Architectural analysis of universal concatenated quantum codes

Christopher Chamberland,1,∗ Tomas Jochym-O’Connor,1,† and Raymond Laflamme1,2,3

1Institute for Quantum Computing and Department of Physics and Astronomy, University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada
2Perimeter Institute, Waterloo, Ontario, N2L 2Y5, Canada
3Canadian Institute For Advanced Research, Toronto, Ontario, M5G 1Z8, Canada

We analyze the resource overhead of recently proposed methods for universal fault-tolerant quantum computation using concatenated codes. Namely, we examine the concatenation of the 7-qubit Steane code with the 15-qubit Reed-Muller code, which allows for the construction of the 49 and 105-qubit codes that do not require the need for magic state distillation for universality. We compute a lower bound for the adversarial noise threshold of the 105-qubit code and find it to be $8.33 \times 10^{-6}$. We obtain a depolarizing noise threshold for the 49-qubit code of $9.69 \times 10^{-4}$ which is competitive with the 105-qubit threshold result of $1.28 \times 10^{-3}$. We then provide lower bounds on the resource requirements of the 49 and 105-qubit codes and compare them with the surface code implementation of a logical $T$ gate using magic state distillation. For the sampled input error rates and noise model, we find that the surface code achieves a smaller overhead compared to our concatenated schemes.

PACS numbers: 03.67.Pp

I. INTRODUCTION

Fault-tolerant computations provide a means to control and suppress error rates to arbitrarily low levels, without a detrimental overhead in terms of the number of qubits and computation time. However, estimating the additional resources that would be required for such computations is an important area of study as physical architectures begin to approach the realms of scalability [1–5]. As such, it has become increasingly important to evaluate whether different architectures have particular advantages over one another with respect to targeting the implementation of a given algorithm with required accuracy.

The early proposals for fault-tolerant architectures were given by concatenated quantum error correcting codes, where qubits forming an error correcting code are re-encoded for further protection. This provides a means to reduce the error rate in a double-exponential manner as the number of concatenation levels increase, assuming the physical error rate is below some threshold value, deemed the fault-tolerance threshold [6]. Subsequently, topological quantum codes were proposed, beginning with the surface code. There, logical information is stored in highly non-local degrees of freedom, while using local stabilizer checks, thus providing the ability to increase the protection of the code by increasing the size of the physical lattice encoding the information. One of the primary advantages of schemes such as the surface code is its high threshold value in comparison to concatenated code schemes. For depolarizing noise on each gate and memory location, the threshold value of the surface code is on the order of $10^{-2}$ [7] in comparison to $10^{-3}$ [8, 9] for most concatenated schemes. Moreover, stabilizer syndrome checks are simpler as the weight of the checks remains fixed as the distance increases, unlike in the case of concatenated architectures.

The goal of this work is to estimate the overhead with a recently proposed scheme for fault-tolerant computation using concatenated codes that allows for the implementation of a universal set of gates without the need for magic state distillation [10, 11]. Magic state distillation forecasts to be a challenge for logical computation on 2D topological codes, such as the surface codes. As such, many recent developments in the area of quantum error correction have focused on circumventing no go theorems regarding the implementation of universal quantum logic using transversal gates (the simplest form of fault-tolerant operation) [12–14]. The 105-qubit scheme circumvents these no go theorems by concatenating complementary sets of transversal gates and lead to a fault-tolerant threshold of $1.28 \times 10^{-3}$ for depolarizing noise. While this threshold rate compares favourably with other concatenated methods, it is still an order of magnitude below that of the surface code, thus will require higher distance iterations to reach a given target error rate when compared to the surface code. However, the primary advantage of the universal scheme is to avoid magic state distillation and as such this potential reduction in complexity would allow for the concatenated model to have a reduced overall overhead. The main result of this work is to provide a lower bound on the number of qubits and gates that would be required for the universal concatenated scheme to reach particular target error rates, given a physical depolarizing error rate. In order to do so, the overhead in state preparation needed for Steane error correction [15] as well as the suppression of the logical error rate as a function of concatenation level are determined.

In Section II we review the concepts behind Steane error correction and outline important parameters for the counting of resources related to the logical ancilla state
II. FAULT-TOLERANT SCHEME FOR ERROR CORRECTION

In this section we describe the fault-tolerant error correction scheme used in the implementation of the universal concatenated quantum codes considered in this paper. We use Steane’s method for fault-tolerant error correction [8, 15] which applies to CSS codes with stabilizer generators given by tensor products of all X and I operators (X generators) or tensor products of all Z and I operators (Z generators).

In Steane’s method, the ancilla qubits used for syndrome extraction are encoded using the same CSS code that protects the data qubits. Since the stabilizer generators are separated into X and Z generators, we can measure them separately. To measure the Z generators, we prepare the ancilla in the encoded state \(|\uparrow\rangle = (|0\rangle + |1\rangle)/\sqrt{2}\) and apply transversal CNOT gates with the data block as control and the ancilla block as target. This will propagate each X error in the data block to the corresponding position in the ancilla block. The ancilla is encoded in the \(|\uparrow\rangle\) state since it is an eigenstate with eigenvalue one of the logical X Pauli operator and thus the CNOT gate has no effect on the encoded state of the ancilla. If no errors occur in either the preparation of \(|\uparrow\rangle\) or the encoded CNOT gate, then measuring the ancilla block in the Z-basis detects the X errors when considering the Z stabilizers of the code which are products of the individual measurements. Similarly, to detect and correct Z errors, we prepare an ancilla state in the \(|\overline{0}\rangle\) basis and apply transversal CNOT gates with the data block as target and ancilla block as control. The sequence ends by performing a measurement of the ancilla in the X-basis. In general, the circuits used for encoding the \(|\uparrow\rangle\) and \(|\overline{0}\rangle\) ancilla states are not fault-tolerant, i.e., if errors occur during the encoding step. If an error is detected (or the −1 eigenvalue of a logical Pauli operator is measured), the ancilla states are rejected and the error correction round starts over. The terms \(p_{i,j}^{(k)}\) and \(p_{j}^{(k)}\) (where \(i, j \in \{1, 2\}\)) correspond to the probabilities that no X (Z) errors are detected in the blocks \(|\overline{0}\rangle_i\text{ and } (|\overline{0}\rangle)\). Conditioned on acceptance of the previous blocks, \(p_{i,j}^{(k)}\) denote the probabilities that no Z (X) errors are detected in the last verifier blocks. These probabilities will be used in the depolarizing noise overhead calculations of the 49-qubit and 105-qubit codes.

![FIG. 1: Illustration of Steane’s error correction circuit.](image-url)
The leading and trailing error correction circuits are implemented using Steane’s method (see Section II) which is a protocol for a fault-tolerant error correction.

measurement in the $Z$ basis. Not only are $X$ errors detected but after a classical error correction step the eigenvalue of the encoded $Z$ operator is also found. If a non-trivial syndrome is measured or the eigenvalue of $Z$ is found to be $-1$, all ancilla blocks are rejected and the error correction protocol starts over. The latter will play an important role when considering the resource overhead of the concatenated 49-qubit and 105-qubit code.

III. MALIGNANT SET COUNTING OVERVIEW

Following [8], we define a $k$-Rec to be the encoded gate under consideration acting on codewords followed by a trailing error correction (TEC) circuit at the $k$-th level of concatenation. A $k$-exRec (where exRec stands for extended rectangle) corresponds to a leading error correction (LEC) circuit followed by a $k$-Rec (see Figure 2).

Consider an arbitrary 1-exRec. A location can either be a 0-preparation, 0-measurement or a gate (note that the identity gate corresponds to a resting qubit). We say that a set of $m$ locations is benign if the 1-Rec contained in the 1-exRec is correct for arbitrary faults occurring at those locations. Otherwise, the set of locations are defined to be malignant. A 1-Rec is correct if it takes any input with no more than one error per block to an output with no more than one error per block.

The idea behind malignant set counting is that for a fixed number of locations in the circuit, we would like to count all such sets of locations which are malignant. A 1-exRec will be bad if it contains faults at a malignant set of locations, otherwise we will define it to be good. We can generalize the definition of goodness and badness to a $k$-exRec (for $k > 1$) by defining the $k$-exRec to be bad if it contains independent bad $(k-1)$-exRecs at a malignant set of locations, otherwise we define it to be good.

In this section, we will consider an independent stochastic noise model where fault locations are independently and identically distributed. The operations at the chosen locations are arbitrary trace-preserving completely positive maps. The latter noise model is also known as adversarial noise. Since an arbitrary fault can be expanded in terms of Pauli operators, a set of locations will be benign if the 1-Rec contained in the 1-exRec is correct for all Pauli faults at those locations. Further, if all of the locations are in the LEC of the 1-exRec, then by definition the 1-Rec contained in the 1-exRec is correct. Therefore, when counting malignant sets of locations, we will exclude the cases where all faults are in a LEC. As in Ref. [8], for a fixed set of locations, we insert all combinations of $X$ and $Z$ Pauli faults at those locations and propagate them through the 1-exRec. If for all combinations of Pauli faults the 1-Rec contained in the 1-exRec is correct, then the set of locations are benign.

Consider a 1-exRec with a total of $L$ locations and suppose that the EC’s have $L_{EC}$ locations. For a circuit simulating a $t$-qubit gate, we define

$$L_{n,t} \equiv \binom{L}{n} - t \binom{L_{EC}}{n},$$

where $n$ is the number of faulty locations in the circuit. Similarly, define

$$A_{n,t} \equiv \frac{n_{mal}}{N},$$

where $n_{mal} = n_{mal}/N$ is the fraction of malignant locations containing $n$ faults. Due to limits in computation time, when calculating the noise threshold for a particular gate, we count malignant sets of locations for up to $m$ faults and assume that all $m+1$ sets of locations are malignant. For large enough $m$, the error in the truncation can be made very small [8]. Defining the probability of a $k$-exRec to be bad by $\varepsilon^{(k)}$, we can use the statistical independence of bad $(k-1)$-exRecs to calculate an upper bound on $\varepsilon^{(k)}$

$$\varepsilon^{(k)} \leq A_{2,t}(\varepsilon^{(k-1)})^2 + A_{3,t}(\varepsilon^{(k-1)})^3 + \ldots + A_{m-1,t}(\varepsilon^{(k-1)})^{m-1} + L_{m,t}(\varepsilon^{(k-1)})^m.$$
well as single qubit measurements in the X and Z eigen-basis. The level-0 locations used in our circuits are given in Table I. Since the wait time of a qubit during the application of a gate could differ from the wait time of a measurement, we can associate different failure rates for the two rest cycles. Generalizing Eq. 3 to include different failure rates for distinct level-0 locations, we obtain

\[
e^{(k)} \leq \sum_{j \leq i = 1}^{l_{max}} \alpha_{ij,t}(k-1) \varepsilon_{j}^{(k-1)} + \sum_{l \leq j \leq 1}^{l_{max}} \alpha_{ij,t}(k-1) \varepsilon_{j}^{(k-1)}(k-1) + \cdots + L_{m,t} \varepsilon_{j}^{(k-1)}m,
\]

(4)

where \(l_{max}\) depends on the types of locations in a particular exRec. Hence, exRec’s excluding \(H\) or \(T\) gates would have \(l_{max} = 7\). The indices correspond to a particular level-0 location and are summed over all location types of the exRec under consideration. \(\alpha_{ij,t}^{(2)}\) corresponds to the number of malignant pairs of types \(i\) and \(j\) for a gadget acting on \(t\) qubits. For example, following the indexing from Table I, 5 is the label used for a measurement in the \(X\) basis and 7 is the label for a CNOT gate. Therefore, \(\alpha_{ij,t}^{(2)}\) corresponds to the number of malignant pairs in the exRec where one location is a measurement in the \(X\) basis and the other location is a CNOT gate. Generalizing to larger sets of locations, \(\alpha_{i_{1},i_{2},\ldots,i_{n}}^{(n)}\) is the number of malignant sets of \(n\) locations of type \(i_{1},i_{2},\ldots,i_{n}\). For the remainder of this section, we will assume that all location types have the same failure probability, so that the upper bound in Eq. 3 will be used for the threshold calculation (in this case \(A_{2},t\) is simply the sum of all the elements of the \(\alpha_{ij,t}^{(2)}\) matrix, and \(\varepsilon_{j}^{(k-1)} = \varepsilon_{j}^{(k-1)} = \varepsilon^{(k-1)}\)). From Eq. 3, it follows that

\[
e^{(k)} \leq A_{t}^{(k)} \varepsilon^{(k-1)}^{2},
\]

(5)

where the threshold estimate is

\[
e \leq e_{0} = (A_{1}^{(k)})^{-1}.
\]

(6)

\(A_{t}^{(k)}\) can be calculated from the polynomial equation

\[
(A_{t}^{(k)})^{m} - A_{2,t}(A_{t}^{(k)})^{m-1} - A_{3,t}(A_{t}^{(k)})^{m-2} - \cdots - A_{m-1,t}A_{t}^{(k)} = 0.
\]

(7)

Since the coefficients \(A_{j,t}\) are strictly increasing for increasing \(j\), there will only be one positive solution to Eq. 7.

Recall that the ancilla blocks used to extract the error syndrome need to successfully pass a verification test, otherwise the ancilla’s are rejected and the computation is started over. Instead, the failure probability for a \(k\)-exRec should be upper bounded conditioned on the acceptance of all ancilla blocks. In calculating the coefficients \(A_{j,t}\), we count sets of locations such that faults at those locations cause the \(k\)-exRec to fail but lead to acceptance of all ancilla blocks. Hence, the upper bound in Eq. 3 actually corresponds to the joint probability of acceptance of all ancilla blocks and failure of the \(k\)-exRec. We can use Bayes’ rule to obtained the conditional probability of failure given the acceptance of all ancillas. Using the notation from Ref. [8], we need to calculate the probability \(P(\overline{0},\text{accept})\) and \(P(\overline{1},\text{accept})\) that a level-\(k\) encoded \(\overline{0}\) or \(\overline{1}\) passes the verification test. Distinction between \(P(\overline{0},\text{accept})\) and \(P(\overline{1},\text{accept})\) is important since the encoding circuits for \(\overline{0}\) and \(\overline{1}\) are not symmetric for the 15-qubit Reed-Muller code (since the code is not self-dual) and so they will contain a different number of locations. We define

\[
P_{\text{min},\text{accept}} = \min\{P(\overline{0},\text{accept}), P(\overline{1},\text{accept})\},
\]

(8)

which corresponds to the smallest of the two acceptance probabilities. In Steane’s error correction protocol, there will always be an equal number of encoded \(\overline{0}\) and \(\overline{1}\) circuits in the \(k\)-EC’s. We define \(n_{\text{anc}}\) to be the number of encoded \(\overline{0}\) or \(\overline{1}\) circuits in the \(k\)-exRec under consideration. Using Bayes’ rule, the probability of failure \(e^{(k)}\) for the \(k\)-exRec, conditioned on acceptance of all ancillas can be upper bounded as

\[
e^{(k)} \leq (P_{\text{min},\text{accept}})^{n_{\text{anc}}} e_{\text{joint}},
\]

(9)

where \(e_{\text{joint}}\) is upper bounded by Eq. 3. Let \(C_{\overline{0}}\) and \(C_{\overline{1}}\) correspond to the number of locations in the encoding and verification circuits of \(\overline{0}\) and \(\overline{1}\) in the EC’s. For an ancilla to be rejected, the previous encoding and verification circuits must contain at least one bad \((k-1)\)-exRec. Therefore, a lower bound on \(P_{\text{min},\text{accept}}\) is given by

\[
P_{\text{min},\text{accept}} \geq 1 - C_{\text{max}} e^{(k-1)},
\]

(10)

where \(C_{\text{max}} = \max\{C_{\overline{0}}, C_{\overline{1}}\}\). Using 8–10 and assuming that \(e^{(k-1)} < (A_{t}^{(k)})^{-1}\), we have that

\[
e^{(k)} \leq \left(1 - \frac{C_{\text{max}}}{A_{t}^{(k)}}\right)^{-n_{\text{anc}}} A_{t}^{(k)} e^{(k-1)}^{2}.
\]

(11)

Defining

\[
A_{t}^{(\text{min})} \equiv \left(1 - \frac{C_{\text{max}}}{A_{t}^{(k)}}\right)^{-n_{\text{anc}}} A_{t}^{(k)},
\]

(12)
the threshold estimate is then given by

\[
\varepsilon_0 = (A^t_k)^{-1}. \tag{13}
\]

IV. THE 105 AND 49-QUBIT CODES

In this section we will review the construction of the 105-qubit concatenated code that allows for universal fault-tolerant logical gate implementations, as well as a simplification of this construction to 49 qubits [16]. We shall then discuss the different ways we implement decoding and error correction given syndrome measurements, while concluding with how ancilla logical state preparation is performed with a reduced number of time steps.

A. Fault-tolerant universal concatenated quantum codes

The idea behind the 105-qubit construction is to use two different error correcting codes, with different sets of transversal gates (logical gates that can be implemented by applying individual gates to each qubit composing the code), in concatenation in order to implement a universal set of fault-tolerant logical operations [10]. In the construction of the 105-qubit code, the outer code is designated as the 7-qubit code and therefore contains transversal Clifford operations. The inner code, that is each qubit composing the 7-qubit code, is the 15-qubit code which has transversal logical $CNOT$ and $T$ gates, where $T = |0\rangle\langle 0| + e^{i\pi/4} |1\rangle\langle 1|$ and completes the universal gate set when combined with Clifford operations. Since the $T$ gate cannot be implemented transversally for the 7-qubit code, any logical construction will necessarily have to couple different qubits in the code (in the case of the 105-qubit code, coupling qubits corresponds to coupling blocks of qubits composing the code). There exists a construction for the implementation of the $T$ gate using a sequence of $CNOT$ gates and a $T$ gate on the qubits of the 7-qubit code. However, since each of these operations are implemented at the logical level from the perspective of the 15-qubit code, they will all be transversal with respect to this code. Therefore, any single qubit fault occurring during the action of this sequence of gates may lead to a propagation of the faults, but in a controlled manner to only a single location to each of the codeblocks. Therefore, the logical gate remains fault-tolerant as any single fault remains correctable.

The logical Hadamard is implemented by applying the logical Hadamard gate on each of the seven encoded codeblocks (as the Hadamard is transversal for the 7-qubit code). However, the Hadamard is not transversal on each 15-qubit codeblock, and as such a single error may spread to form a logical fault on an individual codeblock. However, since only one codeblock is corrupted, overall the error will remain correctable as such an error can be detected and corrected by the 7-qubit outer code syndrome. This complication will have important consequences for decoding, as explained further in this section.

An important observation is that the re-encoding of the qubits into 15-qubit codeblocks is only important to protecting the blocks that have active gates in the implementation of the logical $T$ gate. Therefore, it is only necessary to encode three codeblocks, corresponding to the set of blocks that would correspond to a logical Pauli operator for the 7-qubit code [16]. Therefore, the total qubit count can be reduced to be $3 \times 15 + 4 = 49$ qubits. While the overall distance of the code will be lower, reduced from a distance 9 to distance 5 code, the logical operations will still be able to correct for arbitrary single qubit faults, just as in the case of the 105-qubit code. This idea can be generalized further to any construction using as an outer code a 2D color code, which has transversal Clifford operations, and choosing a set of qubits that contain both logical Pauli operators and re-encoding these qubits in a 3D color code containing the transversal $CNOT$ and $T$ gates required from the original scheme [17].

B. Decoding the 105-qubit code

As we have previously explained, the 105-qubit code is a distance 9 quantum error correcting code whose distance is sacrificed for the implementation of the non globally-transversal $H$ and $T$ gates. However, the decoder should be designed such that for the $CNOT$ gate any weight-4 error can be corrected. The 105-qubit syndromes consist of the 15-qubit syndromes on each of the codeblocks composing the outer code, as well as the 6 syndromes of the 7-qubit code which correspond to syndromes across 4 blocks (since all stabilizers of the 7-qubit code have weight 4).

The most basic decoding scheme is the greedy decoder, where each of the 15-qubit codeblocks are corrected according to their measured syndromes individually, and then the outer 7-qubit code is corrected independently according to the remaining syndrome \(^1\). The greedy decoder will fail to correct for all weight-4 errors. For example, consider the case when two 15-qubit codeblocks each have a weight-2 $Z$ error. Each codeblock is then corrected with the identified single qubit recovery (since the 15-qubit code is weight-3 for $Z$ errors, it can only correct weight-1 errors) resulting in a logical error on the two 15-qubit codeblocks. Then, the outer 7-qubit stabilizers will identify this weight-2 logical fault with a logical fault on a third codeblock, and thus “correct” by implementing a logical $Z$ on the third codeblock. As such, the

\(^1\) Note that in Steane error correction, all of the syndromes are measured in parallel for both the 15-qubit codeblocks as well as the outer 7-qubit code. Any correction made at the 15-qubit level that would modify the 7-qubit syndrome results can be accounted for in software.
final state will undergo a global logical $Z$ error from the composition of the 3 logical codeblock errors. Therefore, having corrected each of the 15-qubit codeblocks, when correcting the 7-qubit codeblocks, information from the 15-qubit syndromes will have to be used in the correction of the outer 7-qubit code.

The decoding of the 105-qubit code will be implemented using the following steps:

1. Correct all of the 15-qubit codeblocks individually, and store which codeblocks underwent any correction.
2. Update the 7-qubit syndromes according to the corrections from the 15-qubit codeblocks.
3. If the 7-qubit syndrome is trivial (no syndrome identifies an error) then correction complete.
4. If the 7-qubit syndrome identifies an error on a codeblock that did not undergo a 15-qubit correction, and in addition there is a set of complementary blocks $^2$ that were corrected at the 15-qubit level, then perform further logical operations on the complementary blocks, then correction complete.
5. Otherwise, correct the identified codeblock by applying a logical correction.

We illustrate the advantage of this decoding scheme by highlighting the example that the greedy decoder failed to correct. Consider weight-2 $Z$ errors on the first and second codeblocks. Each of these codeblocks are corrected by applying a weight-1 correction, resulting in logical $Z$ errors on each of the codeblocks. Then, the 7-qubit syndrome would identify an error on codeblock 3, since the syndrome associated with $Z_1Z_2$ is equivalent to $Z_3$. Since the 15-qubit decoder did not make any corrections on codeblock 3, yet did make corrections to codeblocks 1 and 2, which are complementary to codeblock 3, logical $Z$ corrections are applied to each of these codeblocks, therefore correcting all of the errors. One can verify that all weight-4 errors will be corrected by this scheme, thus achieving the promised distance of the code. Of course, for the implementation of the logical $H$ and $T$ gates, not all weight-4 errors will be corrected as the gates are not globally transversal. However, all weight-1 errors will still be corrected by our decoding scheme.

C. Decoding the 49-qubit code

The 49-qubit code sacrifices the full distance of the 105-qubit code by only encoding three codeblocks, therefore reducing the overall distance to be 5. As such, any decoder will be able to correct at most any weight-2 error. The correction scheme implemented for the logical $CNOT$ and $T$ gate is as follows (as described below, a different decoder is used for the logical $H$):

1. Correct the three 15-qubit codeblocks, tracking which blocks contained errors.
2. Update the 7-qubit syndromes according to the corrections from the 15-qubit codeblocks.
3. If the 7-qubit syndrome identifies a 15-qubit codeblock that had a trivial syndrome in Step 1, then perform a weight-2 correction on the complementary single qubit blocks, then correction complete.
4. For any other non-trivial 7-qubit syndrome, correct according to the identified single or 15-qubit codeblock.

For the transversal $CNOT$, any weight-1 error will either be corrected originally by the 15-qubit syndrome measurement if it occurs on a codeblock or at the 7-qubit level if it occurs on an unencoded block. If two errors occur there are four possibilities. If both errors occur on the same encoded codeblock, then they are corrected at the 15-qubit level, potentially resulting in a logical fault on that codeblock. However, such a logical fault will be detected by the 7-qubit stabilizers and as such will be corrected according to the protocol. If the two errors occur on different encoded 15-qubit codeblocks, they will each be correctly identified at the 15-qubit level. If one error occurs on a codeblock and one on an unencoded block, then the error on the 15-qubit codeblock will be corrected, and only the error on the single qubit will remain when measuring the 7-qubit syndromes and will be correctly identified. The tricky case comes when errors occur on two different unencoded blocks, therefore not identified by the 15-qubit stabilizers. In this case, the codeblock that is complementary to these two errors will be identified when the 7-qubit stabilizers are measured, however since no error had been identified on that codeblock at the 15-qubit level, the error complementary to that block is corrected on single qubits resulting in an “error” that will be logically equivalent to the identity.

In the case of the $T$ gate, some of the distance is sacrificed in order to implement the non-transversal $T$ gate, yet remains fault-tolerant and can correct for any weight-1 error. Any weight-1 error is corrected equivalently to a Greedy decoder in the case of our protocol, and as such is corrected in the same manner as in the case of $CNOT$. The only difference in potential realizations is that a weight-1 error on any of the codeblocks may spread to a weight-1 error on the other codeblocks, yet they all remain correctable through the protocol by correcting at the 15-qubit level.

Unlike in the case of the 105-qubit code, it is important to point out that a different decoder must be used in the case of the logical Hadamard. Due to the non fault-tolerant construction of the logical Hadamard on a given

---

$^2$ The complementary blocks are the two blocks that along with the identified block would form a logical error for the 7-qubit code. For example, $Z_1Z_2Z_3$ forms a logical error for the 7-qubit code, therefore complementary blocks for block 1 are (2,3), yet also blocks (4,5) and (6,7), since they are logically equivalent.
15-qubit codeblock, a single fault on such a codeblock could result in a logical fault on that codeblock at the conclusion of the gate. Such an error would be corrected in the wrong fashion according to the above decoder, as it would go unrecognized by the 15-qubit stabilizers and would falsely identify a correction on the complementary single qubits. As such, the solution is to correct in a greedy fashion at the 7-qubit level, thus resulting in a correction on the single 15-qubit block. Therefore, the resulting protocol would be to correct in a greedy manner at the 15-qubit level and the 7-qubit level, and weight-2 errors on two single-qubit codeblocks would therefore result in a logical fault unlike in the case of the transversal CNOT. As mentioned previously, not being able to correct all weight-2 errors is due to sacrificing the full code distance for the implementation of the logical $H$.

### D. Ancilla preparation and exRec circuits for the 105-qubit and 49-qubit code

In order to describe the ancilla states $|0\rangle$ and $|\mp\rangle$ for the 105 and 49-qubit codes, we first obtain their encoding circuits for the 7 and 15-qubit codes. Note that the 15-qubit Reed-Muller code is not self-dual which means that the encoding circuit $|\mp\rangle_{15}$ cannot simply be obtained by reversing the direction of each CNOT gate and swapping the physical $|0\rangle$ and $|+\rangle$ states in $|0\rangle_{15}$. As will be shown, this asymmetry will result in a larger number of physical locations in $|\mp\rangle_{15}$ compared to $|0\rangle_{15}$.

For CSS stabilizer codes, the encoded states can be obtained by solving a partial Latin rectangle using the codes stabilizer generators [18]. However, as was shown in Ref. [9], the circuits obtained from the previous method can be further optimized by considering overlaps in the codes stabilizer generators. As an example, consider the encoding circuit for the state $|0\rangle_7$ of the 7-qubit code. The circuit in Fig. 3(a) (obtained via Steane’s Latin rectangle method) contains nine CNOT gates. The logical state $|0\rangle_7$ can also be obtained by the circuit in Fig. 3(b) which uses eight CNOT gates, one fewer than the previous circuit. To see this, recall that the $X$ stabilizer generators for the 7-qubit code take the form $g_1 = IIIXXXX$, $g_2 = IXXIXXX$ and $g_3 = XIXIXIX$. In Fig. 3(a), it can be seen that qubit seven is the target of qubits two and four and the corresponding stabilizer generators $g_1$ and $g_2$ overlap on qubits six and seven. Consequently, it is possible to replace the two CNOT’s with control qubits two and four and target qubit seven with a CNOT having control on the sixth qubit and target on the seventh qubit. We use the circuit in Fig. 3(b) as the outer level of the state $|0\rangle_{105}$. Since the 7-qubit code is self-dual, the $|\mp\rangle_7$ can be obtained by reversing the direction of each CNOT gate and swapping the physical $|0\rangle$ and $|+\rangle$ states.

For the 49-qubit code, we use the circuits in Fig. 4 at the outer level in order to minimize the total number of wait times experienced by each qubit. To see this, recall that the logical $Z$ operator for the 15-qubit code can take the form $Z = Z_1Z_2Z_3$ (a weight 3 operator, since the first three bits of the codewords have even parity). Consequently, it is possible to apply a logical CNOT gate with the control lying on the first three qubits of a 15-qubit codeblock and target on a single-qubit codeblock. This would require the use of only three physical CNOT gates, one per time step. In this way, a $Z$ error on a single-qubit block would propagate to a logical $Z$ error on a 15-qubit codeblock. Since CNOT gates between 15-qubit codeblocks can be implemented transversally, a closer look at Fig. 4 shows that the first two CNOT gates can be implemented in three time steps and all other CNOT gates can be implemented in a single time step. Hence the overall number of time steps required to implement the CNOT gates in Fig. 4 is $3 + 1 + 1 = 5$. If instead we were to use the circuits in Fig. 3, there are three CNOT gates coupling 15-qubit codeblocks to single qubit blocks. Two of these CNOT gates have their control on a 15-qubit codeblock encoding a $|\mp\rangle$ state and so would be implemented in three time steps each. The
These circuits have a lower space-time overhead compared to the one used in Fig. 3(b). For example, the CNOT with block 1 as control and block 3 as target in (a) and (b) can be implemented in three time steps instead of seven since \( Z = Z_1 Z_2 Z_3 \) is a logical \( Z \) operator for the 15-qubit Reed-Muller code.

Furthermore, any CNOT gate between blocks 1, 6 and 7 can be implemented transversally. A CNOT gate coupling the \(+\) state to the 15-qubit \( |0\rangle \) codeblock would require 7 time steps, since the minimum weight logical \( X \) for the 15-qubit code is 7. The total number of time steps would have been \( 7 + 3 + 3 = 13 \) instead of 5.

Next we apply the stabilizer overlap method to obtain optimized encoded \( |0\rangle_{15} \) and \( |+\rangle_{15} \) states of the 15-qubit Reed-Muller code. Fig. 5(a) and Fig. 5(b) give the encoding circuits for the \( |0\rangle_{15} \) and \( |+\rangle_{15} \) states using Steane’s Latin rectangle method. There are 28 CNOT gates in \( |0\rangle_{15} \) and 32 CNOT gates in \( |+\rangle_{15} \). For the circuit \( |0\rangle_{15} \), the stabilizer overlap method removes six CNOT gates. For the \( |+\rangle_{15} \), instead of using the stabilizer overlap method to reduce the number of CNOT gates, we consider all pairs of CNOT gates that have the same target qubit. For a given pair, we replace the pair by a single CNOT with a different control qubit. We test all 13 different possible controls until we obtain a correct \( |+\rangle_{15} \) state. Applying the latter technique, we were able to remove seven CNOT gates from the \( |+\rangle_{15} \) state (see Figs. 6(a) and 6(b)).

Concatenating the optimized circuits of Fig. 3 with the circuits of Fig. 6 and taking the 7-qubit code as the outer code, we obtain the \( |0\rangle_{105} \) and \( |+\rangle_{105} \) ancilla states. The 49-qubit code ancilla states are obtained by concatenating the states of Fig. 4 with those of Fig. 6 (see Table II for an enumeration of all the types of locations in 49 and 105-qubit ancilla states). A 1-EC contains four \( |0\rangle \) states, four \( |+\rangle \) states, eight encoded CNOT gates, two storage locations and four encoded \( Z \)-measurement and \( X \)-measurement locations (see Fig. 1). Using the values of Table II, the total number of locations in a 1-EC circuit of the 105-qubit code is thus 7110.

Threshold estimates for the 7-qubit CSS code and the 23-qubit Golay code were calculated in Ref. [8, 9]. The threshold calculation was limited by the exRec with the largest number of locations which, in both cases, corresponded to the CNOT-exRec. Given the non-transversal nature of the \( T \)-gate in the 7-qubit code and the Hadamard gate in the 15-qubit code, the threshold
FIG. 6: Optimized encoding circuits for the 15-qubit Reed-Muller code using the stabilizer overlap method for $|0\rangle_{15}$ and a computer search algorithm for $|+\rangle_{15}$. $|0\rangle_{15}$ contains 22 CNOT gates and 27 resting qubit locations. $|+\rangle_{15}$ contains 25 CNOT gates and 31 resting qubit locations.

FIG. 7: Logical Hadamard $H$ circuit for $[[15,1,3]]$ Reed-Muller code. Logical $H$ for the 105-qubit code is implemented fault-tolerantly by applying each non-fault-tolerant logical $H$ gates in parallel. Note that there are a total of nine time steps. Consequently, the 49-qubit code Hadamard circuit will contain a single physical Hadamard and 8 resting qubit locations on blocks 2-5.

calculation will no longer be limited by the exRec with the largest number of locations. As will be shown below, the threshold calculation is instead limited by the Hadamard-exRec. It is thus worthwhile to analyze the Hadamard circuit in more detail.

The encoding circuit for the Hadamard gate must map the stabilizer generators of the 15-qubit code to an element of the stabilizer group. Furthermore, since $HXH = Z$ and $HZH = X$, the logical $X$ operator of the 15-qubit code must be mapped to the logical $Z$ operator and vice-versa. The circuit in Figure 7 satisfies these properties. To derive such a circuit, we wrote a program in Matlab which inserted CNOT gates at random locations within the 15 physical qubits and propagated all stabilizer generators and logical operators through the circuit. If all operators transformed appropriately as described above, then the locations of the CNOT gates were recorded. The best circuit that we found contains a total of 14 CNOT gates, one physical Hadamard gate and has a depth of 9 time steps (see Table III for a complete enumeration of the locations in the Hadamard circuit for the 15-qubit, 49-qubit and 105-qubit code). However, it is still an open question whether a circuit using fewer CNOT gates with a smaller depth can be found.

One important aspect of the circuit in Fig. 7 is that input $Z$ errors from the LEC are much more likely to lead to a logical error at the output of the Hadamard circuit than input $X$ errors. To illustrate this, we consider two different cases. In the first case, we insert a single $X$ error at the input of the Hadamard circuit and propagate the error throughout the circuit to determine if a logical error occurred at the output of the circuit, which occurs in the case of qubits 4, 8 and 12. In the second case, we perform the exact same operations but with a single $Z$ error.
| Number of CNOT’s | $H_{15}$ | $H_{49}$ | $H_{105}$ |
|-----------------|---------|---------|---------|
| Number of Resting qubits | 14 | 42 | 98 |
| Number of Physical $H$ states | 1 | 7 | 7 |
| Total | 121 | 399 | 847 |

TABLE III: Enumeration of all the types of locations for the 15-qubit, 49-qubit and 105-qubit Hadamard gate. Restricting to the 105-qubit code, since a 1-EC circuit contains 7110 locations, the total number of locations in the Hadamard-exRec is 15067. These quantities will be relevant in the adversarial noise threshold calculation of the Hadamard-exRec as well as in the resource overhead calculation.

FIG. 8: Logical $T$ circuit for [[7, 1, 3]] CSS code. All stabilizer generators map to elements in the stabilizer group. Furthermore, $X \rightarrow X(I + Z)$ and $Z \rightarrow Z$ as required.

In this case, qubits 1, 4, 5, 8, 9, 12 and 13 produce different logical faults at the output of the Hadamard circuit compared to when input $X$ errors were considered. It should then be expected that $Z$ errors will play an important role when calculating the noise threshold for the concatenated scheme.

For the 105-qubit code, logical Hadamard is obtained by applying the non-transversal logical Hadamard of Fig 7 to each of the encoded 15-qubit codeblocks. For the 49-qubit code, the logical Hadamard gate is obtained by applying the non-transversal logical Hadamard of Fig 7 to blocks 1, 6 and 7. Blocks 2-5 will consist of a physical Hadamard gate along with 8 resting qubit locations (since there are a total of 9 time-steps in Fig 7).

As was explained above, a single error can lead to a logical fault when propagating through the 15-qubit Hadamard circuit. However, since the Hadamard circuit is transversal for the 7-qubit CSS code, a weight one error will lead to at most a single fault on one of the codeblocks which will be corrected by the outer code. Thus the Hadamard circuit for the 49 and 105-qubit code is fault-tolerant.

Next we analyze the construction of the $T$-gate circuit for the 7-qubit code following the ideas of Ref. [10, 11]. Similarly to the Hadamard circuit, to obtain a circuit that correctly encodes the $T$-gate for the 7-qubit code, all stabilizer generators must be mapped to elements of the stabilizer group. Furthermore, since $T X T^\dagger = \frac{1}{\sqrt{2}} X + Y$ and $T Z T^\dagger = Z$, we require that the logical operators for the 7-qubit code transforms as $X \rightarrow X(I + Z)$ and $Z \rightarrow Z$. The circuit in Fig. 8 satisfies the above properties. It should be noted that regardless of the error model studied, we treated the transformation of $X$ errors in our simulations by taking an adversarial approach to their transformation to $X$ or $Y$ errors, depending on the other errors in the circuit. That is, we considered both the case when the $X$ error transformed to either $X$ or $Y$, and treated the worst case logical error outcome based on the choice of error. This sufficed to prove a lower bound on the appropriate threshold.

Logical $T$ for the 49 and 105-qubit code is constructed from the circuit in Fig. 8 with each codeblock encoded using the 15-qubit Reed-Muller code (for the 49-qubit code, blocks 2-5 only contain one qubit). The construction is not fault-tolerant on the outer code since errors can spread between codeblocks. However, since the underlying logical gates are transversal on the 15-qubit codeblocks, a single error would propagate to at most a single error on each codeblock which would be corrected by the inner code.

E. Adversarial noise threshold results for the 105-qubit code

We present the results of the threshold analysis for the adversarial noise model described in section III. We will begin by computing the noise threshold for the Hadamard-exRec which turns out to be the circuit that provides a lower bound for the threshold of the 105-qubit code. From the analysis leading to Eq. 3, we first need to compute the coefficients $A_{j,t}$ where $j \in \{2, 3, \ldots, m\}$ for some cutoff value $m$. Given the large number of locations in the exRec’s for the 105-qubit code, it is computationally impractical to count all malignant sets of locations. In order to overcome this difficulty, we use a Monte Carlo method following the ideas of Ref. [19]. Suppose we want to count all malignant locations for a set of $j$ faults, i.e. the coefficient $A_{j,t}$. Instead of looking at every combination of $j$ locations within the full exRec, we can uniformly sample the set of all fault paths for a fixed set of $j$ faults. We thus obtain an estimate of the fraction $\hat{f}_{j,t}$ of malignant faults for $j$ sets of locations which in turn provides an estimate of the exact coefficient $A_{j,t}$. The standard error is obtained from the relation

$$\sigma_{j,t} = \sqrt{\hat{f}_{j,t}(1 - \hat{f}_{j,t})/N},$$ (14)
TABLE IV: Value of Hadamard-exRec $A_{j,1}$ coefficients obtained from a Monte Carlo simulation which are used in the threshold calculation for adversarial noise.

| $A_{j,1}$ coefficients for Hadamard 1-exRec | Monte Carlo estimate |
|---------------------------------------------|-----------------------|
| $A_{2,1}$                                  | $(6.80 \pm 0.24) \times 10^4$ |
| $A_{3,1}$                                  | $(1.73 \pm 0.03) \times 10^9$ |
| $A_{4,1}$                                  | $(1.30 \pm 0.05) \times 10^{13}$ |
| $A_{5,1}$                                  | $(6.44 \pm 0.20) \times 10^{76}$ |
| $A_{6,1}$                                  | $(2.42 \pm 0.06) \times 10^{20}$ |
| $L_{7,1}$                                  | $3.47 \times 10^{25}$ |

The goal of designing fault-tolerant architectures is to allow arbitrary long computations to be performed on large-scale quantum computers where the probability of failure can be made as small as desired. For concatenated coding schemes, fault-tolerant architectures give rise to asymptotic thresholds which corresponds to the error rate $p_{th}$ such that for physical error rates $p < p_{th}$, the logical error rate can be made as small as desired for sufficiently large number of concatenation levels. Furthermore, a quantum circuit containing $A$ gates can be simulated with probability of error at most $\epsilon$ with a space/time overhead which scales as $O(poly(log A/\epsilon)A)$ [20].

In this study, fault-tolerant syndrome measurement and error correction is implemented using Steane’s method (see Section II for a detailed description of the method) in order to take advantage of the CSS structure of the codes. Error correction steps are interleaved between the implementation of each fault-tolerant gate. Following Ref. [8], at the first level of concatenation, each logical gate is represented by a 1-exRec as illustrated in Fig 2. Hence, the components in a level-1 logical gate will consist of state preparation and measurement, physical gates and memory locations. We use a recursive simulation for higher levels of concatenation where a fault-tolerant gate at level $k$ is constructed by replacing each level-0 location in the level-$(k-1)$ logical gate by the corresponding level-1 rectangle.

The noise threshold for the 49 and 105-qubit code is computed by considering a depolarizing noise model for each physical (level-0) location. The depolarizing channel for a single qubit is defined by the quantum operation

$$\varepsilon(\rho) = (1 - \frac{3p}{4})\rho + \frac{p}{4}(X\rho X + Y\rho Y + Z\rho Z),$$

where $p$ will be referred to as the physical error rate. Using the same parameters as Ref. 18, the depolarizing channel is generalized to all forms of quantum operations as follows:

1. A noisy CNOT gate is modelled as applying a CNOT gate followed by, with probability $\frac{15p}{16}$, a

\[\varepsilon(\rho) = (1 - \frac{3p}{4})\rho + \frac{p}{4}(X\rho X + Y\rho Y + Z\rho Z),\]

\[\varepsilon(\rho) = (1 - \frac{3p}{4})\rho + \frac{p}{4}(X\rho X + Y\rho Y + Z\rho Z),\]

\[\varepsilon(\rho) = (1 - \frac{3p}{4})\rho + \frac{p}{4}(X\rho X + Y\rho Y + Z\rho Z),\]

\[\varepsilon(\rho) = (1 - \frac{3p}{4})\rho + \frac{p}{4}(X\rho X + Y\rho Y + Z\rho Z),\]
two-qubit Pauli error drawn uniformly and independently from \{I, X, Y, Z\}^\otimes 2 \setminus \{I \otimes I\}.

2. A noisy preparation of the \ket{0} state is modelled as the ideal preparation of the \ket{0} state with probability \(1 - \frac{3}{4}p\) and \ket{1} = X\ket{0} with probability \(\frac{3}{4}p\) (we use \(\frac{3}{4}p\) instead of \(\frac{1}{2}p\) since \(Y\) errors have the same effect as \(X\) errors). Similarly, the noisy preparation of the \ket{+} state is modelled as the ideal preparation of the \ket{+} state with probability \(1 - \frac{3}{4}p\) and \ket{−} = Z\ket{+} with probability \(\frac{3}{4}p\).

3. A noisy measurement in the \(Z\)-basis is modelled by applying a Pauli \(X\) error with probability \(\frac{3}{4}p\) followed by an ideal measurement in the \(Z\)-basis. Similarly, a noisy measurement in the \(X\)-basis is modelled by applying a Pauli \(Z\) error with probability \(\frac{3}{4}p\) followed by an ideal measurement in the \(X\)-basis.

4. A single-qubit gate error or storage error is modelled by applying the ideal gate (identity gate for a resting qubit) with probability \(1 - 3p/4\). With probability \(3p/4\), the ideal gate is implemented followed by a Pauli error chosen uniformly from the set \{\(X, Y, Z\)\}.

To determine the probability of having a logical fault at the output of an \(\text{exRec}\), we first define the notion of a malignant error event. Let \(\ket{\psi_1}\) be a single or two-qubit logical state obtained by applying ideal decoders immediately after the LEC circuit and \(\ket{\psi_2}\) the logical state obtained by applying ideal decoders immediately after the TEC. The event \(\text{mal}_E\) is defined as \(\ket{\psi_2} = EU\ket{\psi_1}\) where \(E\) is a single or two-qubit error and \(U\) is the desired gate. We now describe our simulation protocol to obtain estimates of the event \(\text{mal}_E\) for various logical gates.

1. Given a \(\text{exRec}\) encoding a particular gate, we fix a particular value of \(p\) and \(N\), where \(N\) corresponds to the total number of iterations that the depolarizing channel is applied to the \(\text{exRec}\).

2. We would like to calculate the probability of the event \(\text{mal}_E\) conditioned on acceptance of all ancillas in the LEC and TEC circuits. For every location in the ancilla state preparation and verification circuits, we insert Pauli errors according to the depolarizing error model described above. We propagate the errors through the encoding and verification circuits of \(\ket{0}\) and \(\ket{\overline{\text{T}}}\). If no errors are detected at the ancilla measurement locations of the EC’s, we record the errors that lead to acceptance in the matrices \(M_{\ket{0}}\) and \(M_{\ket{\overline{\text{T}}}\}}\). Each row of \(M_{\ket{0}}\) will correspond to an error of the form \(e = [k, i, l, t]\) where \(k\) corresponds to the error type, \(i\) and \(l\) encode the logical and physical qubit number and \(t\) is the particular time step where the error occurred. Note that since an ensemble of errors can combine leading to acceptance of the ancillas, the rows of \(M_{\ket{0}}\) will be grouped into several blocks. We repeat this process until the number of blocks of \(M_{\ket{0}}\) reaches a predefined size (in most cases, we chose the size to be \(10^6\) as we believe it to be an accurate representation of the noise for the state preparation circuits).

3. For all the remaining locations of the 1-exRec (excluding the state preparation encoding and verification circuits), we insert errors according to the depolarizing noise model. Furthermore, we randomly pick a block from the matrices \(M_{\ket{0}}\) and \(M_{\ket{\overline{\text{T}}}\}}\). The combined errors propagate through the 1-exRec and we project the final output errors back onto the codespace. Steps 1-3 are repeated \(N\) times.

4. For single qubit gates, the logical errors are recorded into the vector \(v_i = [a_x, a_y, a_z]\) where, for example, \(a_x\), corresponds to the number of logical \(X\) errors that occurred after \(N\) iterations. For a two-qubit gate, the errors are recorded into a vector with 15 columns, one for each error type. For a gate \(G\) at a physical error rate \(p\), the estimate of the probability of the event \(\text{mal}_E\) is given by \(\text{Pr}(\text{mal}_E|G, p) = a_E/N\). Hence, larger values of \(N\) will lead to better estimates of \(\text{Pr}(\text{mal}_E|G, p)\) by reducing the standard deviation.

For a 1-exRec encoding a logical gate \(G\), the pseudo-threshold is defined as the crossing point \(p = p_G(p)\), where \(p_G(p) = \sum_{E_i} \text{Pr}(\text{mal}_{E_i}|G, p)\) for all possible logical errors \(E_i\) for a given logical gate \(G\). The pseudo-threshold thus corresponds to the physical error rate below which the logical error rate is smaller than the physical error rate. To obtain the asymptotic threshold, we first upper bound \(\text{Pr}(\text{mal}_{E_i}^{(1)}|G, p)\) (the probability of a malignant event \(\text{mal}_E\) at the first level of concatenation) by

\[
\text{Pr}(\text{mal}_{E_i}^{(1)}|G, p) \leq \sum_{k=\lceil\frac{d}{2}\rceil}^{L_G} c(k)p^k \equiv \Gamma_{G, E_i}^{(1)},
\]

where the coefficients \(c(k)\) are positive integers that parametrize the number of possible weight-\(k\) errors that can lead to a logical fault, \(L_G\) is the total number of circuit locations in the logical gate \(G\) and \(d^*\) characterizes the minimal distance of a given logical gate. As was shown in Ref. [9], \(\Gamma_{G, E_i}^{(1)}(p)\) is a polynomial that is monotonically increasing as a function of the physical error rate and serves as an upper bound for the failure probability at the first level of concatenation. Following Refs. [9, 11], to obtain the probability of having the event \(\text{mal}_E\) at the second level of concatenation, we can treat each level-1 exRec in the level-2 simulation as a physical location with a modified noise model (no longer depolarizing) given by the \(\Gamma_{G, E_i}^{(1)}(p)\) terms. This procedure can be
Asymptotic threshold

| Gate          | Pseudo-Threshold | Asymptotic threshold |
|---------------|------------------|----------------------|
| CNOT gate     | (2.11 ± 0.02) × 10⁻³ | (1.95 ± 0.01) × 10⁻³ |
| T gate        | (4.89 ± 0.11) × 10⁻⁴ | (1.58 ± 0.02) × 10⁻³ |
| Hadamard gate | (4.47 ± 0.29) × 10⁻⁵ | (1.28 ± 0.02) × 10⁻³ |
| 105-qubit     | (4.47 ± 0.29) × 10⁻⁶ | (1.28 ± 0.02) × 10⁻³ |

TABLE V: Lower bounds for the pseudo and asymptotic threshold results for the Hadamard, T gate and CNOT gates of the 105-qubit code. The Hadamard asymptotic-threshold is larger than its pseudo-threshold resulting from the double protection of the CNOT gates as seen by the high CNOT pseudo-threshold.

generalized to the $k$-th level of concatenation, enabling the upper bound on $\Pr[\text{mal}]_{E}^{(k)}[G,p]$ to be given by

$$\Pr[\text{mal}]_{E}^{(k)}[G,p] \leq \sum_{l=\lceil \frac{k}{m} \rceil}^{L_G} c(l)(\Gamma_{G,E}^{(k)})^{l} \equiv \Gamma_{G,E}^{(k)},$$

where the $c(l)$ coefficients are the same as those in Eq. 19. It was then showed in Ref. [11] that for physical error rates $p$ smaller than the crossing point between $\Gamma_{G,E}$ and $\Gamma_{G}^{(2)}$ (which we define to be $p_{th,G}$), the logical error rates for the $m$-th level of concatenation ($m \geq 2$) could be upper bounded by

$$\Pr[\text{mal}]_{E}^{(m)}[G,p] \leq \Gamma_{G,E}^{(m)} \leq e^{\left[ \frac{d^*}{T}\right] m-2} \frac{1}{2} \Gamma_{G,E}^{(1)}.$$ 

Due to the exponential suppression seen in Eq. 21 of the logical error rate for $p \leq p_{th,G}$, this serves as a lower bound for the asymptotic threshold of the logical gate $G$.

In Ref. [11], the pseudo and asymptotic thresholds for the 105-qubit code were calculated for the $H$, $T$ and CNOT gates. Note that $\Gamma_{G,E}^{(m)}$ was also calculated for all other location types (storage, measurement and state-preparation) and were shown to have much higher threshold than the logical gates. The results are summarized in Table V, and Fig. 9 illustrates the noise behaviour of the Hadamard and CNOT gates for several concatenation levels. An important feature of these results is that, for the $H$ and $T$ gates, noise can be further suppressed by several orders of magnitude even for physical error rates above the pseudo-threshold. To understand this type of noise behaviour, it is important to point out that since the CNOT gate is transversal in both the 7 and 15-qubit codes, it receives a double protection from both codes. The double protection results in a much larger pseudo-threshold for the CNOT gate compared to the pseudo-thresholds for the $H$ and $T$ gates. Furthermore, the asymptotic threshold of the CNOT is comparable to its pseudo-threshold. Another important feature is that CNOT gates are the gates that are most present in the Steane’s EC circuits as well as in the logical $H$ and $T$ gate circuits. Consequently, when going to higher levels of concatenation, and for error rates below the CNOT asymptotic threshold, all the CNOT gates will be less likely to fail compared to the previous level of concatenation. Even if the physical $H$ and $T$ gate locations are more likely to fail, the lower logical failure rates of the CNOT gates will compensate, resulting in overall further noise suppression.

![FIG. 9: Probability $\Pr[\text{mal}]_{E}$ of logical error as function of physical error rate $p$ for the level-1, level-2 and level-3 logical (a) Hadamard and (b) CNOT gates of the 105-qubit code. The crossing point between the level-1 and level-2 curves allows for the determination of a lower bound for the asymptotic threshold for each of the logical gates. The CNOT gate exhibits a much lower logical error rate than the Hadamard at the first level.](image)

We now provide the threshold results for the 49-qubit code. The plots in Fig. 15 illustrate $\Gamma_{G,E}^{(m)}$ for the Hadamard, $T$ and CNOT gates. As was the case for the 105-qubit code, the pseudo-threshold of the 49-qubit code is limited by the Hadamard gate and is given by $p_{th}^{(1)} = (7.76 ± 0.17) \times 10^{-5}$. A lower bound on the asymptotic threshold was found to be $p_{th} = (9.69 ± 0.28) \times 10^{-5}$. However, Fig. 15(c) shows a few noticeable differences with the 49-qubit code. First, logical $Z$ errors are what limit the Hadamard asymptotic threshold for the 49-qubit code, not logical $X$ errors as in the 105-qubit code. Second, the Hadamard circuit is less sensitive to input $Z$ errors than in the 105-qubit code. Furthermore, the...
Asymptotic threshold (1)

IV. RESOURCE OVERHEAD FOR THE 49 AND 105-QUBIT CODE

A. Raw qubit overhead

In the simulation of a particular gate (say H or CNOT), we would like to obtain the resource overhead required to achieve a particular target logical error rate $p_{\text{target}}$. The overhead can be measured in several ways. The raw qubit overhead measures how many physical qubits are required in the simulation of a logical gate to achieve a particular target logical error rate. The gate overhead measures the total number of gates used in the simulation of a logical gate in order to achieve a particular target logical error rate.

We begin with the analysis of the raw qubit overhead. Recall that Steane error correction (EC) is implemented by the circuit in Fig. 1 where the ancilla states $|0\rangle$ and $|\mp\rangle$ (for the 49 and 105-qubit code) are prepared using the circuits of section IV D.

We assume that the qubits used in the preparation of the ancilla states can be reused at each time step in the computation prior to a measurement in the X or Z basis. Before proceeding with the overhead calculation, we provide a few definitions. Let $|0\rangle^{(k)}$ and $|\mp\rangle^{(k)}$ corre-

| Gate      | Pseudo-Threshold      | Asymptotic threshold      |
|-----------|-----------------------|--------------------------|
| CNOT gate | $(1.21 \pm 0.04) \times 10^{-3}$ | $(1.10 \pm 0.01) \times 10^{-3}$ |
| $T$ gate  | $(4.18 \pm 0.24) \times 10^{-4}$ | $(1.03 \pm 0.03) \times 10^{-3}$ |
| Hadamard gate | $(7.76 \pm 0.17) \times 10^{-5}$ | $(9.69 \pm 0.28) \times 10^{-4}$ |
| 49-qubit  | $(7.76 \pm 0.17) \times 10^{-6}$ | $(9.69 \pm 0.28) \times 10^{-4}$ |

TABLE VI: Lower bounds for the pseudo and asymptotic threshold results for the Hadamard, $T$ gate and CNOT gates of the 49-qubit code. The Hadamard asymptotic-threshold is larger than its pseudo-threshold resulting from the double protection of the CNOT gates as seen by the high CNOT pseudo-threshold.

49-qubit Hadamard pseudo-threshold is larger than for the 105-qubit code. To understand these results, we first point out that the 49-qubit code Hadamard circuit contains four blocks consisting only of physical $H$ and storage gates (no CNOT gates). As was explained in section IV D, the cascading sequence of CNOT gates in the 15-qubit Hadamard circuit makes it very sensitive to input $Z$ errors since any $Z$ error that lands on the target of the CNOT gates (appearing before the physical $H$ gate) will lead to a logical $X$ error. Furthermore, $Z$ errors occurring at CNOT or storage locations within the 15-qubit Hadamard circuit also play a dominant role in producing a logical $X$ fault at the output of the circuit. The four blocks consisting only of physical $H$ and storage gates will treat $X$ and $Z$ errors on the same footing and these blocks contain much fewer locations were $Z$ errors can occur. A numerical simulation also showed that many of the errors leading to a logical $Z$ fault were $Z$ errors that occurred on CNOT gates after the physical $H$ gate combined with $Y$ errors on one of the single-qubit codeblocks. We also note that for higher concatenation levels, given the smaller number of locations in a storage gate exRec, storage gates are less likely to fail than CNOT gates and so the single qubit codeblocks are less likely to acquire a logical fault. Consequently, the 49-qubit Hadamard circuit will produce less logical $X$ errors compared to the circuit for the 105-qubit code. Since the 15-qubit code offers less protection against $Z$ errors, logical $Z$ errors become the dominant source of error for the Hadamard circuit.

Note that as in the 105-qubit code, the double protection of the CNOT gates from both the 7 and 15-qubit code results in a larger pseudo and asymptotic threshold relative to the $H$ and $T$ gates. The threshold results for the 49-qubit code are summarized in Table VI. Although the 49-qubit code Hadamard pseudo-threshold is larger than the 105-qubit code pseudo-threshold, the asymptotic threshold is slightly smaller due to the lower CNOT pseudo and asymptotic threshold.
spond to the state-preparation circuits of $|0\rangle$ and $|+\rangle$ at the $k$-th level of concatenation. In the first part of the Steane EC circuit, the ancilla states are verified for errors by entangling a pair of ancilla states of the same type and performing a measurement in the appropriate basis (see section IV D for more details). We define $n^{(k)}_{\overset{\scriptscriptstyle \mathcal{E}}{\mathcal{E}}}$ and $n^{(k)}_{\mathcal{T}}$ to be the number of qubits required for the ancillas $|0\rangle^{(k)}$ and $|\mathcal{T}\rangle^{(k)}$ to pass the verification test in the circuits of Fig. 10(a) and Fig. 10(b) at the $k$-th level of concatenation. If an error is detected from the syndrome measurement, the ancilla is rejected and the process is repeated using a fresh batch of qubits which we assume are readily available. The probabilities of acceptance $p^{(k)}_{|0\rangle}$ and $p^{(k)}_{|\mathcal{T}\rangle}$ were computed from a Monte Carlo algorithm. For an error correcting code with $n$ physical qubits, we have at the first level of concatenation

$$n^{(1)}_{\overset{\scriptscriptstyle \mathcal{E}}{\mathcal{E}}} = \frac{2n(\frac{1}{p_{|0\rangle}} + \frac{1}{p_{|\mathcal{T}\rangle}})}{p^{(1)}_{|0\rangle}}, \quad (22)$$

$$n^{(1)}_{\mathcal{T}} = \frac{2n(\frac{1}{p_{|0\rangle}} + \frac{1}{p_{|\mathcal{T}\rangle}})}{p^{(1)}_{|\mathcal{T}\rangle}}. \quad (23)$$

Note that the term $2n/p^{(1)}_{|j\rangle}$ (where $j \in \{0, 1, 2\}$) corresponds to the expected number of qubits for preparing a $|j\rangle$ state free of $X$ errors (the first step in the ancilla verification test). Hence Eq. 22 corresponds to the expected number of qubits for the full $|0\rangle$ ancilla verification test. The analogous equation holds for $|\mathcal{T}\rangle$ state preparation. Defining $n^{(k)}_{\mathcal{EC}}$ to be the raw qubit overhead for a Steane EC circuit at level-$k$ (excluding the overhead from the data qubits), we have that

$$n^{(1)}_{\mathcal{EC}} = n^{(1)}_{\overset{\scriptscriptstyle \mathcal{E}}{\mathcal{E}}} + n^{(1)}_{\mathcal{T}}. \quad (24)$$

Since a logical CNOT gate consists of four EC circuits and two logical qubits, the level-1 overhead for the CNOT gate is

$$q^{(1)}_{\mathcal{CNOT}} = 2n^{(1)}_{\mathcal{EC}} + 2n. \quad (25)$$

Note that the factor of 2 and not 4 in front of the $n_{\mathcal{EC}}$ term. This is due to the fact that we assume that the qubits used in the LEC circuit can be reused in the TEC circuits. Similarly, the level-1 Hadamard qubit overhead is given by

$$q^{(1)}_{\mathcal{H}} = n^{(1)}_{\mathcal{EC}} + n. \quad (26)$$

At the second level of concatenation, $|0\rangle^{(2)}$ will contain $|0\rangle^{(1)}$ and $|\mathcal{T}\rangle^{(1)}$ state preparation circuits which will be followed by an EC circuit. However, all other gates (CNOT’s and storage) will also contain their respective LEC and TEC circuits (since recall that for a level-2 simulation each physical gate is replaced by a level-1 exRec). The EC circuits of $|0\rangle^{(1)}$ and $|\mathcal{T}\rangle^{(1)}$ will overlap with the LEC circuit of the gate that follows (see Fig. 11) and so it is important to take into account the overhead of the overlapping EC circuit. The full EC circuits (including the data qubits) each have an overhead of $n^{(1)}_{\mathcal{EC}} + n$. We assume that the qubits that were used in the EC circuit following $|0\rangle^{(1)}$ and $|\mathcal{T}\rangle^{(1)}$ can be reused for all other EC circuits that follows. Hence, we only take into account the overhead of the first EC circuit. The entire $|0\rangle^{(2)}$ circuit will have an overhead of $n(n^{(1)}_{\mathcal{EC}} + n)$. Generalizing to the $k$-th concatenation level (for $k \geq 2$), we have the recursive relation

$$n^{(k)}_{\overset{\scriptscriptstyle \mathcal{E}}{\mathcal{E}}} = \frac{2n(n^{(k-1)}_{\mathcal{EC}} + n^{(k-1)})}{p^{(k)}_{|0\rangle}} + \frac{1}{p^{(k)}_{|0\rangle}} \quad (27)$$

$$n^{(k)}_{\mathcal{T}} = \frac{2n(n^{(k-1)}_{\mathcal{EC}} + n^{(k-1)})}{p^{(k)}_{|\mathcal{T}\rangle}} + \frac{1}{p^{(k)}_{|\mathcal{T}\rangle}}. \quad (28)$$

The EC circuit at level-$k$ has an overhead given by

$$n^{(k)}_{\mathcal{EC}} = n^{(k)}_{\overset{\scriptscriptstyle \mathcal{E}}{\mathcal{E}}} + n^{(k)}_{\mathcal{T}}. \quad (29)$$

The CNOT and Hadamard overhead at level-$k$ are then given by

$$q^{(k)}_{\mathcal{CNOT}} = 2n^{(k)}_{\mathcal{EC}} + 2nk, \quad (30)$$

$$q^{(k)}_{\mathcal{H}} = n^{(k)}_{\mathcal{EC}} + nk. \quad (31)$$

FIG. 11: Typical segment in a level-2 $|0\rangle^{(2)}$ or $|\mathcal{T}\rangle^{(2)}$ illustrating the overlapping EC circuits between the $|0\rangle^{(1)}$ (could also be $|\mathcal{T}\rangle^{(1)}$) and the following logical gate (a CNOT gate in this example). The overhead of the first EC circuit needs to be taken into account in the calculation of $n^{(2)}_{\mathcal{EC}}$ and $n^{(2)}_{\mathcal{EC}}$. Each dashed box has an overhead of $n^{(1)}_{\mathcal{EC}} + n$. 

![Diagram](image_url)
head for the $|0\rangle$ ancilla verification components, $n^{(1)}_{g,|\overline{0}\rangle_j}$ to be the gate overhead for the $|\overline{T}\rangle$ ancilla verification components and $n^{(1)}_{\text{rem}}$ to be the gate overhead for the remaining EC circuit, see Fig. 12 for a circuit description.

Since CNOT and measurement locations can be implemented transversally, they each contribute a factor of $n$ to the gate overhead at the first level so that

$$n^{(1)}_{g,|\overline{0}\rangle_1} = 2(n^{(1)}_{g,|\overline{0}\rangle_2} + n^{(1)}_{g,|\overline{0}\rangle_3}) = 2(2g^{(1)}_{|\overline{0}\rangle} + n),$$

$$n^{(1)}_{g,|\overline{T}\rangle_1} = 2(n^{(1)}_{g,|\overline{T}\rangle_2} + n^{(1)}_{g,|\overline{T}\rangle_3}) = 2(2g^{(1)}_{|\overline{T}\rangle} + n),$$

$$n^{(1)}_{g,|0\rangle_1} = 2n,$$

$$n^{(1)}_{\text{rem}} = 6n.$$  \hfill (37)

Taking into account that the ancilla states are rejected if a non-trivial error is detected at the measurement locations, we can compute the expected number of gates for a $|\overline{0}\rangle$ and $|\overline{T}\rangle$ verification test (defined as $n^{(1)}_{g,\text{EC}|\overline{0}\rangle}$ and $n^{(1)}_{g,\text{EC}|\overline{T}\rangle}$) according to their success probabilities as

$$n^{(1)}_{g,\text{EC}|\overline{0}\rangle} = \frac{n^{(1)}_{g,|\overline{0}\rangle_1} \left( \frac{p^{(1)}_{|\overline{0}\rangle_1}}{p^{(1)}_{|\overline{0}\rangle_2}} + \frac{p^{(1)}_{|\overline{0}\rangle_3}}{p^{(1)}_{|\overline{0}\rangle_2}} \right) + n^{(1)}_{g,|\overline{0}\rangle_3}}{p^{(1)}_{|\overline{0}\rangle_2}},$$

$$n^{(1)}_{g,\text{EC}|\overline{T}\rangle} = \frac{n^{(1)}_{g,|\overline{T}\rangle_1} \left( \frac{p^{(1)}_{|\overline{T}\rangle_1}}{p^{(1)}_{|\overline{T}\rangle_2}} + \frac{p^{(1)}_{|\overline{T}\rangle_3}}{p^{(1)}_{|\overline{T}\rangle_2}} \right) + n^{(1)}_{g,|\overline{T}\rangle_3}}{p^{(1)}_{|\overline{T}\rangle_2}}. \hfill (38)$$

From Eqs. 37 and 39, the total gate overhead $n^{(1)}_{g,\text{EC}}$ for an EC circuit at the first level of concatenation is given by

$$n^{(1)}_{g,\text{EC}} = n^{(1)}_{g,\text{EC}|\overline{0}\rangle} + n^{(1)}_{g,\text{EC}|\overline{T}\rangle} + n^{(1)}_{\text{rem}}.$$ \hfill (40)

Using Eq. 40, the overhead of a level-1 CNOT and storage exRec is given by

$$g^{(1)}_{\text{CNOT}} = 4n^{(1)}_{g,\text{EC}} + n,$$

$$g^{(1)}_{\text{mem}} = 2n^{(1)}_{g,\text{EC}} + n.$$ \hfill (41) \hfill (42)

The logical Hadamard circuit consists of storage, CNOT and physical Hadamard gates. Defining $\gamma_\gamma j$ to be the number of gates of type $j$ in the level-1 logical Hadamard circuit and using Eq. 42, the gate overhead for the level-1 Hadamard exRec is

$$g^{(1)}_{\text{Had}} = 2n^{(1)}_{g,\text{EC}} + \gamma_\gamma j \text{CNOT} + \gamma_\gamma j \text{mem} + \gamma_\gamma j \text{Had}.$$ \hfill (43)

Before obtaining the overhead at higher levels of concatenation, it is important to point out that consecutive exRec’s will have overlapping EC’s. In a recursive calculation, if we were to use the overhead terms $g^{(k-1)}_j$ at the $k$-th concatenation level in the counting procedure,
we would be over counting the gates appearing in overlapping EC’s (see Fig. 13). The over counting can be avoided by ignoring the gate overhead in LEC’s from the $(k-1)$-exRec’s appearing in a $k$-exRec. For a $t$-qubit gate, we define

$$
\tilde{g}^{(k)}_j = g^{(k)}_j - tr^{(k)}_{g,EC}. \tag{44}
$$

Therefore, in the overhead counting of a $k$-exRec, we will include the contributions from $\tilde{g}^{(k-1)}_j$ terms instead of $g^{(k-1)}_j$.

We now consider the gate overhead at the $k$-th concatenation for $k \geq 2$. The gate overhead in a $|0\rangle$ and $|\bar{1}\rangle$ circuit at level-$k$ is given by

$$
g^{(k)}_{\bar{0}} = \alpha_{\text{CNOT}} \tilde{g}^{(k-1)}_{\text{CNOT}} + \alpha_{\text{mem}} \tilde{g}^{(k-1)}_{\text{mem}} + \alpha_{|0\rangle} \tilde{g}^{(k-1)}_{|0\rangle} + \alpha_{|+\rangle} \tilde{g}^{(k-1)}_{|+\rangle} + n^{(k)}_{g,EC}, \tag{45}
$$

$$
g^{(k)}_{\bar{1}} = \beta_{\text{CNOT}} \tilde{g}^{(k-1)}_{\text{CNOT}} + \beta_{\text{mem}} \tilde{g}^{(k-1)}_{\text{mem}} + \beta_{|0\rangle} \tilde{g}^{(k-1)}_{|0\rangle} + \beta_{|+\rangle} \tilde{g}^{(k-1)}_{|+\rangle} + n^{(k)}_{g,EC}. \tag{46}
$$

Note that the $n^{(k-1)}_{g,EC}$ term in Eq. 46 was included to take into account the gate overhead from the level-$(k-1)$ EC circuit that precedes $|0\rangle^{(k)}$ and $|\bar{1}\rangle^{(k)}$.

In the first part of the $|0\rangle$ ancilla verification test, we must include the contributions from the two $|0\rangle^{(k)}$ circuits, the CNOT exRec and the measurement locations. Since the EC circuit prior to performing a measurement is included in the contribution from the TEC of the CNOT gate, level-$k$ measurement locations will always have a gate overhead of $n^k$ so that

$$
n^{(k)}_{g,|0\rangle_1} = n^{(k)}_{g,|0\rangle_2} = 2g^{(k)}_{|0\rangle_1} + n^{(k)}_{g,\text{CNOT}} + n^k, \tag{47}
$$

$$
n^{(k)}_{g,|\bar{1}\rangle_1} = n^{(k)}_{g,|\bar{1}\rangle_2} = 2g^{(k-1)}_{|\bar{1}\rangle_1} + n^{(k-1)}_{g,\text{CNOT}} + n^k, \tag{48}
$$

$$
n^{(k)}_{g,|0\rangle_3} = n^{(k)}_{g,|\bar{1}\rangle_3} = n^{(k-1)}_{g,\text{CNOT}} + n^k, \tag{49}
$$

$$
n^{(k)}_{\text{rem}} = 2n^{(k)}_{g,\text{CNOT}} + n^{(k)}_{g,\text{mem}} + 2n^k. \tag{50}
$$

The overhead for the full ancilla verification procedure is obtained in the same way as in Eq. 39:

$$
n^{(k)}_{g,EC|\bar{0}\rangle} = \frac{n^{(k)}_{g,|\bar{0}\rangle_1} \left(\frac{1}{p^{(k)}_{|\bar{0}\rangle_1}} + \frac{1}{p^{(k)}_{|\bar{1}\rangle_2}}\right) + n^{(k)}_{g,|\bar{1}\rangle_2}}{p^{(k)}_{|\bar{0}\rangle_2}}, \tag{51}
$$

$$
n^{(k)}_{g,EC|\bar{1}\rangle} = \frac{n^{(k)}_{g,|\bar{1}\rangle_1} \left(\frac{1}{p^{(k)}_{|\bar{0}\rangle_1}} + \frac{1}{p^{(k)}_{|\bar{1}\rangle_2}}\right) + n^{(k)}_{g,|\bar{1}\rangle_2}}{p^{(k)}_{|\bar{1}\rangle_2}}. \tag{52}
$$

The overhead for the complete EC circuit at level-$k$ is then the sum of the contributions from $n^{(k)}_{g,EC|\bar{0}\rangle}$, $n^{(k)}_{g,EC|\bar{1}\rangle}$ and $n^{(k)}_{\text{rem}}$,

$$
n^{(k)}_{g,EC} = n^{(k)}_{g,EC|\bar{0}\rangle} + n^{(k)}_{g,EC|\bar{1}\rangle} + n^{(k)}_{\text{rem}}. \tag{53}
$$

It is then straightforward to obtain the overhead for the CNOT, storage and Hadamard exRecs, which we state below:

$$
g^{(k)}_{\text{CNOT}} = 4n^{(k)}_{g,EC} + n^{(k-1)}_{\text{CNOT}}, \tag{54}
$$

$$
g^{(k)}_{\text{mem}} = 2n^{(k)}_{g,EC} + n^{(k-1)}_{\text{mem}}, \tag{55}
$$

$$
g^{(k)}_{\text{Had}} = 2n^{(k)}_{g,EC} + \gamma^{(k)}_{\text{CNOT}} \tilde{g}^{(k-1)}_{\text{CNOT}} + \gamma^{(k)}_{\text{mem}} \tilde{g}^{(k-1)}_{\text{mem}} + \gamma^{(k)}_{\text{Had}} \tilde{g}^{(k-1)}_{\text{Had}}. \tag{56}
$$

C. 49 and 105-qubit code overhead results

In this section we use the formalism of section VI A and VI B to obtain the raw qubit and gate overhead results of the 49 and 105-qubit codes. Since the Hadamard gate limits the threshold value of both codes, we will focus on the overhead for performing a logical Hadamard and CNOT gate. Given a target logical error rate $p_{\text{target}}$, we can use the threshold results of Section V to determing the appropriate level of concatenation to reach $p_{\text{target}}$. We can then calculate the raw qubit overhead and gate overhead from Eqs. 30–31 and 54–56, respectively. The coefficients $\alpha_j$, $\beta_j$ and $\gamma_j$ are given in Section IV D.

Figs. 16 and 17 illustrate the physical qubit and gate overhead for the 49 and 105-qubit codes. The key results are summarized in Table VII, VIII, IX and X.
In this Section, we aim to estimate the qubit (space) overhead for the implementation of fault-tolerant logic in the surface code. The primary obstacle to surface code is the physical error rate $p$ that can be achieved for the particular concatenation level so that the logical error rate is below $p_{\text{target}} = 10^{-15}$. For the 49-qubit logical Hadamard gate, if we limit the implementation to two levels of concatenation it would require a level of precision of $2.36 \times 10^{-6}$, far below the asymptotic threshold, in order to reduce the logical error rate to $10^{-15}$. The raw qubit and gate overheads are given by $(1.75 \pm 0.38) \times 10^5$ and $(4.20 \pm 0.92) \times 10^7$ in such a scenario. Going to a third level of concatenation allows for higher physical error rates, up to $8.47 \times 10^{-5}$, as the increase concatenation level will further reduce logical error rate. However there is a tradeoff to increasing the concatenation level as it requires further resources. The raw qubit and gate overheads for an error rate of $8.47 \times 10^{-5}$ are given by $(7.33 \pm 1.57) \times 10^7$ and $(2.14 \pm 0.66) \times 10^{11}$ (an increase of roughly three orders of magnitude). One aspect that is quite apparent in the overhead results is how well the 49-qubit logical Hadamard performs compared to the 105-qubit version. For the 105-qubit code, the logical Hadamard gate can achieve the desired target error rate at the second concatenation level for physical error rates below $5.35 \times 10^{-7}$, as opposed to $2.36 \times 10^{-6}$, due to the complexity of the 105-qubit Hadamard construction compared to that of the 49-qubit code. The third level of concatenation pushes this number to $1.60 \times 10^{-5}$. Furthermore, for each concatenation level the overheads are roughly an order of magnitude larger than for the 49-qubit logical Hadamard.

### Comparison to the surface code overhead using state distillation

In this Section, we aim to estimate the qubit (space) overhead for the implementation of fault-tolerant logic in the surface code. The primary obstacle to surface code

| Level of concatenation | Hadamard gate | Physical error rate (49) | Qubit overhead (49) | Gate overhead (49) |
|------------------------|---------------|--------------------------|---------------------|-------------------|
| 1                      | N/A           | (1.75 ± 0.38) × 10^5    | (4.20 ± 0.92) × 10^7 |
| 2                      | 2.36 × 10^{-6} | (3.60 ± 0.79) × 10^5    | (8.38 ± 0.18) × 10^7 |
| 3                      | 8.47 × 10^{-5} | (1.94 ± 0.37) × 10^8    | (5.45 ± 1.50) × 10^{11} |

### CNOT overhead results for the 49-qubit code

| Level of concatenation | CNOT gate | Physical error rate (49) | Qubit overhead (49) | Gate overhead (49) |
|------------------------|-----------|--------------------------|---------------------|-------------------|
| 1                      | N/A       | (1.14 ± 0.22) × 10^6    | (1.86 ± 0.34) × 10^8 |
| 2                      | 5.24 × 10^{-5} | (3.00 ± 0.47) × 10^9    | (4.82 ± 0.73) × 10^{12} |
| 3                      | 3.92 × 10^{-4} | (2.27 ± 0.44) × 10^6    | (3.55 ± 0.66) × 10^9 |

### CNOT overhead results for the 105-qubit code

| Level of concatenation | Hadamard gate | Physical error rate (105) | Qubit overhead (105) | Gate overhead (105) |
|------------------------|---------------|--------------------------|---------------------|-------------------|
| 1                      | N/A           | (6.01 ± 0.94) × 10^5    | (9.28 ± 1.40) × 10^{12} |
| 2                      | 4.56 × 10^{-4} | (2.27 ± 0.44) × 10^6    | (3.55 ± 0.66) × 10^9 |
| 3                      | 1.39 × 10^{-3} | (6.01 ± 0.94) × 10^5    | (9.28 ± 1.40) × 10^{12} |
code implementations is the need to distill a special ancillary state for the purposes of implementing the $T$ gate through gate teleportation. High-fidelity logical state preparation of this special state is obtained through a process called state distillation. State distillation is implemented using only logical Clifford gates, resulting in a non-stabilizer state that can be used to implement the $T$ gate, called a magic state \[21\].

The idea behind magic state distillation begins by assuming the initial magic state has an error rate $p$. Then by using multiple noisy logical magic states and near-perfect Clifford operations (since the Clifford operations are performed using fault-tolerant logical operations in the surface code) a higher-fidelity magic state can be distilled from multiple noisy states. Depending on the distillation scheme used, the output state of the scheme will have a fidelity of $c p^b$, for some constant value of $b$ and $c$. The process can then be repeated with multiple copies of the newly distilled logical states to obtain a state with error rate $c (cp^b)^k = c^{b+1} p^{b^2}$. Iterating this process $k$ times, the final logical output state will have an error rate:

$$p_k = \frac{1}{e^{c^2 p}} \left( \frac{1}{c^{b+1} p} \right)^b.$$ \[57\]

The magic state distillation process can be probabilistic, yielding a distilled state based on the result of a set of measurements, and in general will succeed with probability $1/r$ (where $r$ depends on the particular distillation scheme being used). Therefore, if $n$ logical qubits are required for the distillation, and the probability of success for a given round is $1/r$, the total number of logical qubits required for $k$ distillation rounds is given by $(rn)^k$. The number of qubits required is thus exponential in the number of rounds, however the scheme remains theoretically efficient as the logical error rate is suppressed double-exponentially. Table XI summarizes the different parameters for the magic state distillation schemes.

Suppose we would like to obtain a lower bound on the number of physical qubits that are required to implement the logical $T$ gate in the surface code. Given a target error rate $p_{\text{target}}$, and a depolarizing physical error rate of $p$, we must first determine the number of distillation levels required to obtain the desired target rate, that is choose a value of $k$ from Eq. 57 such that $p_k < p_{\text{target}}$. Having determined the level of distillation, recall that we argued that the Clifford gates must have low levels of noise with respect to the target error rate. Therefore, for a given distillation level, we will need to choose logical gates that have a small enough logical error rate. For the surface code, by considering the probability of a logical string being created given that the syndrome extraction scheme is $8$ time steps long, the logical gate error rate can be approximated as \[7\]:

$$p_L(d) = d \left( \frac{d}{[d/2]} \right) (8p)^{[d/2]}$$ \[58\],

where $d$ is the distance of the code. Therefore, if the output of a given distillation level is given by $p_k$, as argued in Ref. \[7\], the logical error rate must be small enough such that the resulting accumulation of errors from the distillation circuit does not negatively affect the distilled qubit, that is:

$$8 \cdot 1.25 \cdot n_q \cdot d \cdot (8p)^{[d/2]} \leq p_k.$$ \[59\]

where $n_q$ is the number of qubits in the distillation scheme. Therefore, the distance must be chosen large enough to reduce the logical error rate to sufficiently small levels (assuming we are always below threshold). Having distilled at a given level, the logical qubits can be enlarged through a fault-tolerant growing operation, in order to be of the required distance for the next distillation level. In our lower bound of the number of qubits required, we will assume that this operation is done error free, in reality this may slightly increase the number of physical qubits required. Therefore given the required distance at each level, and the fact that the number of physical qubits for a distance $d$ surface code is $(2d)^2$ (including the syndrome qubits), a lower bound of the distillation overhead can be obtained. Figure 14 illustrates the physical qubit overhead for two different distillation schemes used in the implementation of the $T$ gate for the surface code. It can be seen that for a target logical error rate of $p_{\text{target}} = 10^{-15}$, both schemes yield an overhead on the order of $10^4$ physical qubits for input error rates $10^{-4} < p < 10^{-3}$.

### VII. CONCLUSION

In this paper we reviewed the fault-tolerant construction of the 49 and 105-qubit codes obtained from concatenating Steane’s 7-qubit code with the 15-qubit Reed-Muller code. One advantage of the concatenation scheme is that universal fault-tolerance can be achieved without using state distillation protocols. Taking advantage of the CSS structure of the 7 and 15-qubit codes, the error correction blocks were constructed in the framework of Steane error correction. In the study of the performance of the concatenated codes, we obtained the threshold for the 105-qubit code for an adversarial noise model using malignant set counting and found it to be $(8.33 \pm 0.28) \times 10^{-6}$, which is slightly below that of

| Distillation Type | Qubits | Error rate | Success prob. |
|-------------------|--------|------------|---------------|
| \(|T\)\-type      | 5      | $p/2$      | 1/6           |
| \(|H\)\-type      | 15     | $1/(3\sqrt{5}p)$ | $\approx 1 - p/15$ |
| 10\(-to\)-2       | 5      | $1/(9p)^{2^k}$ | $\approx 1$ |

TABLE XI: Parameters for different magic state distillation schemes. The \(|T\)\ and \(|H\)\ schemes are given in Ref. \[21\], while the 10\(-to\)-2 qubit scheme is presented in Ref. \[22\].
We proceeded by developing general methods to compute the overhead of concatenated codes using Steane error correction and applied our methods to compute the physical qubit and gate overhead of the 49 and 105-qubit code. We also computed the physical qubit overhead for surface codes implementing the $T$ gate using the $H$-type magic state and the 10-to-2 distillation schemes [21, 22]. Comparing the plots of Fig. 14 with those of Fig. 16 and Fig. 17, it is clear that surface code requires a smaller overhead than the 49 and 105-qubit codes given the sampled input error rates. For example, for an input error rate of $p = 5 \times 10^{-5}$, the 49-qubit code requires more than $10^7$ physical qubits to implement a logical Hadamard gate compared to roughly $10^4$ physical qubits to implement the $T$ gate in the surface code.

To explain the differences in overhead, it is first important to point out that the surface code thresholds are about an order of magnitude larger than the studied concatenated code thresholds. Hence, for comparable input error rates below threshold, the logical noise rate would be further suppressed for the surface codes requiring the use of fewer qubits to achieve a particular target logical error rate, even when using more rounds of distillation. Furthermore, the size of the EC blocks for Steane error correction represent a big drawback for concatenated codes. To illustrate this, it can be seen in Fig. 16(a) that the second level of concatenation requires the use of more than $10^5$ physical qubits for the 49-qubit code when implementing a logical Hadamard gate. Since the data qubits require the use of $49^2 = 2401$ physical qubits, more than 97% of the overhead comes from the size of the EC blocks. Consequently, even if the threshold for the 49 and 105 qubit codes were significantly improved, two levels of concatenation would require more than $10^5$ physical qubits which is more than the largest number of physical qubits used in the surface code implementation of the $T$ gate for all sample error rates. The latter shows that in order to improve the overhead results of the studied concatenated codes, smaller EC blocks would need to be used that maintain the fault-tolerant properties of the concatenated scheme, even at the cost of lowering the asymptotic threshold. Additionally, it would be interesting to determine the resource overhead for the generalized construction of the concatenated model, using higher distance 2D and 3D color codes as the base codes for the implementation of the fault-tolerant universal gate set.

VIII. ACKNOWLEDGEMENTS

T. J. would like to acknowledge the support of NSERC and the Vanier-Banting Secretariat through the Vanier CGS. C. C. would like to acknowledge the support of QEII-GSST and to thank Steve Weiss for providing the necessary computational resources. This work was supported by CIFAR, NSERC, and Industry Canada.
[1] M. H. Devoret and R. J. Schoelkopf, “Superconducting circuits for quantum information: an outlook,” *Science*, vol. 339, no. 6124, pp. 1169–1174, 2013.

[2] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White, J. Mutus, A. G. Fowler, B. Campbell, et al., “Superconducting quantum circuits at the surface code threshold for fault tolerance,” *Nature*, vol. 508, no. 7497, pp. 500–503, 2014.

[3] J. M. Chow, J. M. Gambetta, E. Magesan, D. W. Abraham, A. W. Cross, B. R. Johnson, N. A. Masluk, C. A. Ryan, J. A. Smolin, S. J. Srinivasan, et al., “Implementing a strand of a scalable fault-tolerant quantum computing fabric,” *Nature communications*, vol. 5, 2014.

[4] J. Kelly, R. Barends, A. G. Fowler, A. Megrant, E. Jeffrey, T. C. White, D. Sank, J. Y. Mutus, B. Campbell, Y. Chen, et al., “State preservation by repetitive error detection in a superconducting quantum circuit,” *Nature*, vol. 519, no. 7541, pp. 66–69, 2015.

[5] M. Takita, A. D. Córcoles, E. Magesan, B. Abdo, M. Brink, A. Cross, J. M. Chow, and J. M. Gambetta, “Demonstration of weight-four parity measurements in the surface code architecture,” *arXiv:1605.01351*, 2016.

[6] D. Aharonov and M. Ben-Or, “Fault-tolerant quantum computation with constant error,” in *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pp. 176–188, ACM, 1997.

[7] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, “Surface codes: Towards practical large-scale quantum computation,” *Phys. Rev. A*, vol. 86, p. 032324, 2012.

[8] P. Aliferis, D. Gottesman, and J. Preskill, “Quantum accuracy threshold for concatenated distance-3 codes,” *Quant. Inf. Comput.*, vol. 6, pp. 97–165, 2006.

[9] A. Paetznick and B. W. Reichardt, “Fault-tolerant ancilla preparation and noise threshold lower bounds for the 23-qubit galay code,” *Quant. Inf. Compt.*, vol. 12, pp. 1034–1080, 2011.

[10] T. Jochym-O’Connor and R. Laflamme, “Using concatenated quantum codes for universal fault-tolerant quantum gates,” *Phys. Rev. Lett.*, vol. 112, p. 010505, 2014.

[11] C. Chamberland, T. Jochym-O’Connor, and R. Laflamme, “Thresholds for universal concatenated quantum codes,” *Phys. Rev. Lett.*, vol. 117, p. 010501, 2016.

[12] B. Eastin and E. Knill, “Restrictions on transversal encoded quantum gate sets,” *Phys. Rev. Lett.*, vol. 102, p. 110502, 2009.

[13] B. Zeng, A. W. Cross, and I. L. Chuang, “Transversality Versus Universality for Additive Quantum Codes,” *IEEE Transactions on Information Theory*, vol. 57, pp. 6272–6284, 2011.

[14] F. Pastawski and B. Yoshida, “Fault-tolerant logical gates in quantum error-correcting codes,” *Phys. Rev. A*, vol. 91, p. 012305, 2015.

[15] A. W. Steane, “Active stabilization, quantum computation, and quantum state synthesis,” *Phys. Rev. Lett.*, vol. 78, no. 11, p. 2252, 1997.

[16] E. Nikahd, M. Sedighi, and M. S. Zamani, “Non-uniform code concatenation for universal fault-tolerant quantum computing,” *arXiv preprint arXiv:1605.07007*, 2016.

[17] J. T. Anderson and G. Duclos-Cianci, “Private communication,” 2013.

[18] A. M. Steane, “Fast fault-tolerant filtering of quantum codewords,” *arXiv preprint quant-ph/0202036*, 2002.

[19] P. P. Aliferis, *Level reduction and the quantum threshold theorem*. PhD thesis, California Institute of Technology, 2007.

[20] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, England, 2000.

[21] S. Bravyi and A. Kitaev, “Universal quantum computation with ideal Clifford gates and noisy ancillas,” *Phys. Rev. A*, vol. 71, p. 022316, 2005.

[22] A. M. Meier, B. Eastin, and E. Knill, “Magic-state distillation with the four-qubit code,” *arXiv:1204.4221*, 2012.
FIG. 15: All plots in Fig. 15 pertain to the 49-qubit code. The plots on the left column illustrate the probability of logical error as function of physical error rate for logical (a) CNOT, (c) Hadamard and (e) T gate. The crossing point of the fitted curve allows for the determination of the level-1 pseudo-threshold for each of the logical gates. The CNOT pseudo-threshold is the largest among all three gates due to the double protection of the 7-qubit and 15-qubit code. The plots on the right column illustrate the polynomials upper bounding the probability of obtaining a logical error $E$ for the first, second and third level of concatenation. The crossing point between the level-one and level-two polynomials determine the asymptotic threshold for the gate under consideration. For the logical CNOT gate (b), it is the event $\text{mal}_Z$ which limits the threshold value. The same behaviour holds for the 105-qubit code. However, given that the effective distance of the 49-qubit code is 5 compared to 9 for the 105-qubit code (when the gate is transversal in both the 7 and 15-qubit code), the 105-qubit code offers greater suppression for logical error rates below threshold. For the logical gate $H$ (d) and $T$ gate (f), $\text{mal}_Z$ limits the threshold value. Note that for the 105-qubit code, $\text{mal}_X$ limited the Hadamard threshold value. See V for more details.
FIG. 16: Physical qubit and gate overhead log-log plots for the 49-qubit code. The numbers in parenthesis indicate the level of concatenation required to achieve the target error rates shown in the legend. The plots have a step-like function behaviour since above certain physical error rates a higher level of concatenation is required to achieve the particular target error rate. For $p_{\text{target}} = 10^{-15}$ and a physical error rate $p = 10^{-5}$, roughly $10^8$ physical qubits are required to encode one logical Hadamard gate.
FIG. 17: Physical qubit and gate overhead log-log plots for the 105-qubit code. The numbers in parenthesis indicate the level of concatenation required to achieve the target error rates shown in the legend. The plots have a step-like function behaviour since above certain physical error rates a higher level of concatenation is required to achieve the particular target error rate. For $p_{\text{target}} = 10^{-15}$ and a physical error rate $p = 10^{-5}$, roughly $10^9$ physical qubits are required to encode one logical Hadamard gate.