logical identity channel and memory experiments. In Sec. 5 I describe the construction of the lattice-surgery-based CNOT operation. In Sec. 6 I explain some of the details of the numerical simulations, and finally in Sec. 7 I show the results about the structure of the logical noise channel, and error thresholds.

3 Quantum error correction with the surface code

The surface code is an encoding scheme, originally proposed by Kitaev [3], in which a logical quantum bit is encoded into a highly entangled state of many physical quantum bits. The so-called rotated surface code [21] realizes a slightly better utilization of resources than the original geometry. A rotated surface code patch consists of $n = d^2$ physical quantum bits, with d an odd integer, as it is depicted in Fig 1. The logical quantum bit is encoded into the +1 eigensubspace of the X - and Z -stabilizers, defined as:

$$B_f = \prod_{j \in \partial f} Z_j; \quad A_f = \prod_{j \in \partial f} X_j, \quad (1)$$

where X_j and Z_j are the usual Pauli operators acting on qubit j and ∂f consists of qubits at the corners of the corresponding face f .

The logical states are the +1 eigenstates of all the stabilizers,

$$B_f |\bar{\psi}\rangle = |\bar{\psi}\rangle; \quad A_f |\bar{\psi}\rangle = |\bar{\psi}\rangle \quad \forall A_f, B_f. \quad (2)$$

These states form a two-dimensional subspace, which is called the logical subspace. More precisely the stabilizers defined in Eq. (1) are just the generators of the stabilizer group, which leaves the logical states invariant. In this work, I will use the term stabilizer for the generators of the stabilizer group, following the usual practice of the literature.

The logical subspace is spanned by the logical $|\bar{0}\rangle$ and $|\bar{1}\rangle$ states, which are defined as:

$$|\bar{0}\rangle = \prod_{f \in \text{light}} \frac{1 + A_f}{\sqrt{2}} |0\rangle^{\otimes n}; \quad |\bar{1}\rangle = \prod_{f \in \text{light}} \frac{1 + A_f}{\sqrt{2}} |1\rangle^{\otimes n}. \quad (3)$$

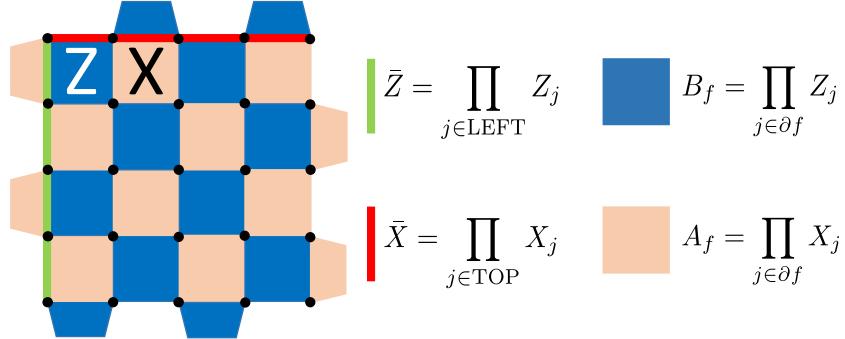


Figure 1: A distance 5 surface code patch. Physical qubits are represented as black dots. Z - (X -) stabilizers are products of Z (X) operators acting on the qubits at the corners of dark (light) faces. A logical Z (X) operator is a string of single qubit Z (X) operators laying at the left (top) boundary. The logical Z (X) operator is connecting the bottom (left) and the top (right) boundaries, it is depicted as a green (red) line.

Logical operators act inside the logical subspace (commuting with all the stabilizers), but do not act trivially on the logical states. Logical \bar{X} ($\bar{X}|0\rangle = |\bar{1}\rangle$, $\bar{X}|1\rangle = |\bar{0}\rangle$) and logical \bar{Z} ($\bar{Z}|0\rangle = |\bar{0}\rangle$, $\bar{Z}|1\rangle = -|\bar{1}\rangle$) operators can be identified as X and Z strings acting on the qubits located at the top and left side of the code:

$$\bar{X} = \prod_{j \in \text{TOP}} X_j; \quad \bar{Z} = \prod_{j \in \text{LEFT}} Z_j. \quad (4)$$

Notably, the products of a logical operator and stabilizers are just as good representations of the logical operator as the original one, because they are acting in the same way inside the logical subspace.

The boundary conditions of the surface code play an important role. In the rotated geometry at the top and at the bottom boundaries only Z -stabilizers are present, therefore, these are called Z -boundaries. On the other hand, left and right boundaries are called X -boundaries because only X -stabilizers are present there. Logical X (Z) operators can be seen as Pauli X (Z) strings connecting X - (Z -) boundaries. The topological protection of the encoded logical quantum information is based on the spatial separation of the same boundary types.

The so-called code distance quantitatively describes the separation of boundaries and, therefore, the efficiency of the topological protection. The code distance d is the weight of the smallest weight logical error operator. In the case of the surface code, this is equal to the linear size of the square grid, in Fig. 1 this is $d = 5$.

3.1 Correcting Pauli errors

To show how the surface code can detect and correct local Pauli noise let me consider a Pauli error model with random Pauli X , Y and Z errors. In this model Pauli errors occur on each physical qubit with probabilities p_x , p_y and p_z . This can be described with the following error channel acting on the j -th data qubit:

$$\varepsilon_j(\rho) = (1 - p_x - p_y - p_z)\rho + p_x X_j \rho X_j + p_y Y_j \rho Y_j + p_z Z_j \rho Z_j, \quad (5)$$

where ρ is the density matrix of the whole surface code.

The overall error channel describes a probabilistic process, where different error scenarios can happen with some probabilities. In a concrete error scenario an error string E is acting on the quantum state of the surface code. This error string describes the situation when some of the qubits ($j \in E_x$) suffered an X error, and some of them ($k \in E_z$) suffered a Z error. Y errors are considered as joint X and Z errors, so qubits which are in E_x and E_z simultaneously suffered a Y error,

$$E = \prod_{k \in E_x} X_k \prod_{j \in E_z} Z_j. \quad (6)$$

Those error strings which bring the quantum state of the code out of the logical subspace anti-commute with at least one stabilizer. Therefore, these error strings can be detected with the measurement of the stabilizers. The collection of the stabilizer measurement outcomes after the realization of a concrete error scenario is the so-called syndrome,

$$s = (s_1, \dots, s_{n-1}), \quad (7)$$

where $s_f = \pm 1$ is the outcome of the stabilizer measurement corresponds to the face f . Each error string E uniquely determines the stabilizer measurement outcomes. I denote the syndrome corresponding to the error string E as

$$s(E) = \partial E, \quad (8)$$

where the ∂ symbol refers to the fact that syndromes can be seen as zero-dimensional boundaries of one-dimensional error strings [6].

It is important to note that ∂E is not an invertible function, there are multiple different error strings, which cause the same syndrome. This degeneracy among error strings makes the error correction a non-trivial problem. An example of an error string and the corresponding syndrome is shown in Fig.2

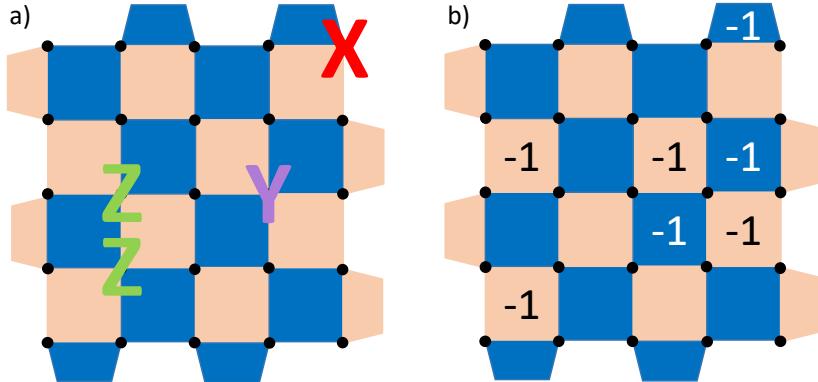


Figure 2: An example of an error string and the corresponding syndrome. In a) the red X, the green Z, and the purple Y letters denote the locations of X , Z and Y errors, which altogether form the error string E . In b) the corresponding syndrome ∂E is shown. The -1 stabilizer measurement outcomes are shown explicitly, while the blank faces stand for the $+1$ measurement outcomes.

In order to correct errors it is necessary to find a correction operator C_s for the given syndrome s , which brings back code into the logical subspace. For this the correction must fulfil the criterion

$$\partial C_s = s. \quad (9)$$

The problem of finding the correction operator for a given syndrome is called the decoding problem. The maximum likelihood decoder [6], provides the best possible solution for this problem, but the runtime of this decoder typically scales exponentially with the code distance. Therefore, several other sub-optimal, but more scalable approaches have been constructed [22–25]. In this work I used the so-called minimum weight perfect matching decoder [6, 23] which I will explain in more detail later.

The correction operator C_s corrects the given error string E if any logical state $|\bar{\psi}\rangle$ remains unchanged after error correction:

$$C_s E |\bar{\psi}\rangle = |\bar{\psi}\rangle. \quad (10)$$

Otherwise, the surface code suffers a logical Pauli error:

$$C_s E |\bar{\psi}\rangle = \bar{P} |\bar{\psi}\rangle, \quad \bar{P} \in \{\bar{X}, \bar{Y}, \bar{Z}\}. \quad (11)$$

The performance of the surface code under Pauli errors and various decoders is well understood, and can be found in the literature [5–7, 26].

3.2 Correcting phenomenological readout errors

To model the environmental noise more accurately measurement errors have to be taken into account in addition to the previously described Pauli errors. The most simple error model of noisy measurements is the so-called phenomenological readout error model, which describes the measurement errors as classical bit-flip errors in the measurement outcome during the readout process. So each bit of the measured syndrome can change sign with probability q :

$$P(s_i \rightarrow -s_i) = q; \quad P(s_i \rightarrow s_i) = 1 - q. \quad (12)$$

To protect the encoded logical quantum bit during a logical operation from phenomenological readout errors as well as from random Pauli noise multiple rounds of stabilizer measurements have to be performed. To tackle this problem in a similar manner as random Pauli errors I define check generators, which are the generalizations of stabilizers in the case of multiple measurement rounds. Each check generator corresponds to a stabilizer at given timesteps t and $t - 1$. The value of a check generator \mathfrak{s}_f^t is defined through the values of the measurement outcomes of the corresponding stabilizer in timesteps t (s_f^t) and $t - 1$ (s_f^{t-1}),

$$\mathfrak{s}_f^t = \begin{cases} +1 & \text{if } s_f^{t-1} = s_f^t \\ -1 & \text{if } s_f^{t-1} \neq s_f^t. \end{cases} \quad (13)$$

Check generator value generalize the measurement outcome corresponding to stabilizers.

Check generators lead to the formation of the so-called 3D syndrome as a bit-string composed of the values of check generators.

$$\mathcal{S} = (\mathfrak{s}_1^1, \dots, \mathfrak{s}_{n-1}^T), \quad (14)$$

where T is the number of check generator layers during a logical operation which is closely related to the number of stabilizer measurement rounds, but it also depends on the boundary conditions. The boundary conditions for check generators at the lower and upper timelike boundaries is determined by the logical operation which we want to perform. Now I leave the question of boundary conditions open and I will get back to it in Sec. 4.

Check generators can detect both random Pauli and phenomenological readout errors. To see this it is useful to think of errors in terms of spacelike and timelike errors instead of Pauli and readout errors.

Spacelike error: Pauli error on a physical qubit.

Timelike error: Phenomenological readout error on a stabilizer. (15)

To handle these two kinds of errors in a unified framework I introduce virtual qubits representing stabilizer measurements. Virtual qubits are located at the middle of stabilizers (in space) and at regular intervals between consecutive layers (time steps) of physical qubits (in time), see Fig. 3. Spacelike errors are located on physical qubits and timelike errors are on virtual ones. In this picture check generators are checking the parity of errors (both spacelike and timelike) happened on the physical qubits, correspond to a given stabilizer in a given timestep, and on the the virtual qubits "above" and "below" this stabilizer. The check generators form the so-called check graph, which contains check generators as vertices and qubits (both physical and virtual) as edges. In Fig.3 the check generators are shown for multiple measurement rounds.

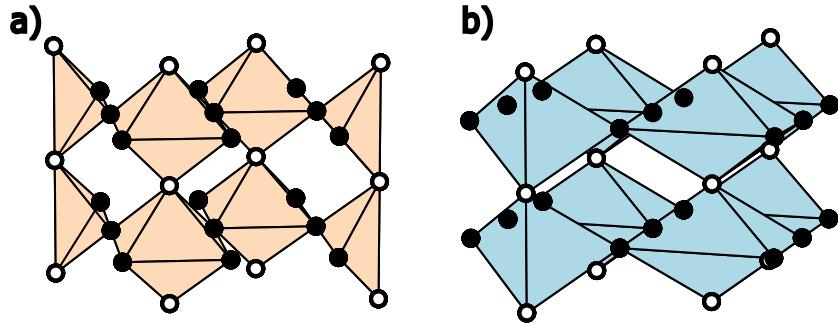


Figure 3: Check generators of a surface code considered during three consecutive stabilizer measurement rounds, the time flows upwards. Black dots represent physical, while white dots represent virtual qubits. In a) the X -check generators are shown as orange volumes. An X -check generator can detect Z -type errors (both timelike and spacelike) occurred on the qubits (both physical and virtual) at its corners. In b) Z -check generators are shown as blue volumes. These can detect X -type errors.

It is important to distinguish between X - and Z -check generators. X -check generators correspond to X -stabilizers, while Z -check generators to Z -stabilizers. X -check generators can detect Z -type spacelike errors (Pauli Z errors on physical qubits) and Z -type timelike errors (readout errors on virtual qubits correspond to the measurement of X -stabilizers). Similarly Z -check generators can detect X -type errors.

The problem of error correction can be treated in the same way as I described in Sec. 3.1. I generalize the concept of error strings, from now an error string E can contain both timelike and spacelike errors. With these more general error strings Eq.(8) generalizes to

$$\mathcal{S}(E) = \partial E, \quad (16)$$

which leads to a very similar decoding problem as in the case of random Pauli errors and perfect measurements. We need to find a correction operator for the measured three-dimensional syndrome \mathcal{S} , which satisfies the following equation:

$$\partial C_{\mathcal{S}} = \mathcal{S}. \quad (17)$$

It is harder to solve this decoding problem in three dimension than in two, however, the minimum weight perfect matching works in this case as well [6].

It is important to note that in contrast to the case of random Pauli errors here Eq.(17) does not automatically ensure that the state of the surface code after error correction will be in the logical subspace. However, everything works out well in the case of properly defined logical operations as I will show in Sec. 4.

3.3 Correcting circuit-level noise

So far I have not mentioned the technical difficulties of stabilizer measurements. Usually multi-qubit X and Z parity measurements are unfeasible on physical devices. However, the stabilizer measurements can also be performed with one extra ancilla qubit per stabilizer, two-qubit gates and single-qubit measurements. The quantum circuits realizing X - and Z -stabilizer measurements are depicted in Fig.4.

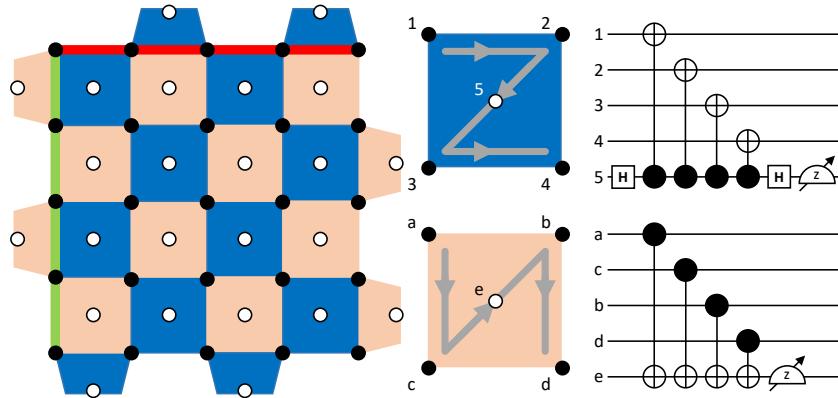


Figure 4: circuit-level implementation of a distance 5 surface code patch and stabilizer measurement circuits. Extra ancilla qubits are depicted as white dots in the middle of the faces. Stabilizer measurement circuits: CNOT gates entangle data and ancilla qubits in the right order (gray arrows), then the measurement of ancilla qubits effectively realize stabilizer measurements.

In real physical devices stabilizer measurements are realized through syndrome extraction circuits [8, 10, 12]. So an accurate modelling of noise has

to be at the circuit level. In this work I consider random Pauli errors in the syndrome extraction circuit, which can be simulated efficiently with Clifford simulations [27, 28]. A more realistic model of noise should include non-Pauli noises, such as coherent errors [29–32] and leakage [33, 34]. Although, these are typically hard to simulate on a classical computer, in some specific cases it has been shown that these are not more harmful than Pauli noise.

The circuit-level noise considered in this work contains four elements:

- **Initialization errors:** Faulty initialization of data qubits.
- **Gate errors:** Two-qubit depolarizing noise after two-qubit gates.
- **Idling errors:** Single qubit depolarizing errors on data qubits during the ancilla measurements.
- **Readout errors:** Classical bit-flips on the ancilla measurement outcomes.

Single ($\text{Dep}_1(\rho)$) and two-qubit ($\text{Dep}_2(\rho)$) depolarizing noise channels can be written in the following forms:

$$\text{Dep}_1(\rho) = (1 - p)\rho + \frac{p}{3} \sum_{P \in \{X, Y, Z\}} P\rho P; \quad (18)$$

$$\text{Dep}_2(\rho) = (1 - p)\rho + \frac{p}{15} \sum_{P \in \{I, X, Y, Z\}^{\otimes 2} / \{II\}} P\rho P. \quad (19)$$

A part of the noisy syndrome extraction circuit is shown in Fig 5.

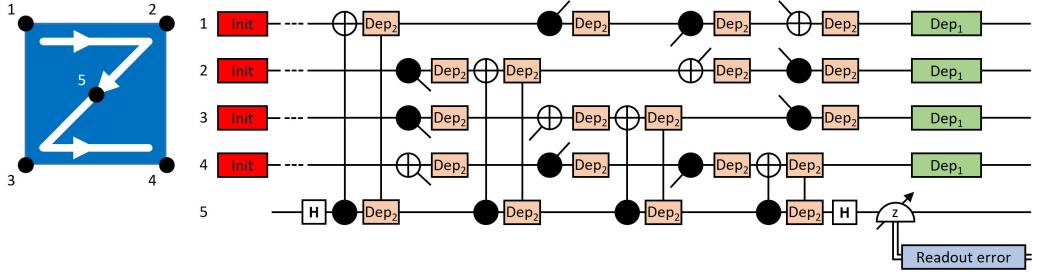


Figure 5: A noisy syndrome extraction circuit representing the measurement of a Z -stabilizer. Red "Init" boxes represent the the initialization errors (if any) at the beginning of the logical protocol. Orange "Dep₂" and green "Dep₁" boxes represent gate and idling errors. Blue "Readout error" box on the classical outcome channel represents readout errors.

Even though the above introduced circuit-level noise model contains the most important realistic noise processes, it is not specific enough to describe

a real device. However, modeling a real quantum processor with circuit-level Pauli noise is technically not harder than using this simple circuit-level model.

Surprisingly, circuit-level noise can be decoded in a similar way as phenomenological noise. The check generator values can be constructed by comparing consecutive measurement outcomes of each ancilla qubit. The identification of spacelike and timelike errors will be more difficult, because here these do not correspond uniquely to Pauli and readout errors. Hook errors [6] can also occur, which introduce diagonal edges between different check generators in the check graph. Nevertheless, the minimum weight perfect matching decoder provide a correction string based on the measured 3D syndrome \mathcal{S} in a similar way as in the case of phenomenological noise.

3.4 Minimum weight perfect matching decoder

The minimum weight perfect matching decoder [6, 23] produces a correction string for a given 2D syndrome s , or 3D syndrome \mathcal{S} , via solving the minimum weight perfect matching problem separately on the X - and Z -parts of the syndrome graph. The syndrome graph contains $s_f = -1$ (in the 2D case) or $\mathfrak{s}_f^t = -1$ (in the 3D case) points as vertices and the locations of possible error events as edges. In the 2D case these possible error event locations are the physical qubits, in 3D these are the physical and virtual qubits. Note that in the case of circuit-level noise hook errors introduce additional edges into the syndrome graph. The X - (Z -) part of the syndrome graph contains -1 values of X - (Z -) stabilizers or check generators.

The minimum weight perfect matching decoder finds the minimum weight graph that connects each vertex to an other vertex or to the boundary. This problem can be solved efficiently with Edmond's Blossom algorithm [35]. It is also possible to assign different weights to different edges based on the error probabilities. The weight of edge e is given by,

$$w_e = \log\left(\frac{1 - p_e}{p_e}\right), \quad (20)$$

where p_e is the probability of an error event on edge e . In the X -part of the syndrome graph Z and Y errors count as error events, while in Z -part of the syndrome graph X and Y do.

It is important to note that circuit-level noise models must be decomposed into graphlike error models, where each individual error event corresponds to an edge in the syndrome graph. Generally, error events can violate more than two check generators and therefore cannot be represented as edges in the check graph. Instead, they are represented as so-called hyperedges, which

connect more than two vertices. To run minimum weight perfect matching hyperedges have to be decomposed into products of simple edges, e.g., Y errors into products of X and Z errors. In this work I used Stim [36] to decompose circuit-level noise into a graphlike error model, and PyMatching [37, 38] to solve the minimum weight perfect matching problem on the syndrome graphs.

4 Surface code-based fault-tolerant logical operations

To achieve fault-tolerant quantum computation in addition to an error-corrected quantum memory, fault-tolerant logical operations are also required. In this section I present a popular approach to achieve fault-tolerant quantum computation with the surface code. Moreover, I introduce spacetime diagrams which are useful tools to represent surface code-based logical Clifford operations.

4.1 A possible route towards fault-tolerant quantum computation

A logical operation is fault tolerant if logical error rates can be arbitrarily suppressed (for small enough physical errors) by scaling up the logical protocol. Later I will provide a more precise definition, in the case of surface code-based logical operations, using the fault distance of a logical protocol. However, now it is enough to think of it as a computational scheme wherein local error mechanisms are unlikely to induce logical errors in the final output.

The primary objective is to achieve fault-tolerant universal quantum computation, which requires a universal set of quantum gates. A possible universal gateset [39] contains single qubit Clifford gates (H, S), a two-qubit entangling gate (typically CNOT) and a non-Clifford gate (usually T). While the Clifford gateset is pretty much standard, there are different approaches to achieve non-Clifford gates.

Single-qubit Clifford gates can be achieved via braiding twist-defects (corners of the surface code) [18]. The Hadamard gate can also be implemented transversely with the redefinition of stabilizers and logical operators [40]. Two-qubit Clifford gates can be realized by the so-called lattice surgery protocols [17]. In the following sections I will focus on the lattice surgery-based implementation of the logical CNOT operation.

Implementing non-Clifford gates, notably the T gate, presents challenges. A prominent method involves the distillation of magic states [16]. This technique is based on the fact that T gate can be achieved with Clifford gates and so-called magic ancillas, through gate teleportation. Magic states cannot be initialized fault-tolerantly with the surface code, so a more or less noiseless magic state has to be distilled from many noisy copies. Besides magic state distillation, numerous other approaches exist to realize universal fault-tolerant quantum computation, including the utilization of higher-dimensional codes [19], encoding information into islands of qudits [41], using gauge color codes instead of surface codes [42], and implementing non-Clifford gates through small coherent rotations [43]. The costs of these methods have to be compared with state distillation protocols, which is a highly nontrivial problem [44].

4.2 Logical identity channel and memory experiments.

To introduce the spacetime diagram representation of surface code-based logical Clifford operations, I consider three basic examples: X and Z memory experiments and the logical identity channel. In the case of the logical identity channel an arbitrary initial logical state $|\bar{\psi}\rangle$ is considered during T stabilizer measurement rounds. In the case of a Z (X) memory experiment a logical $|\bar{0}\rangle$ ($|\bar{\mp}\rangle$) state is initialized fault-tolerantly, and after T rounds of stabilizer measurements it is measured fault-tolerantly in the logical \bar{Z} (\bar{X}) basis. These three basic examples demonstrate the possible initialization and measurement setups for fault-tolerant logical operations, and thus the possible timelike boundary conditions for check graphs as well.

X - and Z -boundaries, which have been defined for a single surface code patch, can be generalized for check graphs as well – the check graphs of memory experiments and the identity channel are shown in Fig. 6. Spacelike boundaries can be defined as surfaces containing the same boundary types in each stabilizer measurement round. Equivalently, spacelike X - (Z) boundaries are vertical surfaces, where only X - (Z) check generators are present. Timelike boundaries can be defined in a similar manner, timelike X - (Z) boundaries are horizontal surfaces, where only X - (Z) check generators are present. I also define perfect timelike boundaries, where both X - and Z -check generators are present.

Lower timelike X - and Z -boundaries represent fault-tolerant initializations in the $|\bar{\mp}\rangle$ and $|\bar{0}\rangle$ states. Upper timelike X - and Z -boundaries represent fault-tolerant measurements in the logical \bar{X} and \bar{Z} basis. Perfect timelike boundaries are just theoretical constructions enabling the investigation of the

fault-tolerant logical operation itself without errors during initialization and final measurements.

Fault-tolerant initialization of the logical $|\bar{0}\rangle$ ($|\bar{+}\rangle$) state can be performed by initializing each physical quantum bit in the $|0\rangle$ ($|+\rangle$) state and measuring the stabilizers. This initialization protocol generates a lower timelike Z - (X -) boundary, because if all physical qubits are initialized in the $|0\rangle$ ($|+\rangle$) state the values of Z - (X -) stabilizers are predetermined before the first round of stabilizer measurements. Therefore, Z - (X -) check generators at this boundary contain physical qubits from the "zeroth" time step and virtual qubits from the first stabilizer measurement round. In contrast to that, the measurement outcomes of X - (Z -) stabilizers are random. Therefore, there is a missing row of X - (Z -) check generators at this boundary.

A similar argument can be made for the fault-tolerant measurement of the logical \bar{Z} (\bar{X}) operator. In this case all data qubits are measured in the Z - (X -) basis. Based on the measurement outcomes the values of Z - (X -) stabilizers (and also the value of logical \bar{Z} (\bar{X}) operator) can be reconstructed by multiplying the corresponding single qubit measurement outcomes. In this last round virtual qubits are missing, because the Z - (X -) stabilizer measurement values constructed directly from single qubit measurements, therefore, timelike errors cannot occur there. This construction generates an upper timelike Z - (X -) boundary with Z - (X -) check generators consist of 4 physical qubits from the final layer and one virtual qubit from the last real stabilizer measurement round. The X - (Z -) check generators are missing at this boundary, because X - (Z -) stabilizer values cannot be reconstructed from Z basis measurements of the data qubits.

Perfect timelike boundaries cannot be generated via a fault-tolerant protocol. However, in numerical simulations perfect boundaries can be constructed via noiseless initialization of a logical state $|\bar{\psi}\rangle$ and a final noiseless stabilizer measurement round. These noiseless initialization and measurement rounds determine the stabilizer values in the beginning and the end of the protocol. Therefore, virtual qubits are missing at the beginning and at the end of the protocol, but both check generator types are present at both timelike boundaries.

4.3 Spacetime diagrams and logical errors

Surface code-based logical Clifford operations can be visualized with spacetime diagrams [45]. These spacetime diagrams do not contain all the information of the check graph of the protocol, but contain the most important, so-called topological features such as boundaries and twist defects (separating different boundary types). The constituents of spacetime diagrams and

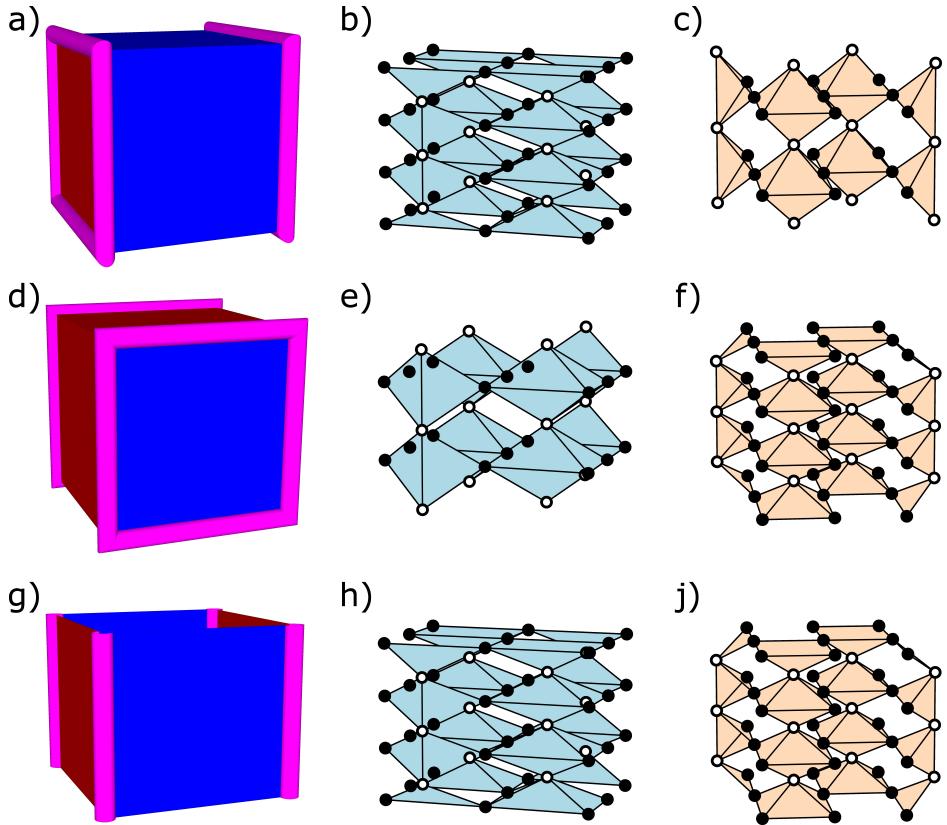


Figure 6: Spacetime diagrams and check graphs of a Z memory experiment (a,b,c), an X memory experiment (d,e,f) and a logical identity channel (g,h,j) with a distance 3 surface code and 3 stabilizer measurement rounds. a) Spacetime diagram of Z memory experiment with upper and lower timelike Z -boundaries. The corresponding Z -check generators in b) and the corresponding X -check generators in c). d) Spacetime diagram of an X memory experiment with lower and upper timelike X -boundaries, and the corresponding Z - (in e)) and X - (in f)) check generators. g) The spacetime diagram of a logical identity channel with perfect timelike boundaries, and the corresponding Z - (in h)) and X - (in j)) check generators.

their implementation are shown in Tab. 1. Spacetime diagrams representing X and Z memory experiments and the logical identity channel can be seen in Fig. 6.

| Name | Implementation | Relation to errors | Visualization |
|---------------------------------|---|---------------------------------------|-------------------|
| Bulk | Repeated measurements of X - and Z -stabilizers | Error strings cannot terminate | Free space |
| Spacelike X -boundary | Repeated measurement of X -stabilizers | X -type error strings can terminate | Red walls |
| Spacelike Z -boundary | Repeated measurement of Z -stabilizers | Z -type error strings can terminate | Blue walls |
| Lower timelike X -boundary | Initialization of qubits in $ +\rangle$ states | X -type error strings can terminate | Red walls |
| Upper timelike X -boundary | Measurement of qubits in X -basis | X -type error strings can terminate | Red walls |
| Lower timelike Z -boundary | Initialization of qubits in $ 0\rangle$ states | Z -type error strings can terminate | Blue walls |
| Upper timelike Z -boundary | Measurement of qubits in Z -basis | Z -type error strings can terminate | Blue walls |
| Perfect lower timelike boundary | Noiseless initialization of a logical state | Error strings cannot terminate | Free space |
| Perfect upper timelike boundary | Noiseless stabilizer measurement round | Error strings cannot terminate | Free space |
| Twist defects | Switch the measured stabilizer type | Separates different boundary types | Magenta cylinders |

Table 1: Summary of spacetime diagram constituents. The table provides the rules of implementation and visualization of different constituents. It also describes which error strings can terminate at a specific boundary without being detected by a check generator.

By representing logical operations with spacetime diagrams the fault-tolerant structure is inherently transparent. To see this I define logical errors as error strings (consist of both timelike and spacelike errors) that connect disconnected boundaries of the same type without violating any check generators. This definition is trivially works for error strings connecting disconnected spacelike boundaries, but it is not clear immediately why error strings that connecting connected spacelike boundaries do not cause logical errors, and how timelike boundary connections can cause logical errors. The latter cannot be seen from the previously presented logical protocols, but I will show an example for this in Sec. 5.

To understand the definition of logical errors let me consider the Z memory experiment, depicted in Fig. 6. The spacetime diagram contains two disconnected spacelike X -boundaries and just one big connected Z -boundary. Error strings connecting a spacelike Z -boundary with an other Z -boundary can bring out the code from the logical subspace or can cause a logical Z error. However, these do not count as real logical errors, because these do not change the measured value of the logical \bar{Z} operator at the end of the operation. Similar arguments can be constructed in other cases as well, and it turns out that only those connections count as logical errors that connect disconnected boundaries from the same type.

Another important concept is the so-called fault distance, which is the generalization of the code distance defined for a single patch. The fault distance of a logical protocol is the weight of the shortest logical error string. Fault distance characterizes the fault tolerance of a logical protocol, because logical errors can be suppressed (for small enough physical errors) by increasing the fault distance of the protocol.

I also note that spacetime diagrams that represent general logical protocols can have a richer repertoire of topological features [40, 45, 46]. If so-called domain walls are present, twists can have other function than just separating different boundary types, and logical error strings are a bit harder to define. Generally, spacetime diagrams can represent all surface code-based logical Clifford operation [45].

5 Fault-tolerant CNOT gate

Fault-tolerant multiqubit Clifford operations can be realized through multiqubit logical Pauli measurements. These multiqubit logical Pauli measurements can be naturally realized with the so-called lattice surgery protocols [17, 20, 47–49] for surface code-based quantum computation in a fault-tolerant manner. In this section, I introduce lattice surgery and show how

to realize measurement-based implementation of the logical CNOT gate [50], depicted in Fig.7, with lattice surgery operations.

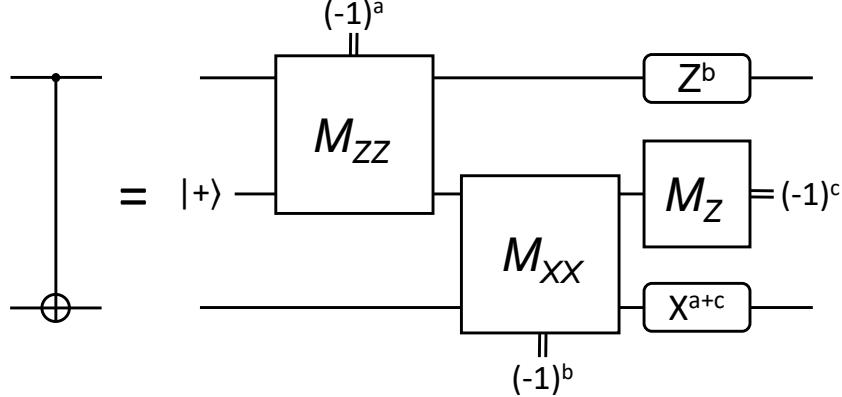


Figure 7: The measurement-based realization of the CNOT operation. This circuit realizes a CNOT gate between the control (top) and target (bottom) qubits, while an ancilla qubit is initialized in the $|+\rangle$ state and at the end of the circuit, measured in the X-basis. Based on the outcomes of mid-circuit measurements, Pauli corrections may be performed on the control and target qubits.

5.1 Lattice surgery

Lattice surgery is one of the leading approaches to implement fault-tolerant quantum computation using surface code patches. The basic two-qubit operation of lattice surgery is the measurement of the logical $\bar{Z}\bar{Z}$ or the $\bar{X}\bar{X}$ operator. In this Section I review how to perform the measurement of the $\bar{Z}\bar{Z}$ operator. The protocol describing the $\bar{X}\bar{X}$ measurement can be obtained by straightforward generalization.

Let me consider two surface code patches of distance d with X -boundaries facing each other, separated by w columns of data qubits (coupling qubits). In practical applications w can vary, depending on the physical distance of the two patches on a large quantum computer; with some definitions of the patches $w = 0$ is also feasible [47]. The steps of the protocol are the following:

1. Initialize all coupling qubits in $|+\rangle$;
2. Merge the patches, by measuring the stabilizers (multiple times) of the extended rectangular patch;
3. Measure the coupling qubits in the X -basis.

Ideally, the outcome of the $\bar{Z}\bar{Z}$ measurement is the product of the measurement outcomes of the newly defined intermediate Z -stabilizers. The three steps of the lattice surgery protocol realizing a $\bar{Z}\bar{Z}$ measurement are depicted in Fig.8.

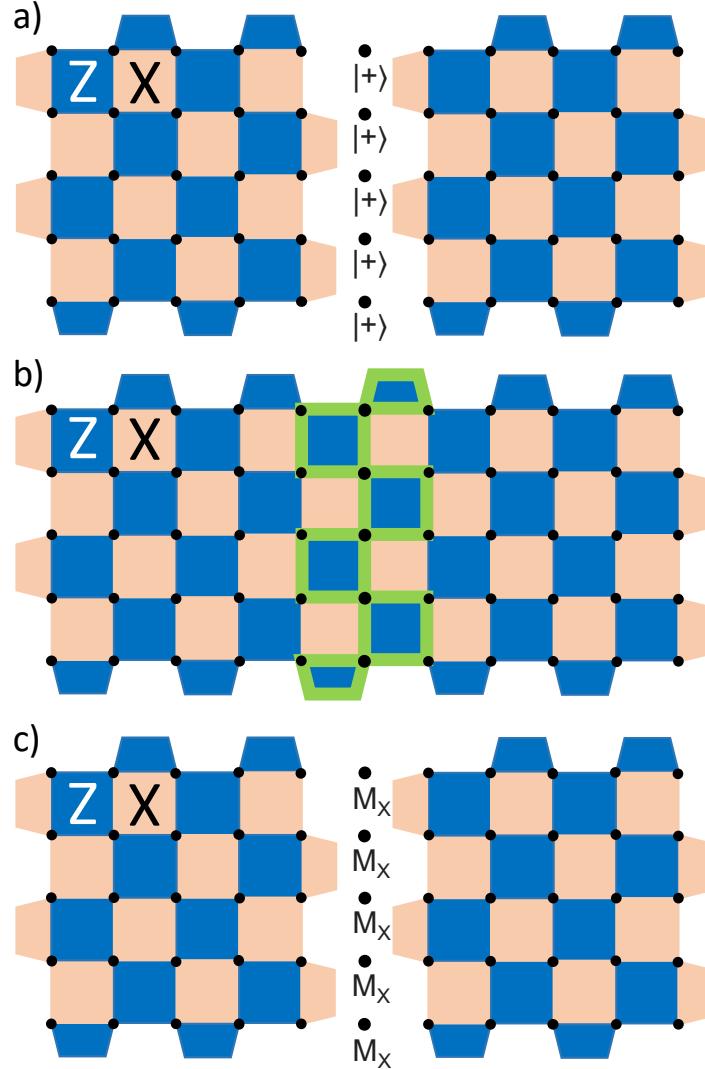


Figure 8: Three steps of a $\bar{Z}\bar{Z}$ measurement realized via lattice surgery. a) Two surface code patches with X -boundaries facing each other, and coupling qubits initialized in $|+\rangle$. b) A rectangular patch after the merge. The product of green stabilizers is the logical $\bar{Z}\bar{Z}$ operator. c) Two original patches after the split. Coupling qubits are measured in the X basis.

In contrast to memory experiments and the identity channel, in the case of

a $\bar{Z}\bar{Z}$ measurement purely timelike errors can cause logical errors. Imagine a readout error on one of the intermediate Z -stabilizers, if there is only a single round of stabilizer measurements in the merged phase, this error flips the outcome of the $\bar{Z}\bar{Z}$ measurement. In the case of h_2 rounds of stabilizer measurements h_2 readout errors are needed to cause logical readout error. In conclusion, the fault distance of the $\bar{Z}\bar{Z}$ measurement is the minimum of the distance of the underlying code patches and the number of stabilizer measurement rounds in the merged phase. In the spacetime diagram of the $\bar{Z}\bar{Z}$ measurement, see Fig.9, the fault distance is the shortest distance between disconnected boundaries of the same type, so by looking at the spacetime diagram, the above argument about the fault distance should be clear immediately.

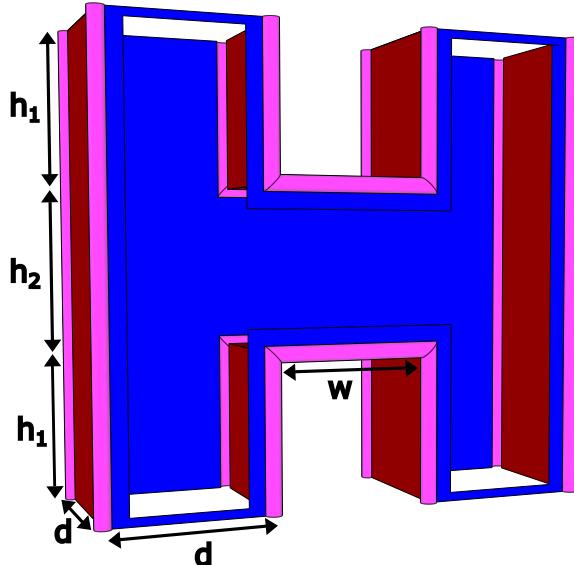


Figure 9: The spacetime diagram of the lattice surgery-based $\bar{Z}\bar{Z}$ measurement. Parameters are the linear size d (in data qubits) of each patch, and the length w of the "bridge" between them (the number of coupling qubit columns); and the number of measurement rounds h_1 before/after, and h_2 during the merged phase.

5.2 Spacetime diagram and logical error classes of the CNOT protocol

The lattice-surgery-based implementation of the logical CNOT gate consists of a $\bar{Z}\bar{Z}$ and an $\bar{X}\bar{X}$ measurement, as it is shown in Fig. 7. These multiqubit

Pauli measurements are realized with the previously described lattice surgery protocols, the spacetime diagram of the logical CNOT gate is depicted in Fig.10

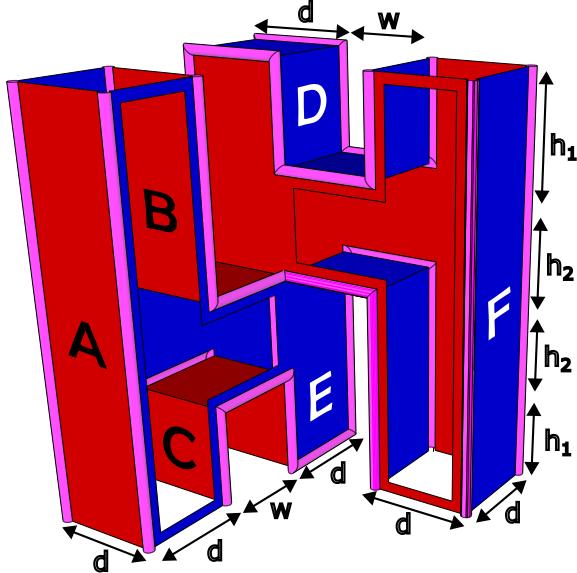


Figure 10: Spacetime diagram of the lattice-surgery-based CNOT. Control (left) and target (right) patches have perfect lower and upper timelike boundaries, ancilla patch (middle) has lower timelike X - and upper timelike Z -boundaries: initialized in $|+\rangle$, measured in Z . The X - (Z -) boundaries consist of three disconnected parts labeled A, B, C (D, E, F). Parameters are the fault distance of the protocol d , the length of the "bridges" w and the number of measurement rounds, h_1 before and after the lattice surgeries, h_2 during the two merged phases.

The spacetime diagram of the logical CNOT gate has three disconnected X -boundaries, labeled by A, B, C , and three disconnected Z - boundaries: E, F, G . Any string connecting two disconnected boundaries of the same kind leads to a Pauli error on the logical level. Since any of these errors, squared, gives the identity, there are 16 distinct connection classes based on the parity of string endings at boundaries A, B, C, D, E, F . Note that a string connecting BC is homologically equivalent [6] to a pair of strings connecting AB and BC , therefore they correspond to the same connection class, and likewise for EF, FG , and EG . These connection classes correspond to the 16 inequivalent two-qubit logical Pauli errors (including the identity) incurred during the CNOT. The correspondence between string ending parities, connection classes and logical Paulis is summarized in Table 2.

| N_A | N_B | N_C | Connection | Logical Pauli |
|-------|-------|-------|-------------------|---------------|
| 0 | 0 | 0 | \emptyset_{ABC} | 1 |
| 1 | 1 | 0 | A - B | X_1 |
| 1 | 0 | 1 | A - C | X_1X_2 |
| 0 | 1 | 1 | B - C | X_2 |
| N_D | N_E | N_F | Connection | Logical Pauli |
| 0 | 0 | 0 | \emptyset_{DEF} | 1 |
| 1 | 1 | 0 | D - E | Z_1 |
| 1 | 0 | 1 | D - F | Z_2 |
| 0 | 1 | 1 | E - F | Z_1Z_2 |

Table 2: Correspondence between logical Pauli errors and connection classes of the boundaries. First column: the number of error string endings modulo 2 at each boundary specifies the connection class – the sum of every row must be even, since each string has two ends. Second column: The minimally required connections between boundaries in each class, \emptyset_{ABC} (\emptyset_{DEF}) signals no connections between X - (Z -) boundaries. Third column: the corresponding logical Paulis. The 16 logical Pauli error classes can be obtained by multiplying 1 row from the top and 1 from the bottom half of the table.

6 Simulation methods

The simulation of quantum error correction during a logical CNOT gate requires different approaches for different error models. In the case of phenomenological errors I used the so-called Pauli frame simulation, while in the case of circuit-level noise I used Stim [36], which is an efficient tool for Clifford circuit simulations.

6.1 Phenomenological errors and Pauli frame simulation

To simulate phenomenological noise I constructed the check graph of the CNOT gate consists of physical qubits, virtual qubits and check generators. Instead of storing the quantum state of the system it is enough to monitor the error history of physical and virtual qubits. A Pauli frame can be associated to each qubit in each time step, which records whether X,Y,Z or no error occurred on the qubit during the corresponding step. This method allows me to store the error strings in an n -element vector, where n is the total number of physical and virtual qubits. Each element can be 0,1,2, or 3, so reserve 2

bits. It is important to note that for virtual qubits only one type of error is possible, so these qubits require only one bit per step.

For a given error string E , calculating the values of the check generators, thus the corresponding three-dimensional syndrome ∂E , is straightforward. We need to check the parity of the overlapping qubits between the error string and each check generator. To decode the three-dimensional syndrome ∂E and get a correction operator $C_{\partial E}$, I constructed the syndrome graph and ran minimum weight perfect matching decoder on it as it is implemented in PyMatching [37, 38]. By checking the parity of error + correction string endings at each boundary I identified a logical Pauli error based on Tab. 2.

To obtain probabilities for each logical Pauli error I sampled error strings from two inequivalent error distributions a.k.a. error models.

The independent error model contains independent and equally probable X and Z errors. This model is described by the following single qubit error channel:

$$\begin{aligned}\varepsilon(\rho) &= (1-p)^2\rho + p(1-p)(X\rho X + Z\rho Z) + p^2Y\rho Y; \\ q &= p,\end{aligned}\tag{21}$$

where q is the probability of readout errors.

The other phenomenological error model, the depolarizing error model is described by the following single qubit error channel and readout error strength:

$$\begin{aligned}\varepsilon(\rho) &= (1-p)\rho + \frac{p}{3}(X\rho X + Z\rho Z + Y\rho Y); \\ q &= \frac{2}{3}p.\end{aligned}\tag{22}$$

In each model I set q to a value that ensures equal timelike and spacelike error rates, thus the physical error rate p fully specifies independent and depolarizing errors.

6.2 Circuit-level noise and process tomography

Syndrome extraction circuits of surface code-based logical operations are Clifford circuits, therefore, in the case of Pauli errors (more generally Clifford errors) these circuits are efficiently simulatable due to the well-known Gottesman-Knill theorem [27]. In this work I used Stim [36] an efficient Clifford simulator to simulate circuit-level noise during the logical CNOT gate.

I considered a circuit-level noise model with a single parameter p (physical error rate), which is the strength of both initialization, gate, idling and

readout errors. I used minimum weight perfect matching to decode the three-dimensional syndrome in this case as well.

The determination of logical Pauli error probabilities is a little bit more involved in the case of circuit-level noise. In contrast to phenomenological errors here, I could not investigate the check graphs, thus the error + correction string endings directly; instead I had to run logical quantum circuits with different initial states and final measurements and perform a process tomography for the logical CNOT gate. Note that the timelike boundaries for control and target patches were still perfect boundaries, I just needed to specify the perfectly initialized logical states and the error-free final measurements. The initial states and the final measurements, determining the logical circuit, are summarized in Table 3.

| Initial state | Final measurements | Obtained probabilities |
|--------------------------|------------------------|--|
| $ \bar{0}\bar{0}\rangle$ | \bar{Z}_1, \bar{Z}_2 | $P_{00}^{ZZ}(-,+), P_{00}^{ZZ}(+,-), P_{00}^{ZZ}(-,-)$ |
| $ \bar{0}+\rangle$ | \bar{Z}_1, \bar{X}_2 | $P_{0+}^{ZX}(-,+), P_{0+}^{ZX}(+,-), P_{0+}^{ZX}(-,-)$ |
| $ \bar{0}i\rangle$ | \bar{Z}_1, \bar{Y}_2 | $P_{0i}^{ZY}(-,+), P_{0i}^{ZY}(+,-), P_{0i}^{ZY}(-,-)$ |
| $ i+\rangle$ | \bar{Y}_1, \bar{X}_2 | $P_{i+}^{YX}(-,+), P_{i+}^{YX}(+,-), P_{i+}^{YX}(-,-)$ |
| $ ++\rangle$ | \bar{X}_1, \bar{X}_2 | $P_{++}^{XX}(-,+), P_{++}^{XX}(+,-), P_{++}^{XX}(-,-)$ |
| $ \bar{+}\bar{0}\rangle$ | $\bar{Y}_1 \bar{Y}_2$ | $P_{+0}^{YY}(-)$ |
| $ \bar{+}i\rangle$ | $\bar{Y}_1 \bar{Z}_2$ | $P_{+i}^{YZ}(-)$ |
| $ i\bar{0}\rangle$ | $\bar{X}_1 \bar{Y}_2$ | $P_{i0}^{XY}(-)$ |
| $ i\bar{i}\rangle$ | $\bar{X}_1 \bar{Z}_2$ | $P_{ii}^{XZ}(-)$ |

Table 3: Initial logical states and final measurement bases for the process tomography of the logical CNOT gate. A comma separating final measurement bases, e.g., \bar{Z}_1, \bar{X}_2 , means that both measurements are deterministic, therefore, logical errors can cause 3 different measurement outcomes $((-,+),(+,-),(-,-))$. Non-separated final measurement bases, e.g., $\bar{Y}_1 \bar{Y}_2$ means that only the product of the measurement outcome is deterministic, therefore, the only erroneous case is when the measurement outcomes are not the same $(-)$. The probabilities of the erroneous cases are shown in the third column.

The outcomes of the final measurements in logical quantum circuits summarized in Table 3 are deterministic in the noiseless case, therefore, by sampling the logical quantum circuits we gain information about the logical noise. The probabilities, which are directly obtained from sampling the logical cir-

cuits ($P_{00}^{ZZ}(-,+)$, $P_{00}^{ZZ}(+,-)$, ...) can be expressed with the probabilities of logical Pauli errors (p_{IX}, p_{IY}, \dots) with the following system of independent equations:

$$\begin{pmatrix} P_{00}^{ZZ}(-,+) \\ P_{00}^{ZZ}(+,-) \\ P_{00}^{ZZ}(-,-) \\ P_{0+}^{ZX}(-,+) \\ P_{0+}^{ZX}(+,-) \\ P_{0i}^{ZY}(-,+) \\ P_{0i}^{ZY}(+,-) \\ P_{i+}^{YX}(-,+) \\ P_{i+}^{YX}(+,-) \\ P_{++}^{XX}(-,+) \\ P_{++}^{XX}(+,-) \\ P_{+0}^{YY}(-) \\ P_{+i}^{YZ}(-) \\ P_{i0}^{XY}(-) \\ P_{ii}^{XZ}(-) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p_{IX} \\ p_{IY} \\ p_{IZ} \\ p_{XI} \\ p_{XX} \\ p_{XY} \\ p_{XZ} \\ p_{YI} \\ p_{YX} \\ p_{YY} \\ p_{YZ} \\ p_{ZI} \\ p_{ZX} \\ p_{ZY} \\ p_{ZZ} \end{pmatrix}. \quad (23)$$

By solving this system of equations one can uniquely determine the probabilities of logical Pauli errors. Note that in Eq. 23 I did not use all the 19 probabilities which can be determined by sampling the logical circuits summarized in Table 3. The remaining 4 probabilities give us 4 non-independent equations.

7 The structure of the logical noise channel

In this section I show that the logical CNOT protocol has a non-trivial symmetry transformation, which results in a special logical error channel structure. Moreover, I show that in the case of independent phenomenological noise the logical error probabilities factorize, therefore, the logical error channel can be described with just 3 logical error parameters.

The numerical data show that the logical error channel displays the symmetry structure for both phenomenological and circuit-level noise. This could be surprising, since in the case of circuit-level noise errors at the boundaries weakly break the symmetry of the protocol. More interestingly, I found that the logical error channel can be described with 3 logical error parameters in all cases, which signals that the correlation between X and Z errors does not appear at the logical level.

7.1 Symmetry transformation of the CNOT protocol

The CNOT protocol has a symmetry, evident from the spacetime diagram in Fig. 11, which constrains the possible logical error rates. The symmetry is composed of:

1. A time reversal;
2. A mirror reflection on the vertical symmetry plane of the "L" shape;
3. Flipping the X and Z labels.

As a result the original check graph effectively does not change. The order of the three steps is interchangeable.

If this symmetry transformation also leaves the syndrome graph invariant, the logical Pauli error strings, which transform into each other under the symmetry, have to have the same probabilities. So not only the check graph, but also the spatial distribution of error probabilities have to be invariant (which holds for phenomenological, but does not hold for circuit-level noise). As a consequence, symmetry partners among logical Pauli errors have the same probabilities, which result in a two-qubit logical error channel shown in Fig. 12.

For circuit-level noise the symmetry is not exact. Namely, the error probability distribution on the check graph do not transform into itself under the symmetry transformation. This is because probability of initialization errors and final Pauli errors on data qubits (composed of gate errors, idling errors and measurement errors) at non-perfect timelike boundaries are different. This difference weakly breaks the symmetry at non-perfect timelike

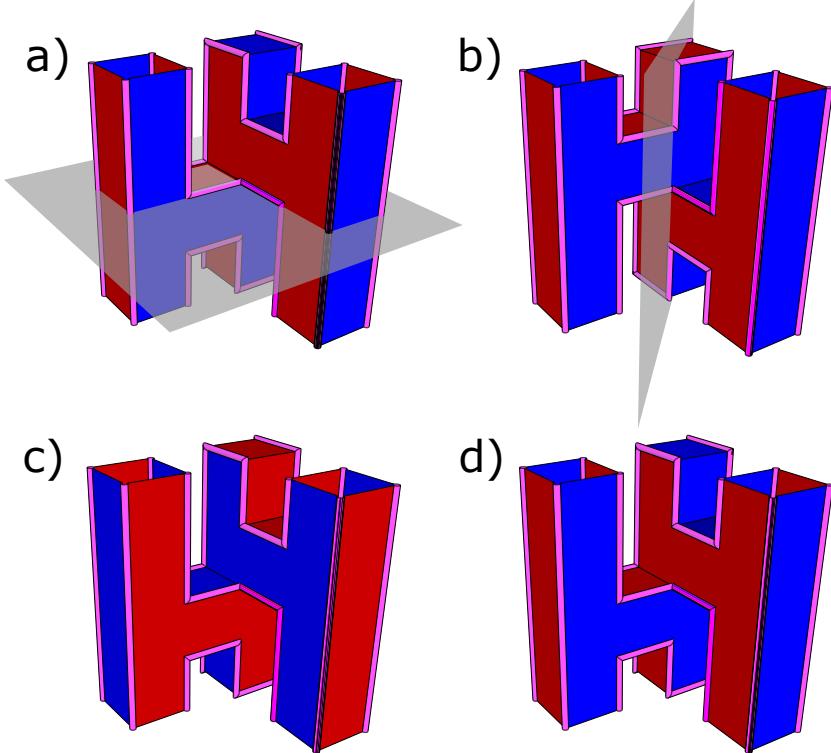


Figure 11: Symmetry of the spacetime diagram of the logical CNOT gate. The check graph has a symmetry transformation which consists of the following steps: 1) time reversal (a), 2) mirror reflection (b), 3) swapping the X and Z labels (colors) (c).

boundaries, thus I expect that the probabilities of symmetry partners will slightly differ as well.

In the case of independent X and Z errors the probabilities of the different error classes factorize. Therefore, if the physical X and Z errors have equal rate all logical error class probabilities can be written using 3 logical error parameters. These parameters are the probabilities of the 3 possible classes of connections of different boundaries of one kind via error + correction strings:

$$p_1 = \mathbb{P}(A - C) = \mathbb{P}(E - F); \quad (24a)$$

$$p_2 = \mathbb{P}(B - C) = \mathbb{P}(D - E); \quad (24b)$$

$$p_3 = \mathbb{P}(A - B) = \mathbb{P}(D - F), \quad (24c)$$

using the notation of Table 2. The probability that all Z -boundaries have an even number of Z -string endings (equal to the corresponding probability

| | | | | |
|-------|----------|----------|----------|----------|
| I_2 | p_0^2 | p_3p_0 | p_2p_3 | p_2p_0 |
| X_2 | p_2p_0 | p_1p_0 | p_1p_2 | p_2^2 |
| Y_2 | p_1p_2 | p_1^2 | p_1p_3 | p_2p_3 |
| Z_2 | p_1p_0 | p_1p_3 | p_3^2 | p_3p_0 |

$I_1 \quad X_1 \quad Y_1 \quad Z_1$

Figure 12: The structure of the two-qubit logical error channel of the lattice-surgery-based CNOT protocol under independent phenomenological noise. Theoretical expectation for the two-qubit logical Pauli error probabilities based on the connection parameters p_1, p_2, p_3 ; non-white colors indicate symmetry partners.

for all X -boundaries) can also be written using p_1 , p_2 , and p_3 , as

$$p_0 = \mathbb{P}(\emptyset_{ABC}) = \mathbb{P}(\emptyset_{DEF}) = 1 - p_1 - p_2 - p_3. \quad (25)$$

The probabilities of logical Pauli errors can be expressed with the probabilities p_1, p_2 and p_3 . I indicated these expressions in Fig. 12.

7.2 Numerical results for the logical noise channel

To see how the symmetry of the CNOT protocol manifests in the logical error channel, I ran extensive numerical simulations, as it has been described in Sec. 6. The codes and the full numerical data of phenomenological noise simulations are available at [51]. Three examples of numerically obtained error channels are shown in Fig. 13. These results show that the logical error channels display the symmetry structure of Fig. 12, namely the error rates of symmetry partners are the same up to good precision.

In order to see how the theoretically predicted structure of the logical noise channel appears in the numerically obtained data, I plotted all the logical Pauli error probabilities for a wide range of physical error rates, as shown in Fig. 14. Moreover, to see the factorization of error probabilities, for each p value I determined the closest error channel that fulfills the theoretically predicted form of Fig. 12. I did this by plotting logical error probabilities as the function of two-qubit Pauli operators and fitted a function to these

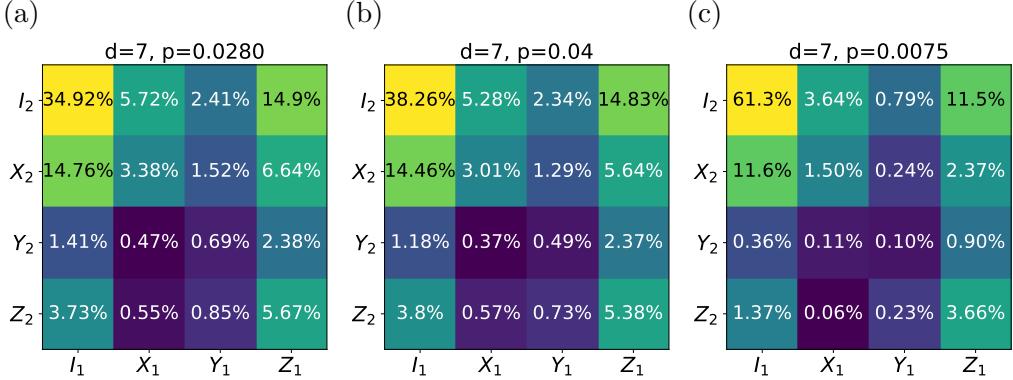


Figure 13: Numerically obtained logical error channels for distance 7 CNOT gates. (a) independent, (b) depolarizing and (c) circuit-level noise. Other parameters were $h_1 = 1$ and $w = 1$

points in the form of Fig. 12. So first I considered the numerically obtained data points:

$$y(P_1 P_2) = \mathbb{P}(P_1 P_2), \quad P_1, P_2 \in \{I, X, Y, Z\}, \quad (26)$$

then fitted the following function to these data points:

$$f_{p_1, p_2, p_3}(P_1 P_2) = \begin{cases} p_0^2 & \text{if } P_1 P_2 = II; \\ p_3 p_0 & \text{if } P_1 P_2 = XI \text{ or } P_1 P_2 = ZZ; \\ p_2 p_3 & \text{if } P_1 P_2 = YI \text{ or } P_1 P_2 = ZY; \\ p_2 p_0 & \text{if } P_1 P_2 = ZI \text{ or } P_1 P_2 = IX; \\ p_1 p_0 & \text{if } P_1 P_2 = XX \text{ or } P_1 P_2 = IZ; \\ p_1 p_2 & \text{if } P_1 P_2 = YX \text{ or } P_1 P_2 = IY; \\ p_2^2 & \text{if } P_1 P_2 = ZX; \\ p_1^2 & \text{if } P_1 P_2 = XY; \\ p_1 p_3 & \text{if } P_1 P_2 = YY \text{ or } P_1 P_2 = XZ; \\ p_3^2 & \text{if } P_1 P_2 = YZ. \end{cases} \quad (27)$$

From the fit I determined the parameters p_1 , p_2 , and p_3 , and their errors. In Fig. 14 I plotted all different values of $f_{p_1, p_2, p_3}(P_1 P_2)$ for each p (physical error rate) value as dashed lines.

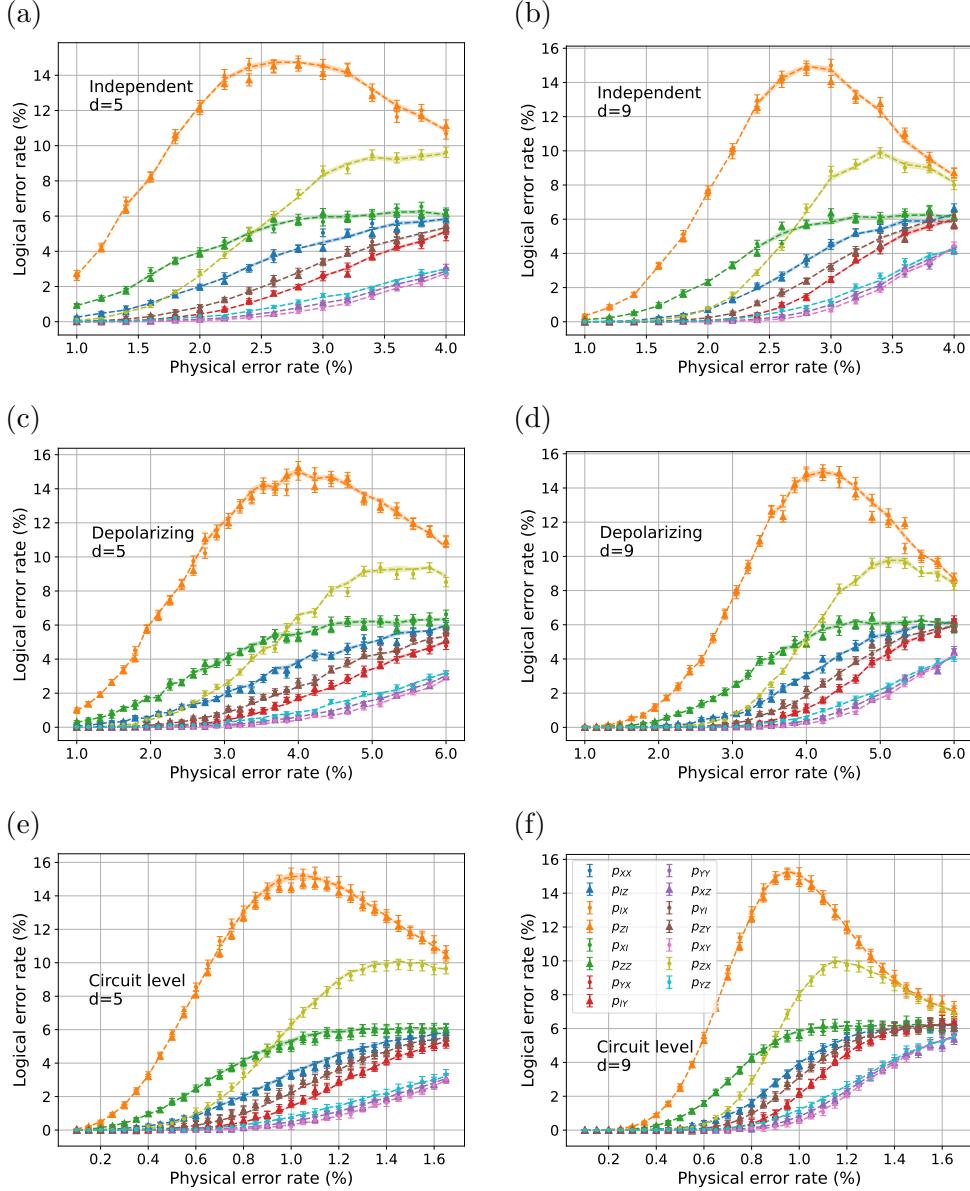


Figure 14: Logical Pauli error rates in CNOT as a function of the physical error rates for patch size $d = 5$ and 9 , with (a) and (b) independent errors, (c) and (d) depolarizing errors, (e) and (f) circuit-level noise. Numerically obtained error rates are shown with symbols with errorbars. Rates of symmetry partners have the same color; they are in most cases equal to good precision. Semi-theoretical expectations for these rates, calculated from p_{XY} , p_{ZX} , and p_{YZ} , as described in the main text, are shown with dashed lines, with shading indicating expected standard deviation. Each point has been created by 10^4 (phenomenological) or 10^5 (circuit level) Monte Carlo rounds.

The numerical results in Fig. 14 show that both the symmetry structure and the probability factorization are present for all types of noises. Therefore, the logical noise channel predicted in Fig. 12 really describes the logical noise of the CNOT gate. This is a quite surprising result, because circuit-level noise weakly breaks the symmetry of the protocol at non-perfect timelike boundaries. Moreover, X and Z errors are correlated in both depolarizing and circuit-level noise models. I suspect that the symmetry structure is robust against weak symmetry breaking at the boundaries, because for large fault distances the bulk dominates the noise. However, the reason for the robustness of the probability factorization is probably the consequence of the independent decoding of the noise [1].

7.3 Threshold behaviour

So far I have not mentioned one of the most interesting phenomena related to surface code-based quantum error correction, the threshold behaviour. It is known [5–7, 26] that the surface code, when used as a quantum memory, has a quantum error correction threshold: There is a critical physical error rate, e.g., $p_{\text{th}} \approx 3\%$ for independent phenomenological noise [7], below which the logical error rates can be suppressed exponentially by increasing the code distance. The value of this error threshold depends on the error model and also the decoder.

Any lattice-surgery-based logical quantum circuit should also have an error threshold, with the same value as the threshold defined above. Thus, when scaling up the size (fault distance) of a logical circuit, the logical error rates should decrease whenever the physical error rates are below threshold. This statement is supported by the fact that in most cases, the decoding procedure can be mapped to a classical statistical mechanical model [6, 7, 52], in which the critical point corresponds to the threshold value. The value of the critical point in the $d \rightarrow \infty$ limit should not depend on the boundaries, therefore, the threshold should be the same for the lattice-surgery-based operations and quantum memory experiments. This was also observed for several logical protocols under phenomenological noise [45]. Note that the mapping of circuit-level noise to classical statistical mechanical models is not worked out for the surface code, however, it exists for other codes [53].

I calculated numerically the threshold of the lattice-surgery-based CNOT gate for independent, depolarizing and circuit-level noise, and found good agreement with the known threshold of the surface code, as shown in Fig. 15. (For circuit-level noise I checked the threshold with my own simulations.)

Although the threshold value is promising for fault-tolerant quantum

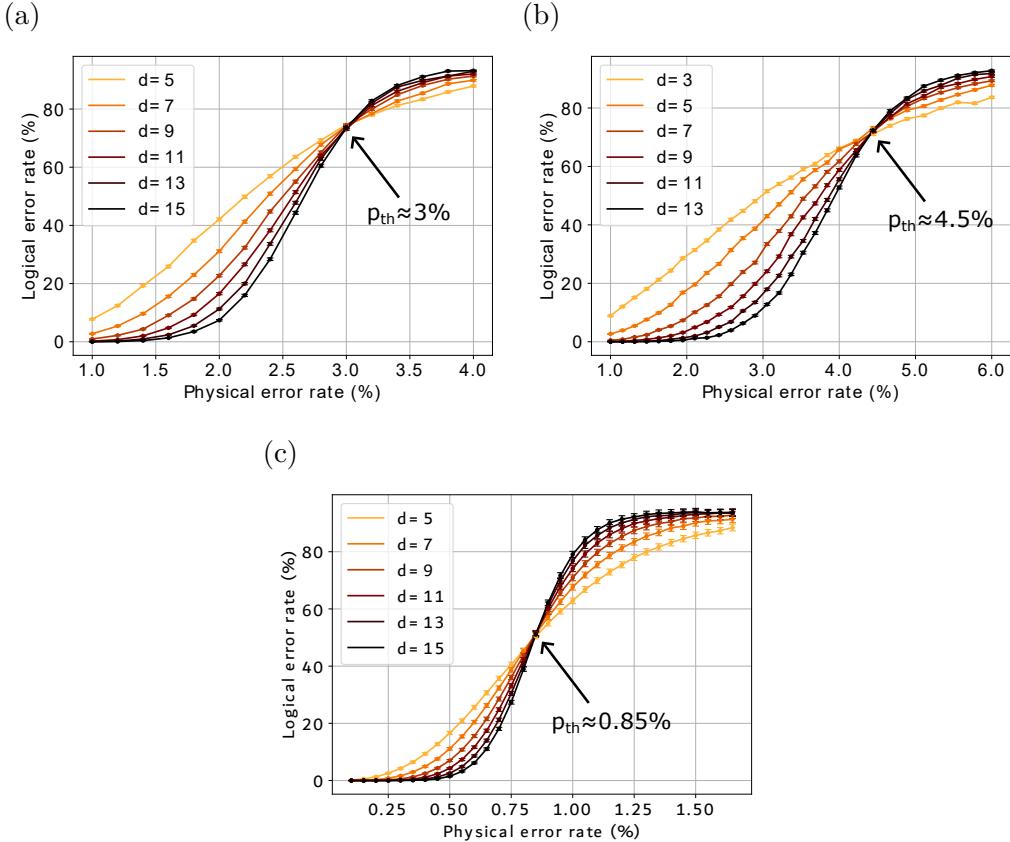


Figure 15: The logical error rates ($1 - p_{II}$) for different fault distances (d) as the function of physical error rate for (a) independent, (b) depolarizing and (c) circuit-level noise. The threshold values are depicted for each case. Other parameters were $h_2 = d$, $h_1 = w = 1$. Each point has been created by 10^4 (phenomenological) or 10^5 (circuit level) Monte Carlo rounds.

computation, the error rates here are much higher than in a simple memory experiment. This is due to the higher number of possible error classes.

8 Conclusion

In this thesis I investigated a fault-tolerant lattice-surgery-based CNOT gate between two surface code patches. I fully characterized the logical noise channel of the CNOT gate under three different physical noise models: independent noise, depolarizing noise and circuit-level noise. I found that the symmetry of the logical CNOT protocol appears in the two-qubit logical noise

channel. The logical Pauli errors which transform into each other under the symmetry transformation occur with equal probabilities. Moreover, the X and Z logical error probabilities factorize, so the logical error channel can be described with just three logical error parameters. Strictly speaking these statements are only hold for independent phenomenological noise, however, I found that the statements remain true (up to good numerical precision) in the case of depolarizing and circuit-level noise as well.

I also determined the error correction threshold of the logical CNOT operation under independent, depolarizing, and circuit-level noise, and found a good agreement with the quantum memory thresholds.

Further investigation of error models that break the symmetry of the protocol would be interesting. For example biased noise (when X and Z error rates differ) obviously breaks the symmetry, however, symmetrizing the noise with e.g. using XZZX surface code [54] maybe restore the symmetry in the logical noise channel. An other interesting direction would be the generalization of this symmetry argument for other lattice-surgery-based logical operations.

I believe that the findings in this thesis provide a deeper understanding of the structure of the logical noise in lattice-surgery-based logical operations. Therefore, they help to model the noise in fault-tolerant quantum algorithms and also provide useful insights for near-term lattice surgery experiments.

References

- [1] Bálint Domokos, Áron Márton, and János K. Asbóth. Characterization of errors in a cnot between surface code patches, 2024.
- [2] Craig Gidney and Martin Ekerå. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. *Quantum*, 5:433, April 2021.
- [3] A.Yu. Kitaev. Fault-tolerant quantum computation by anyons. *Annals of Physics*, 303(1):2–30, January 2003.
- [4] S. B. Bravyi and A. Yu. Kitaev. Quantum codes on a lattice with boundary, 1998.
- [5] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. Surface codes: Towards practical large-scale quantum computation. *Physical Review A*, 86(3), September 2012.

- [6] Eric Dennis, Alexei Kitaev, Andrew Landahl, and John Preskill. Topological quantum memory. *Journal of Mathematical Physics*, 43(9):4452–4505, September 2002.
- [7] 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, January 2003.
- [8] Rajeev Acharya et al. Suppressing quantum errors by scaling a surface code logical qubit. *Nature*, 614:676 – 681, 2022.
- [9] Dolev Bluvstein, Harry Levine, Giulia Semeghini, Tout T. Wang, Sepehr Ebadi, Marcin Kalinowski, Alexander Keesling, Nishad Maskara, Hannes Pichler, Markus Greiner, Vladan Vuletić, and Mikhail D. Lukin. A quantum processor based on coherent transport of entangled atom arrays. *Nature*, 604(7906):451–456, April 2022.
- [10] Sebastian Krinner et al. Realizing repeated quantum error correction in a distance-three surface code. *Nature*, 605(7911):669–674, 2022.
- [11] Youwei Zhao et al. Realization of an error-correcting surface code with superconducting qubits. *Physical Review Letters*, 129(3), July 2022.
- [12] Dolev Bluvstein et al. Logical quantum processor based on reconfigurable atom arrays. *Nature*, December 2023.
- [13] C. Ryan-Anderson et al. High-fidelity and fault-tolerant teleportation of a logical qubit using transversal gates and lattice surgery on a trapped-ion quantum computer, 2024.
- [14] Riddhi S. Gupta et al. Encoding a magic state with beyond break-even fidelity. *Nature*, 625(7994):259–263, January 2024.
- [15] Bence Hetényi and James R. Wootton. Creating entangled logical qubits in the heavy-hex lattice with topological codes, 2024.
- [16] Sergey Bravyi and Alexei Kitaev. Universal quantum computation with ideal Clifford gates and noisy ancillas. *Physical Review A*, 71(2), February 2005.
- [17] Dominic Horsman, Austin G Fowler, Simon Devitt, and Rodney Van Meter. Surface code quantum computing by lattice surgery. *New Journal of Physics*, 14(12):123011, December 2012.

- [18] Benjamin J. Brown, Katharina Laubscher, Markus S. Kesselring, and James R. Wootton. Poking holes and cutting corners to achieve Clifford gates with the surface code. *Physical Review X*, 7(2), May 2017.
- [19] Benjamin J. Brown. A fault-tolerant non-Clifford gate for the surface code in two dimensions. *Science Advances*, 6(21):eaay4929, 2020.
- [20] Daniel Litinski. A game of surface codes: Large-scale quantum computing with lattice surgery. *Quantum*, 3:128, March 2019.
- [21] H. Bombin and M. A. Martin-Delgado. Optimal resources for topological two-dimensional stabilizer codes: Comparative study. *Physical Review A*, 76(1), July 2007.
- [22] Sergey Bravyi, Martin Suchara, and Alexander Vargo. Efficient algorithms for maximum likelihood decoding in the surface code. *Physical Review A*, 90:032326, September 2014.
- [23] Austin G. Fowler. Minimum weight perfect matching of fault-tolerant topological quantum error correction in average $o(1)$ parallel time, 2014.
- [24] Nicolas Delfosse and Naomi H. Nickerson. Almost-linear time decoding algorithm for topological codes. *Quantum*, 5:595, December 2021.
- [25] Oscar Higgott, Thomas C. Bohdanowicz, Aleksander Kubica, Steven T. Flammia, and Earl T. Campbell. Improved decoding of circuit noise and fragile boundaries of tailored surface codes. *Physical Review X*, 13:031007, Jul 2023.
- [26] D. S. Wang, A. G. Fowler, A. M. Stephens, and L. C. L. Hollenberg. Threshold error rates for the toric and planar codes. *Quantum Info. Comput.*, 10(5):456–469, May 2010.
- [27] Daniel Gottesman. The heisenberg representation of quantum computers, 1998.
- [28] Scott Aaronson and Daniel Gottesman. Improved simulation of stabilizer circuits. *Physical Review A*, 70:052328, Nov 2004.
- [29] Sergey Bravyi, Matthias Englbrecht, Robert König, and Nolan Peard. Correcting coherent errors with surface codes. *npj Quantum Information*, 4(1), October 2018.
- [30] Áron Márton and János K. Asbóth. Coherent errors and readout errors in the surface code. *Quantum*, 7:1116, September 2023.

- [31] Dávid Pataki, Áron Márton, János K. Asbóth, and András Pályi. Coherent errors in stabilizer codes caused by quasistatic phase damping, 2024.
- [32] Stefanie J. Beale, Joel J. Wallman, Mauricio Gutiérrez, Kenneth R. Brown, and Raymond Laflamme. Quantum error correction decoheres noise. *Physical Review Letters*, 121:190501, Nov 2018.
- [33] Martin Suchara, Andrew W. Cross, and Jay M. Gambetta. Leakage suppression in the toric code. *2015 IEEE International Symposium on Information Theory (ISIT)*, pages 1119–1123, 2014.
- [34] Kevin C. Miao et al. Overcoming leakage in quantum error correction. *Nature Physics*, 19:1780 – 1786, 2022.
- [35] Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.
- [36] Craig Gidney. Stim: a fast stabilizer circuit simulator. *Quantum*, 5:497, July 2021.
- [37] Oscar Higgott. Pymatching: A python package for decoding quantum codes with minimum-weight perfect matching. *ACM Transactions on Quantum Computing*, 3(3):1–16, 2022.
- [38] Oscar Higgott and Craig Gidney. Sparse blossom: correcting a million errors per core second with minimum-weight matching. *arXiv:2303.15933*, 2023.
- [39] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.
- [40] György P. Gehér, Campbell McLaughlan, Earl T. Campbell, Alexandra E. Moylett, and Ophelia Crawford. Error-corrected Hadamard gate simulated at the circuit level, 2023.
- [41] Katharina Laubscher, Daniel Loss, and James R. Wootton. Universal quantum computation in the surface code using non-Abelian islands. *Physical Review A*, 100(1), July 2019.
- [42] Héctor Bombín. Gauge color codes: optimal transversal gates and gauge fixing in topological stabilizer codes. *New Journal of Physics*, 17(8):083002, 2015.

- [43] Hyeongrak Choi, Frederic T. Chong, Dirk Englund, and Yongshan Ding. Fault tolerant non-Clifford state preparation for arbitrary rotations. *arXiv:2303.17380*, 2023.
- [44] Michael E. Beverland, Aleksander Kubica, and Krysta M. Svore. Cost of universality: A comparative study of the overhead of state distillation and code switching with color codes. *PRX Quantum*, 2(2), June 2021.
- [45] Héctor Bombín, Chris Dawson, Ryan V. Mishmash, Naomi Nickerson, Fernando Pastawski, and Sam Roberts. Logical blocks for fault-tolerant topological quantum computation. *PRX Quantum*, 4(2), April 2023.
- [46] Craig Gidney. Inplace access to the surface code y basis. *Quantum*, 8:1310, April 2024.
- [47] Daniel Litinski and Felix von Oppen. Lattice surgery with a twist: Simplifying Clifford gates of surface codes. *Quantum*, 2:62, May 2018.
- [48] Christopher Chamberland and Earl T. Campbell. Universal quantum computing with twist-free and temporally encoded lattice surgery. *PRX Quantum*, 3(1), February 2022.
- [49] Christopher Chamberland and Earl T. Campbell. Circuit-level protocol and analysis for twist-based lattice surgery. *Physical Review Research*, 4(2), May 2022.
- [50] CWJ Beenakker, DP DiVincenzo, C Emery, and M Kindermann. Charge detection enables free-electron quantum computation. *Physical Review Letters*, 93(2):020501, 2004.
- [51] Numerical data and Python codes. <https://doi.org/10.5281/zenodo.11079788>.
- [52] H. Bombin, Ruben S. Andrist, Masayuki Ohzeki, Helmut G. Katzgraber, and M. A. Martin-Delgado. Strong resilience of topological codes to depolarization. *Physical Review X*, 2(2), April 2012.
- [53] Davide Vodola, Manuel Rispler, Seyong Kim, and Markus Müller. Fundamental thresholds of realistic quantum error correction circuits from classical spin models. *Quantum*, 6:618, January 2022.
- [54] J. Pablo Bonilla Ataides, David K. Tuckett, Stephen D. Bartlett, Steven T. Flammia, and Benjamin J. Brown. The XZZX surface code. *Nature Communications*, 12(1), April 2021.