Concatenation Schemes for Topological Fault-tolerant Quantum Error Correction

Zhaoyi Li[1], Isaac Kim[2], and Patrick Hayden[1]

[1]Department of Physics, Stanford University, Stanford, CA 94305, USA
[2]Department of Computer Science, University of California, Davis, CA 95616, USA

2023-08-05

arXiv:2209.09390v2 [quant-ph] 6 Aug 2023

We investigate a family of fault-tolerant quantum error correction schemes based on the concatenation of small error detection or error correction codes with the three-dimensional cluster state. We propose fault-tolerant state preparation and decoding schemes that effectively convert every circuit-level error into an erasure error, leveraging the cluster state's high threshold against such errors. We find a set of codes for which such a conversion is possible, and study its performance against the standard circuit-level depolarizing model. Our best performing scheme, which is based on a concatenation with a classical code, improves the threshold by 16.5% and decreases the spacetime overhead by 32% compared to the scheme without concatenation, with each scheme subject to a physical error rate of 10^{-3} and achieving a logical error rate of 10^{-6}.

1 Introduction

Quantum computers are expected to provide substantial speedups for a number of practical problems of interest, such as factoring, quantum chemistry, and the simulation of physical models that describe nature [12, 24, 51]. However, the ubiquitous presence of noise makes qubits very fragile and easily affected by their environment, which can cause them to decohere.

The fault tolerance threshold theorem [2, 8, 17, 21, 53] provides a foundation for the long-term development of quantum computers despite the presence of noise. This theorem states that there is a finite error rate below which a noisy quantum computer can simulate a noiseless quantum computer with an error rate that can be made to be arbitrarily small, while only paying a polylogarithmic overhead in space and time. More practically, this theorem asserts that building an effectively noiseless quantum computer is in principle possible.

Zhaoyi Li: ladmon@stanford.edu
Isaac Kim: ikekim@ucdavis.edu
Patrick Hayden: phayden@stanford.edu

However, in order to build such a quantum computer, one must develop a fault-tolerant quantum computation scheme that is experimentally achievable. The figures of merit that are often considered in the literature can be divided into three categories: threshold, overhead, and connectivity. The threshold is an error rate below which scaling up the system decreases the logical error rate. As such, it is desirable to have a system with a higher threshold. Overhead is the ratio of the number of physical qubits to logical qubits, which is a metric best minimized if possible. Lastly, connectivity refers to the set of gates that can be applied between the qubits in the quantum computer. For this last category, it is desirable to have requirements compatible with the capabilities of the actual hardware.

Kitaev's surface code [21] has been the major focus of research on practical fault tolerance over the past two decades. The surface code boasts a high threshold and can be realized with qubits connected in a two-dimensional square lattice, making it the leading candidate architecture for early-generation fault-tolerant quantum computers.

A more traditional approach to fault tolerance was based on concatenated codes. While the associated codes form the major underpinning of the early theory of fault tolerance, the estimated threshold of the schemes tends to be lower; see Ref. [13] for example. However, one notable exception to this rule is Knill's fault tolerance scheme, which is based on postselection [29–31].

Postselection is a technique in which one accepts an outcome only if a certain set of criteria is satisfied. For example, in magic state distillation, one accepts the distilled magic state only if all the syndrome measurements report +1. Knill's key insight was that the threshold for postselection schemes tends to be higher than fault-tolerant quantum error correction schemes. Of course, if one relies on postselection alone, the probability of accepting the result of a post-selected computation decreases exponentially with the number of post-selected operations, making this approach impractical [29, 30]. However, what one can do is concatenate such a scheme with another code that can deal with the events in which the postselection procedure does not accept the outcome.

While Knill's scheme is based on the concatenation of a postselection scheme with a fault-tolerant concatenated quantum error correction scheme, there is a viable alternative: concatenating a postselection scheme with the surface code, or equivalently the three-dimensional (3D) cluster state. It is well-known that these schemes [54] have high thresholds against loss compared to Pauli noise. Therefore, one can hope to improve the logical error rate by converting weight-1 Pauli errors to erasure errors, by declaring loss if the postselection scheme does not accept. The main purpose of this paper is to provide a careful study of this idea.

This paper reports a comprehensive study in this direction for a variety of small quantum error correcting codes concatenated with the 3D cluster state [42, 44, 45]. The codes we employ can be broadly classified into two categories. The first category, which we refer to as Type I codes, involves codes for which the concatenation can be implemented straightforwardly using the fault-tolerant logical gates available to those codes. While we observe a general increase in threshold against the phenomenological noise model, these codes are less effective against circuit-level noise. The second category, which we refer to as Type II codes, are codes for which some of the logic gates used are not obviously fault-tolerant in the sense that a single error can propagate to multiple errors in the other code block. Somewhat surprisingly, some of these codes outperform the bcc cluster state, even against circuit-level noise.

We attribute this fact to a surprising property of our state preparation circuit: any single-qubit circuit-level noise operator as well as the two-qubit circuit-level noise operator that are implemented with the CZ gates propagate, up to stabilizers, to a tensor product of errors on disjoint code blocks which are detectable from the syndrome information within that block. In particular, upon measuring this code block, one can convert the Pauli error into an erasure error. Importantly, via this process, *every* single-qubit circuit-level noise operator propagates to an erasure error of the 3D cluster state. Because the logical gates used for the codes of Type II are not obviously fault-tolerant, the fact that we can achieve this conversion is not obvious.

We note that there has been significant recent work studying a concatenation of the 3D cluster state, or equivalently the 2D surface code, with more hardware-efficient codes [6, 23, 33, 39, 40, 58, 61, 63, 64]. While the underlying idea is similar, there is a crucial difference. The prior work exploits some special structures in the noise or concatenates with a more hardware-specific scheme, e.g., the Gottesman-Kitaev-Preskill (GKP) code. On the other hand, our scheme does not make these extra assumptions. In particular, it should be possible to concatenate these schemes with our scheme, which may lead to further improvements.

The rest of this paper is structured as follows. In Section 2, we introduce stabilizer codes, cluster states, and concatenation schemes. In Section 3, we demonstrate two types of concatenation schemes for fault-tolerant quantum error correction. In Section 4, numerical simulations are performed to study the threshold error rate of these schemes. Finally, in Section 5, we use numerical and analytical results to explain the low-error behavior of the qubit overhead.

2 Background

In this section, we review known facts about quantum error correction that are relevant to this work.

2.1 Stabilizer codes

A *stabilizer code* is a subspace spanned by the joint $+1$ eigenstates of the elements of the *stabilizer group* [25]. A stabilizer group is a commutative subgroup of the Pauli group which does not contain $-I^{\otimes n}$ as an element. Following the usual convention, we shall say a code is an $[\![n, k, d]\!]$-code if the stabilizer group is over n qubits, the number of logical qubits is k, and the code distance is d.

Stabilizer codes are a flexible, ubiquitous foundation upon which one can build schemes for fault-tolerant quantum computation [1, 26]. There are several approaches to achieve that aim, using block codes [26, 27, 55, 56], topological codes [21, 42, 44, 45], and code concatenation [1, 2, 31]. In this paper, we will investigate a hybrid of the latter two.

The topological code of particular relevance to this paper is Kitaev's surface code [28]. This code boasts a high threshold and planar connectivity, making it one of the leading approaches to building a large-scale fault-tolerant quantum computer. It is well-known that there is an intimate relationship between the surface code and the 3D cluster state, which is a resource state for fault-tolerant quantum computation [42, 44, 45].

A *cluster state* is a quantum state defined in terms of a graph $G = (V, E)$, where V is the set of vertices and E is a set of edges. More specifically, the cluster state $|\psi_G\rangle$ is defined as the unique state over $|V|$ qubits – labeled in terms of $v \in V$ – determined by the following equation:

$$X_v \left(\bigotimes_{u:(v,u)\in E} Z_u \right) |\psi_G\rangle = |\psi_G\rangle, \tag{1}$$

for all $v \in V$. Alternatively, $|\psi_G\rangle$ can be defined as:

$$|\psi_G\rangle = \prod_{(v,u)\in E} CZ_{u,v} \left(\bigotimes_{v\in V} |+\rangle_v \right), \tag{2}$$

where $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $CZ_{u,v}$ is the controlled-Z gate between u and v. Thus, the cluster state can be created by preparing all the qubits in the $|+\rangle$ state and applying a sequence of CZ gates.

The 3D cluster state [43, 45] is a cluster state defined over a graph G_{bcc}, a bcc lattice; see Figure 1. The importance of the 3D cluster state comes from the fact that it is a resource state for fault-tolerant measurement-based quantum computation. For instance, by measuring all but the bottom and top layers of Figure 1, up to a Pauli correction, we can establish a logical Bell pair between two surface code states at the bottom and top layers.

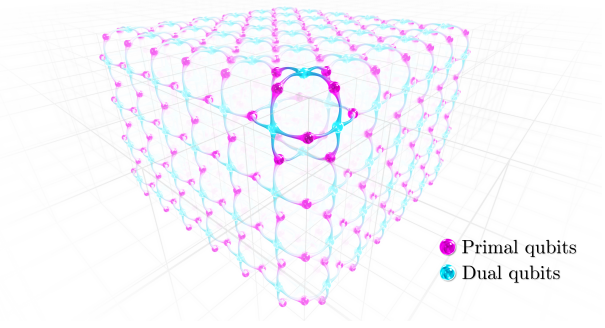


Figure 1: Illustration of the Raussendorf lattice. There are two types of qubits in the lattice, the magenta balls represent primal qubits and the cyan balls represent dual qubits. The linkage between the balls represents CZ gate applications.

It is possible to obtain the surface code from the 3D cluster state through the process of foliation [5]. Moreover, the surface code state can be fault-tolerantly teleported by measuring consecutive layers of the 3D cluster state in the X-basis [43]. In a fault-tolerant cluster state quantum computation, Z-basis measurements are only used for removing the vertices of the graph that defines the cluster state [10]. Y-basis measurements are used only when applying a logical S-gate. [11]. Therefore, for the purpose of storing a quantum information fault-tolerantly, it suffices to use only X-basis measurements.

Thus, while we shall primarily work with the 3D cluster state in this paper, the conclusions made, for instance, on the threshold and the overhead, are easily translatable to that of the surface code. We note in passing that the concatenation scheme may be applicable to other cluster states using crystalline structures [36, 38]. We leave such studies for future work.

2.2 Concatenation

In this paper, we propose a fault-tolerant quantum error correction scheme based on concatenating the 3D cluster state with a quantum error-correcting code. We aim to construct a state we denote as $|\psi_{G,\mathcal{C}}\rangle$, where $G = (V, E)$ is a graph and $\mathcal{C}$ is a stabilizer code that encodes a single logical qubit. Doing so requires performing logical operators on the logical qubit, which we will refer to as a *code block* of physical qubits. Specifically, given $\mathcal{C}$, let $|\overline{+}\rangle$ be its logical $|+\rangle$ state and let $\overline{CZ}_{u,v}$ be a logical $\overline{CZ}$ gate between two code blocks, labeled by $u, v \in V$. The state $|\psi_{G,\mathcal{C}}\rangle$ is defined as

$$|\psi_{G,\mathcal{C}}\rangle = \prod_{(u,v)\in E} \overline{CZ}_{u,v} \left(\bigotimes_{v\in V} |\overline{+}\rangle_v \right). \tag{3}$$

One can formally view $|\psi_{G,\mathcal{C}}\rangle$ as a state living in a code $\bigotimes_{v\in V} \mathcal{C}_v$.

While our main focus lies in fault-tolerant quantum error correction, let us remark that, insofar as the state in Eq. (3) can be prepared, the basic ingredients of fault-tolerant quantum computation delineated in Ref. [45] can be implemented in a straightforward way. The Clifford gates could be performed with logical $\overline{X}$ and $\overline{Z}$ measurements, assisted by an ancillary $+1$ eigenstate of the Pauli-Y operator [21]. The faulty magic state can be encoded into a code block by performing an (unprotected) magic-basis measurement [34], followed by distillation [7]. Thus, a universal set of fault-tolerant quantum gates can be implemented using the state in Eq. (3).

3 Concatenation schemes

In this section, we describe the concatenation schemes in detail. To explain our principle for choosing codes, we first need to explain our approach to decoding. For fault-tolerant error correction, it suffices to perform logical X-basis measurements for each code block, i.e. the *inner* codes, feed the measurement outcomes to higher-level decoding algorithms, such as the minimum-weight perfect matching (MWPM) decoder [21, 42] or the union-find decoder [20], and apply the appropriate Pauli correction. We shall refer to these algorithms as *outer* decoding algorithms. The standard inputs to these decoding algorithms are the X-basis measurement results of the individual physical qubits. Applied to our setup, one way to decode our scheme would be to perform a destructive logical X-basis measurement for every vertex and feed the logical measurement outcome to an outer decoding algorithm.

However, at least when it comes to the threshold, we consider it unlikely for this approach to be able to outperform the body-centered cubic (bcc) cluster state (without concatenation). The main reason is that, to the best of our knowledge, there is no quantum error correcting code whose pseudothreshold[1] is above the fault-tolerant quantum error correction

[1] A pseudothreshold of a code is defined as the error rate below which the logical error rate is below the physical error rate.

threshold of the 3D cluster state. If the pseudothreshold is lower than the threshold of the bcc cluster state, the overall threshold of the scheme will be limited by it. This argument suggests that, unless more information is passed from the lower level to the outer decoder [41], improvements in threshold are likely unachievable.

This is why we seek to consider a different decoding algorithm in which information from the code $\mathcal{C}$ is passed to the outer decoding algorithm. To that end, we remark on a few facts that are relevant to our choice of inner codes. First of all, both MWPM and union-find decoders can also easily accommodate erasure errors. In particular, these decoders boast a much higher threshold for erasure error than for Pauli noise. Secondly, for the bcc cluster state with the phenomenological noise model, i.e., a model in which a noiseless cluster state is subjected to an identical independent noise model, the threshold for the depolarizing noise is $\sim 3\%$ whereas the threshold against loss is 25% [3]. Thirdly, when an error is detected at a low level, instead of correcting the error, one can instead opt to artificially "lose" that logical qubit by erasing it. This information can be fed into the outer decoder as an erasure error. This Pauli-to-loss conversion of the error model is the main idea behind our decoding scheme.

To make this idea work, we must choose the code $\mathcal{C}$ such that there is a circuit that prepares the state $|\psi_{G_{\mathrm{bcc}},\mathcal{C}}\rangle$ in such a way that any single circuit-level error propagates to an error convertible to a (possibly correlated) erasure error upon applying the destructive logical X-basis measurement of $\mathcal{C}$. If $\mathcal{C}$ is a Calderbank-Shor-Steane (CSS) code, the X-basis measurement is unaffected by any X-error. Thus, in order to make a circuit-level error detectable from the logical X measurement of $\mathcal{C}$, it suffices for the propagated error to be equivalent up to stabilizers and Pauli-X operators to a tensor product of Paulis over different code blocks, such that the individual errors within each code block are detectable. More formally, we propose the following definition.

Let us first define the *circuit-level error*. A circuit-level error is an error occurring after individual gates, specifically, on a set of qubits the gate acts on. For instance, if we have a noisy CZ gate, its circuit-level noise will be one or two-qubit Paulis occurring at the qubits that the CZ gate acts on. We now define what it means for these errors to be $\mathcal{C}$-detectable.

Definition 3.1. Let $\mathcal{C}$ be a CSS code. Consider a circuit that prepares $|\psi_{G,\mathcal{C}}\rangle$. A circuit-level noise operator is $\mathcal{C}$-detectable if the noise, up to some stabilizer and Pauli-X operators S, is of the following form:

$$S\prod_{v\in G} P_v, \tag{4}$$

where P_v is an error detectable with respect to $\mathcal{C}$.

Definition 3.2. Let $\mathcal{C}$ be a CSS code. Consider a circuit that prepares $|\psi_{G,\mathcal{C}}\rangle$. This circuit is $\mathcal{C}$-detectable if every circuit-level Pauli noise operator is $\mathcal{C}$-detectable.

There's a subtle difference between $\mathcal{C}$-detectability and conventional fault-tolerant circuits. For a fault-tolerant gadget [2], the output must contain at most one fault, whereas $\mathcal{C}$-detectability only requires the number of faults to be within the detection ability of the inner code up to stabilizers and X operators. If we can create a $\mathcal{C}$-detectable state preparation circuit, we can convert circuit-level errors to erasure errors of the outer code.

The main question we address in this Section is whether there are circuits preparing the state $|\psi_{\mathrm{bcc},\mathcal{C}}\rangle$ in a $\mathcal{C}$-detectable way. We discuss two families of such codes. The first family consists of codes that are equipped with a fault-tolerant logical $|+\rangle$-state preparation and logical (transversal) CZ gate. Such codes, by using these operations, come with a natural $\mathcal{C}$-detectable state preparation circuit. We refer to the inner codes used in these constructions as *Type I* codes.

Surprisingly, we find that there are choices of inner codes without any fault-tolerant logical $\overline{CZ}$ gates, which nevertheless admit a $\mathcal{C}$-detectable state preparation circuits for $|\psi_{\mathrm{bcc},\mathcal{C}}\rangle$. We refer to these inner codes as *Type II* codes.

3.1 Type I

We consider two examples of Type I codes: a $[\![4,1,1,2]\!]$ subsystem code [35] [2] and the Steane code [55]. The stabilizers of the $[\![4,1,1,2]\!]$ are given by X_1X_2, Z_1Z_3, $XXXX$ and $ZZZZ$, and the logical operations are given by

$$\begin{aligned}\overline{X} &= X_1X_3\\ \overline{Z} &= Z_1Z_2.\end{aligned} \tag{5}$$

Transversal logical $\overline{CZ}$ gate construction is trivially possible. Moreover, $[\![4,1,1,2]\!]$ concatenation with the surface code, which is unitary-equivalent to a 4.8.8 color code, has been studied previously [16].

Another code we consider is the $[\![7,1,3]\!]$ Steane code, whose stabilizers are $X_1X_2X_3X_4$, $X_1X_3X_5X_7$, $X_3X_4X_6X_7$, $Z_1Z_2Z_3Z_4$, $Z_1Z_3Z_5Z_7$, and $Z_3Z_4Z_6Z_7$. This is also the smallest 2D color code [4].

Since both codes admit transversal CZ gates, Z errors are local throughout the whole circuit. On the other hand, although a CZ gate will propagate X errors to Z errors, transversality guarantees at most one error will present in a given code block, which could be effectively corrected or detected by the inner code.

[2] In this case, the $[\![4,1,1,2]\!]$ code is derived from a $[\![4,2,2]\!]$ code, the $n=2$ Bacon-Shor code, by employing just one of its logical qubits.

3.2 Type II

In this section, we introduce Type II codes using up to 4 qubits. We first introduce the $[\![2,1,1]\!]$ repetition code. The stabilizer group of the $[\![2,1,1]\!]$ code is $\langle\{X_1X_2\}\rangle$ and the logical operators are

$$\begin{aligned}\overline{Z} &= Z_1Z_2\\ \overline{X} &= X_1.\end{aligned} \tag{6}$$

The logical $\overline{CZ}$ gate can be applied by applying a physical CZ gate to every pair of connected qubits in Figure 2.

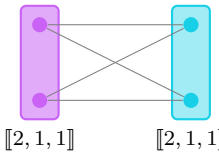


Figure 2: Logical $\overline{CZ}$ gate of the $[\![2,1,1]\!]$ code. Here, two grouped dots represent the two qubits in each $[\![2,1,1]\!]$ code.

A state preparation for the logical $|\overline{+}\rangle$ state entails preparing the $|++\rangle$ state. Thus, no extra entangling gate is needed in the encoding procedure. The concatenated geometric structure, as shown in Figure 3, is referred to as double-edge cubic lattice in Ref. [37] and is equivalent to the crazy-graph construction with $n = 2$ of the bcc lattice [48].

For $[\![3,1,1]\!]$ codes, two different constructions exist. The $[\![3,1,1]\!]_1$ code is the three-qubit repetition code. The $\overline{CZ}$ gate can be performed with nine applications of physical CZ gates [57], as shown in Figure 4a.

For larger repetition codes, we could in principle perform a $\overline{CZ}$ gate by applying $O(n^2)$ CZ gates. However, we expect the threshold to decrease as n increases. Our numerical simulation shows that, indeed, $[\![3,1,1]\!]_1$ code performs worse than the $[\![2,1,1]\!]$ code.

The $[\![3,1,1]\!]_2$ code can be viewed as an "interpolation" between the $[\![2,1,1]\!]$ code and the $[\![4,1,1,2]\!]$ code, which is manifested by its logical operators defined by $\overline{Z} = Z_1Z_3$ and $\overline{X} = X_1$. The logical $\overline{CZ}$ gate of this code is shown in Figure 4b.

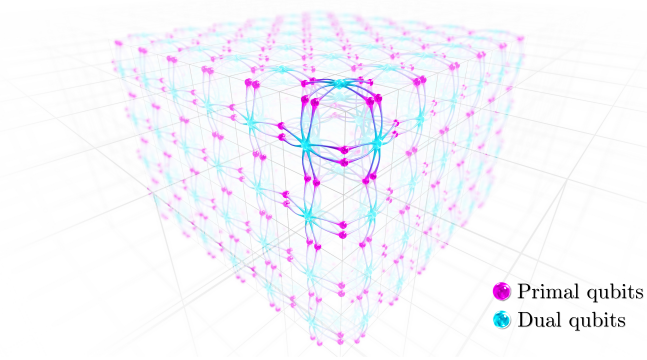


Figure 3: Illustration of the cellular structure of the concatenated $[\![2,1,1]\!]$ lattice.

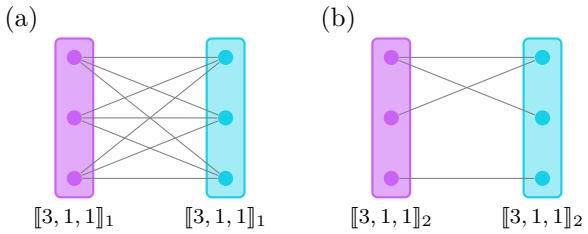


Figure 4: (a) Logical $\overline{CZ}$ gates for the $[\![3,1,1]\!]_1$ code, (b) logical $\overline{CZ}$ gates for the $[\![3,1,1]\!]_2$ code

.

Because none of the codes mentioned in this section can detect an arbitrary single-qubit error, it is a priori not obvious why there should be a $\mathcal{C}$-detectable state preparation procedure. We claim below that this is nevertheless the case, by showing explicit examples. Consider the gate sequence in Figure 5. In the ensuing discussion we shall refer to the individual code blocks as central (C), northern (N), southern (S), western (W), and eastern (E), depending on their relative location with respect to the central block. Since every code block can be interpreted as a central block and the scheduling in Figure 5 can be applied in a uniform way throughout the entire lattice, it suffices to consider single-qubit errors occurring on the central block and two-qubit errors that can occur after each CZ gate.

Let us first consider single-qubit errors. Since CZ gates commute with Z errors, Z errors do not propagate and remain $\mathcal{C}$-detectable. For a single-qubit X error occurring in the central block, we make the following observation: if such an error occurs after Step 4, it propagates to a tensor product of single-qubit Z errors and identity on the N, S, E, and W code blocks. Therefore, in this case, the error propagates to a $\mathcal{C}$-detectable error. On the other hand, if an error occurs between Step 1 and Step 4, it propagates to a tensor product of single or two-qubit Z errors within each block. Although a two-qubit error within a block is a logical Z error of that code, the propagated error is still $\mathcal{C}$-detectable because the resulting error is equivalent, up to an outer code stabilizer, to a tensor product of single-qubit Z errors and identity on each code block. Specifically, the stabilizer is given by $\overline{X}_C\overline{Z}_N\overline{Z}_S\overline{Z}_W\overline{Z}_E$.

Moreover, all the two-qubit circuit-level errors propagate to $[\![2,1,1]\!]$-detectable errors. We have investigated how these errors propagate, up to stabilizers, the result of which is summarized in Appendix A. We also note that the schemes based on the $[\![3,1,1]\!]_1$ and the $[\![3,1,1]\!]_2$ code are $\mathcal{C}$-detectable; see Appendix A.

We emphasize that the ordering of the gates in Figure 5 is important. This gate sequence is in particular

different from the "natural" gate sequence proposed in Ref. [57] that applies all physical CZ gates of a logical $\overline{CZ}$ gate consecutively (shown in Figure 6). As one can see in Figure 6, there are circuit-level errors which propagate to errors that are *not* $[\![2,1,1]\!]$-detectable. Thus, the fact that all circuit-level errors are $[\![2,1,1]\!]$-detectable is a special property of the circuit shown in Figure 5. The inner codes and their respective stabilizers and logical operators are summarized in Table 1.

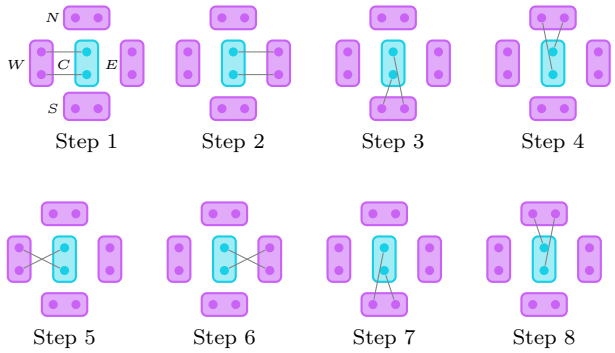


Figure 5: Gate sequence of the $[\![2,1,1]\!]$ scheme.

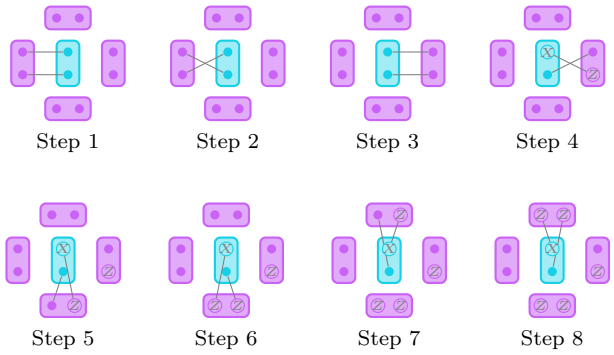


Figure 6: "Natural" gate sequence of the $[\![2,1,1]\!]$ scheme that fails to be $\mathcal{C}$-detectable. Note that a single-qubit X error happening on one of the central qubits could propagate to two undetectable errors in the N and S blocks.

4 Thresholds

In this Section, we discuss the performance of our schemes against the phenomenological noise model and the circuit-level noise model in terms of the threshold error rate. After measuring all the qubits in the X-basis, the decoder works in two steps. First, if an error is detected in any individual code block, that code block is erased. If not, the measured logical information of that code block is passed to the higher level decoder, which we choose to be the MWPM decoder with Manhattan norm. We use the Blossom V algorithm [32] with 10^6 trials to get each data point. We perform the simulation on lattices of varying size L, ranging from 3 to 19.

Inner codes	Cubic	$[\![4,1,1,2]\!]$	$[\![7,1,3]\!]$
Stabilizers		$XXXX$ $ZZZZ$	$g_1,\cdots,g_6$
$\overline{X}$ $\overline{Z}$	X Z	$XXII$ $ZIZI$	$X_1\cdots X_7$ $Z_1\cdots Z_7$

Inner codes	$[\![2,1,1]\!]$	$[\![3,1,1]\!]_1$	$[\![3,1,1]\!]_2$
Stabilizers	XX	XXI IXX	XXX IZZ
$\overline{X}$ $\overline{Z}$	XI ZZ	XII ZZZ	XII ZIZ

Table 1: Definitions of the inner codes in the stabilizer formalism and their respective logical operators.

4.1 Phenomenological noise model

The phenomenological noise model is a model in which the noiseless cluster state $|\psi_{G,\mathcal{C}}\rangle$ is subjected to a depolarizing noise channel $\mathcal{D}_p$:

$$\mathcal{D}_p(\rho) = (1-\frac{3}{2}p)\rho + \frac{p}{2}(X\rho X + Y\rho Y + Z\rho Z) \quad (7)$$

on each qubit[3]. We expect the logical error rate p_L to obey the following scaling relation near the threshold:

$$p_L = f(a + b(p-p_{th})d^{\frac{1}{\nu}}). \quad (8)$$

Fitting to the quadratic scaling ansatz $\sum_{i=0}^{3} A_i(p_i - p_{th})d^{\frac{1}{\nu}}$ [59] with parameters A_i, p_{th}, and ν, we obtain a threshold of 8.034(3)%.

Intuitively, we can understand the value of this threshold in terms of an effective error model on the bcc cluster state. For instance, for the $[\![2,1,1]\!]$ scheme, upon converting the error detected from the code $\mathcal{C}$ to an erasure error, we obtain an effective model in which the rates for Pauli (p_P) and erasure (p_E) errors are

$$\begin{aligned} p_P &= \frac{p^2}{1-2p(1-p)}, \\ p_E &= 2p(1-p). \end{aligned} \quad (9)$$

The model effectively defines a path along the phase diagram of the bcc cluster state as shown in Figure 7. We note that the values of the effective p_P

[3]We choose the normalization such that the value of p corresponds to the probability of an X error occurring. It's important to note that for our purposes, we treat a Y error as being equivalent to an X error followed by a Z error.

and p_E obtained at our threshold are consistent with the thresholds for the respective parameters obtained in Ref. [3]. The conversion of the Pauli errors into loss errors generally leads to improvements in thresholds, as summarized in Table 2.

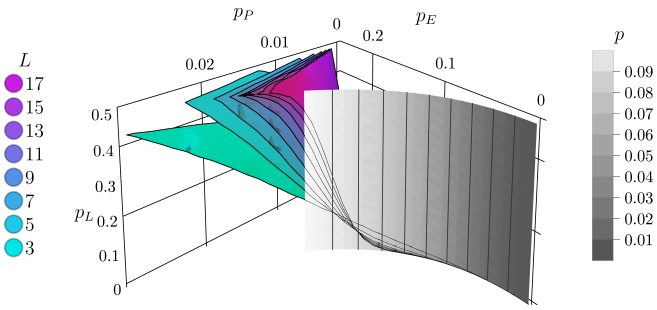


Figure 7: Phase diagram for the bcc scheme [42, 44, 45], here p_E and p_P are the loss and Pauli error rates, respectively, and p_L is the logical error rate. The grayscale surface is the section of effective loss-error relationship for the $[\![2,1,1]\!]$ scheme, from which we can directly read off the error threshold of approximately 8.03%.

Inner codes	Cubic	$[\![4,1,1,2]\!]$	$[\![7,1,3]\!]$
p_{th}	2.936(2)%[a]	4.195(5)%	4.137(5)%
Inner codes	$[\![2,1,1]\!]$	$[\![3,1,1]\!]_1$	$[\![3,1,1]\!]_2$
p_{th}	8.034(3)%	10.26(1)%	5.66(1)%

[a]Our data agrees with the reported results from [59]

Table 2: Thresholds of the concatenation schemes against phenomenological noise.

4.2 Circuit-level noise model

A more realistic error model should take into account errors in individual operations, which we analyze as follows. We assume that the logical $|\overline{+}\rangle$ state of the code $\mathcal{C}$ is prepared perfectly, followed by independent depolarizing noise on every physical qubit. After that, every gate is followed by depolarizing noise, i.e. each two-qubit error can occur with probability $p/15$, and measurements are preceded by depolarizing noise as well. Finally, we also assume that idle qubits experience depolarizing noise.[4]

One may wonder about the motivation behind our non-standard choice of modeling the logical $|\overline{+}\rangle$ state preparation as a perfect preparation followed by depolarizing noise. This is because there are known methods to boost the fidelity of such states using methods such as post-selection [15, 46, 50]. As such, the result of these simulations can be thought of as the best-case-scenario threshold one can hope to get, after incorporating such techniques.

We also note that such an error model reduces to the standard circuit-level depolarizing noise model if the logical $|\overline{+}\rangle$ state is a product state. In our case, there are two examples: the $[\![2,1,1]\!]$ and the $[\![3,1,1]\!]_1$ code. Therefore, for these codes, one can interpret the result as the true threshold against the standard circuit-level noise model.

After the state is prepared, all the qubits are measured in the X-basis. Applying the decoding algorithm in Section 3, we estimate the logical error rate; see Table 3 for the summary. We note that the thresholds are generally better than the bcc cluster state, except for the $[\![3,1,1]\!]_1$-code-based scheme.

Taking into account our non-standard logical $|\overline{+}\rangle$ state preparation error modeling, we can interpret these results as follows. First of all, because our model reduces to the standard circuit-level noise model for the $[\![2,1,1]\!]$ and the $[\![3,1,1]\!]_1$ code, these results can be compared fairly against the other schemes in the literature. We note that the $[\![2,1,1]\!]$ scheme improves upon the threshold of the bcc scheme, whereas the $[\![3,1,1]\!]_1$ scheme does not. Since repetition codes involving a larger number of qubits will require even more gates, we can therefore expect that $[\![n,1,1]\!]$ code-based constructions for $n > 3$ will likely yield worse results.

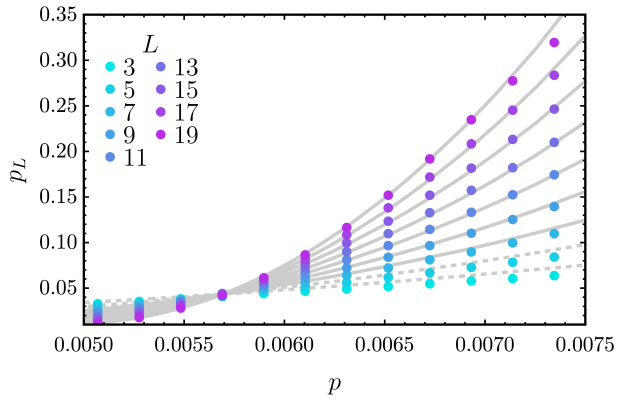
Secondly, for the other schemes, e.g., $[\![3,1,1]\!]_2$, $[\![4,1,1,2]\!]$, and $[\![7,1,3]\!]$, we see improvements in the threshold. Therefore, if we can utilize the high-fidelity logical state preparation schemes, e.g., Ref. [47, 49], we can expect to again have improvements against the bcc scheme. However, we note that a more naive state preparation circuit does not yield improvements. For instance, we studied the state preparation circuit for the $[\![4,1,1,2]\!]$ in which one prepares the logical $|\overline{+}\rangle|\overline{0}\rangle$ state, which can be prepared fault-tolerantly by preparing two Bell pairs. Even for such a simple circuit, we observed a lower threshold than the bcc cluster state, which is shown in Figure 8.

4.3 Biased noise model

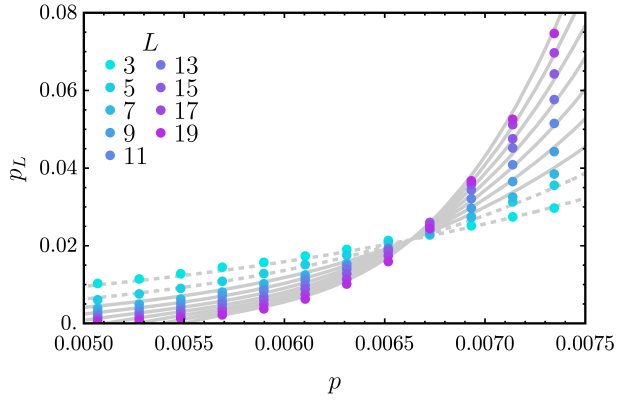
In this section, we study our schemes against the biased noise. We remark that the following analysis is not based on an explicit Monte Carlo simulation. Rather, it is based on an exact mapping of the biased noise model to the phenomenological noise model by exploiting the fact that Z errors do not propagate through the circuit.

In this model, the two-qubit noise in Section 4.2 is replaced with an error model completely biased to Z-

[4]We note in passing that, unlike the other codes, the $[\![3,1,1]\!]_2$ code requires one more idle timestep for two of the qubits, in order to have consistent global scheduling of the gates. We have ignored the extra error incurred in this step. However, the conclusion of our work does not depend significantly on this choice.



(a) Error threshold of the bcc scheme



(b) Error threshold of the $[\![2,1,1]\!]$ scheme

Figure 8: Comparison of error threshold under circuit-level noise. Note that the dashed lines in the figure represent data that suffer from finite size effects and are therefore excluded from the fitting process.

error. For instance, for the two-qubit gates, we apply the following noise model.

$$\mathcal{D}_p^Z(\rho) = (1-p)\rho + \frac{p}{3}(ZZ\rho ZZ + ZI\rho ZI + IZ\rho IZ) \tag{10}$$

This error model is equivalent to a limiting case of the error model studied in Ref. [57], with the bias parameter β taken to be infinity, i.e. we essentially suppress the propagation of gate level noise. Subsequently, the model reduces to an independent error model whose resulting threshold error rate can be calculated by a counting argument. For instance, the $[\![2,1,1]\!]$ scheme results in an effective noise parameter of $p = 4\cdot 2\cdot p_{CZ} + p_M + p_I = 4\cdot 2\cdot 2p/3 + 4p/3 = 20p/3$, leading to a p_{th} of 1.205(1)% [5]. Unsurprisingly, the scheme is extremely tolerant to biased noise, as indicated by the effective error rate listed in Table 4.

[5] Our results point out a possible systematic inconsistency in Ref. [57], as the presented numerical results exceed the worst-case scenario delineated by our semi-analytic approach.

Inner codes	Cubic	$[\![4,1,1,2]\!]$	$[\![7,1,3]\!]$
p_{th}	0.5692(1)%	0.701(1)%	0.678(1)%
Inner codes	$[\![2,1,1]\!]$	$[\![3,1,1]\!]_1$	$[\![3,1,1]\!]_2$
p_{th}	0.664(3)%	0.3216(1)%	0.6947(2)%

Table 3: Threshold of the concatenation schemes against circuit-level noise model, assuming that the initial encoded logical state of the code is prepared perfectly, followed by a depolarizing noise model on all the qubits.

Inner codes	Cubic	$[\![4,1,1,2]\!]$	$[\![7,1,3]\!]$
Biased	0.734(5)%	1.05(1)%	1.03(2)%
Inner codes	$[\![2,1,1]\!]$	$[\![3,1,1]\!]_1$	$[\![3,1,1]\!]_2$
Biased	1.205(1)%	1.090(5)%	1.18(5)%

Table 4: Thresholds of the concatenation schemes against biased noise model.

5 Overhead

We now study the overhead of our schemes assuming a range of aspirational but reasonable circuit-level noise rates. Overall, we note that the only scheme that leads to a lower overhead is based on the $[\![2,1,1]\!]$ code. As such, we primarily focus on this code.

A particularly interesting regime is an error rate approaching 10^{-3}. This error rate is slightly lower than what can currently be achieved experimentally. As such, studies of overhead in this parameter range can provide an accurate estimate of the overhead required in realistic quantum computers.

Normally, the logical error rate decays exponentially with the code distance L. However, our numerical study shows that there are effects that make the fitting to an exponentially decaying function unreliable, which may be a result of the interplay between the decoder and the inner code [22]. Thus, we first focus on demonstrating improvements in overheads in the high-error-rate regime, based on our Monte Carlo simulation. We then explain, based on a combinatorial argument, why we anticipate this improvement to persist in a lower error rate regime as well.

Due to the aforementioned effects, we cannot reliably extrapolate the suppression of p_L, and the overhead advantage may not exist for large L. As shown in Figure 9, the error suppresion relation for the $[\![2,1,1]\!]$ scheme are convex in the log-log plot and eventually crosses that of the bcc scheme. However, we can in-

stead interpolate the data to fairly compare the overhead of our $[\![2,1,1]\!]$ scheme to the bcc cluster state when L is not that large. We fit the logical error rate at every physical error rate with the function $e^{a+bL+cL^3}$ to account for those effects [6]. From this, we can infer overhead for the overall spacetime volume scaling as sL^2, where s is the block size of the inner code. The $[\![2,1,1]\!]$ scheme outperforms the bcc scheme in this regime, as shown in Figure 9.

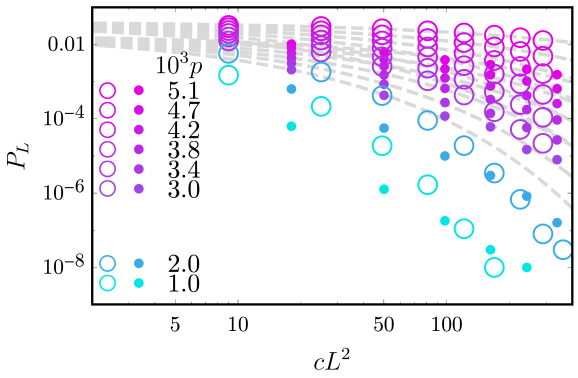


Figure 9: Relationship of p_L v.s. L at different p values. Here the dots represent data points of the $[\![2,1,1]\!]$ scheme and the circles represent that of the bcc scheme.

For smaller and more practical p_{th} values, numerical simulation fails to give accurate results due to the diverging time to convergence. However, the following analytical argument, based on the leading-order approximation, suggests that even in this regime we expect to have lower overhead.

For the bcc lattice with periodic boundary conditions in two directions and a smooth boundary on the third, p_L for a single logical X operator, to the leading order, scales approximately as [62]:

$$L^2 \binom{L}{\lfloor \frac{L}{2} \rfloor} p^{\lfloor L/2 \rfloor} \sim \frac{L^{3/2}}{\sqrt{2\pi}} (p/4)^{\lfloor L/2 \rfloor}, \tag{11}$$

where L^2 counts the number of lowest weight logical operators that percolate to the two rough boundaries. Thus $p_L = o(p^{L/2})$ as $p \to 0$ and $L \to \infty$.

For the $[\![2,1,1]\!]$ scheme with the same boundary condition, a length L path could fit in $N \in \{1, \cdots, L\}$ erasure errors, and the effective code distance is reduced to $L - N$. Using the effective p_P and p_L relations outlined in Subsection 4.1, p_L is given by

$$L^2 \sum_{N=0}^{L} \binom{L}{N} \binom{L-N}{\lfloor \frac{L-N}{2} \rfloor} 2^N p^{N+2\lfloor \frac{L-N}{2} \rfloor}, \tag{12}$$

[6]The choice of the qubic non-linear term is to accommodate to the numerical methods.

which scales as $o(p^L)$ as $p \to 0$ and $L \to \infty$. Moreover, this quantity is upper-bounded by

$$L^3 \binom{L}{\frac{L}{2}}^2 p^L \sim \frac{L^2}{2\pi} (p/4)^L \tag{13}$$

as $L \to \infty$, which is, in turn, upper-bounded by Eq. 11.

Since the $[\![2,1,1]\!]$ scheme only contributes to a constant factor in the overhead, in the low-noise regime we expect to have better performance in terms of resource usage. For the other concatenated schemes, due to $\mathcal{C}$-detectability, the logical Pauli error rate p scales as $o(p^i)$ with some $i \geq 2$. Therefore, we expect similar results to hold. At around $p_L = 10^{-6}$ and $p = 10^{-3}$, the overhead value is listed in Table 5.

Inner codes	Cubic	$[\![4,1,1,2]\!]$	$[\![7,1,3]\!]$
L_c/L indep.	1.00	1.27	2.40
Inner codes	$[\![2,1,1]\!]$	$[\![3,1,1]\!]_1$	$[\![3,1,1]\!]_2$
L_c/L indep.	0.68	4.74	1.43

Table 5: Ratio of overhead between concatenation schemes L_c and the bcc scheme L.

6 Conclusions

We have identified a fault-tolerant quantum error correction scheme that effectively converts circuit-level Pauli errors to erasure errors. Besides the much higher phenomenological error threshold compared with regular foliated topological quantum codes [5], our scheme also demonstrates improvements in circuit-level error thresholds, indicating a path to practical performance advantage. We have restricted our discussion to decoding by the MWPM algorithm with Manhattan norm weightings. Using better decoding schemes, employing lattice correlations, or utilizing predecoders will likely lead to even more improvements [9, 14, 18, 19, 52, 60]. Furthermore, different outer codes, such as those with $> 50\%$ erasure thresholds [37], could be used in similar constructions[7]. Lastly, we leave it as an open problem

[7]Even a trivial cluster state consisting of n EPR pairs has a threshold of 100%. This does not violate the no-cloning theorem since no logical state is encoded in the lattice, but only a logical EPR pair. In fact, the 2-way quantum capacity of a binary erasure channel is exactly 1.

to convert our scheme to a two-dimensional architecture and study its performance against the standard surface code.

7 Acknowledgement

We would like to express our gratitude to Kianna Wan, Noah Shutty, Scott Xu, Ani Krishna, and Michael Vasmer for their valuable insights and helpful discussions. We also thank the anonymous referees for their detailed feedback and constructive suggestions that have greatly improved the quality of this manuscript. P.H. acknowledges the support of ARO (award W911NF2120214), CIFAR, and the Simons Foundation.

References

[1] Dorit Aharonov and Michael Ben-Or. Fault-Tolerant Quantum Computation with Constant Error Rate. *SIAM Journal on Computing*, 38(4): 1207–1282, July 2008. ISSN 0097-5397. DOI: https://doi.org/10.1137/S0097539799359385.

[2] Panos Aliferis, Daniel Gottesman, and John Preskill. Quantum accuracy threshold for concatenated distance-3 codes. *Quantum Info. Comput.*, 6(2):97–165, mar 2006. ISSN 1533-7146. DOI: https://doi.org/10.26421/QIC6.2-1.

[3] Sean D Barrett and Thomas M Stace. Fault tolerant quantum computation with very high threshold for loss errors. *Physical review letters*, 105(20):200502, 2010. DOI: https://doi.org/10.1103/PhysRevLett.105.200502.

[4] Alejandro Bermudez, Xiaosi Xu, Ramil Nigmatullin, Joe O'Gorman, Vlad Negnevitsky, Philipp Schindler, Thomas Monz, UG Poschinger, Cornelius Hempel, Jonathan Home, et al. Assessing the progress of trapped-ion processors towards fault-tolerant quantum computation. *Physical Review X*, 7(4):041061, 2017. DOI: https://doi.org/10.1103/PhysRevX.7.041061.

[5] A Bolt, G Duclos-Cianci, D Poulin, and TM Stace. Foliated quantum error-correcting codes. *Physical review letters*, 117(7):070501, 2016. DOI: https://doi.org/10.1103/PhysRevLett.117.070501.

[6] J Eli Bourassa, Rafael N Alexander, Michael Vasmer, Ashlesha Patil, Ilan Tzitrin, Takaya Matsuura, Daiqin Su, Ben Q Baragiola, Saikat Guha, Guillaume Dauphinais, et al. Blueprint for a scalable photonic fault-tolerant quantum computer. *Quantum*, 5:392, 2021. DOI: https://doi.org/10.22331/q-2021-02-04-392.

[7] Sergey Bravyi and Alexei Kitaev. Universal quantum computation with ideal clifford gates and noisy ancillas. *Phys. Rev. A*, 71:022316, Feb 2005. DOI: https://doi.org/10.1103/PhysRevA.71.022316. URL https://link.aps.org/doi/10.1103/PhysRevA.71.022316.

[8] Sergey Bravyi, David Gosset, Robert Koenig, and Marco Tomamichel. Quantum advantage with noisy shallow circuits. *Nature Physics*, 16(10):1040–1045, 2020. DOI: https://doi.org/10.1038/s41567-020-0948-z.

[9] Nikolas P Breuckmann and Xiaotong Ni. Scalable neural network decoders for higher dimensional quantum codes. *Quantum*, 2:68, 2018. DOI: https://doi.org/10.22331/q-2018-05-24-68.

[10] Hans J Briegel, David E Browne, Wolfgang Dür, Robert Raussendorf, and Maarten Van den Nest. Measurement-based quantum computation. *Nature Physics*, 5(1):19–26, 2009. DOI: https://doi.org/10.1038/nphys1157.

[11] Benjamin J. Brown and Sam Roberts. Universal fault-tolerant measurement-based quantum computation. *Physical Review Research*, 2(3), aug 2020. DOI: 10.1103/physrevresearch.2.033305. URL https://doi.org/10.1103%2Fphysrevresearch.2.033305.

[12] Yudong Cao, Jonathan Romero, Jonathan P Olson, Matthias Degroote, Peter D Johnson, Mária Kieferová, Ian D Kivlichan, Tim Menke, Borja Peropadre, Nicolas PD Sawaya, et al. Quantum chemistry in the age of quantum computing. *Chemical reviews*, 119(19):10856–10915, 2019. DOI: https://doi.org/10.1021/acs.chemrev.8b00803.

[13] Christopher Chamberland, Tomas Jochym-O'Connor, and Raymond Laflamme. Thresholds for universal concatenated quantum codes. *Physical review letters*, 117(1):010501, 2016. DOI: https://doi.org/10.1103/PhysRevLett.117.010501.

[14] Christopher Chamberland, Luis Goncalves, Prasahnt Sivarajah, Eric Peterson, and Sebastian Grimberg. Techniques for combining fast local decoders with global decoders under circuit-level noise. *Quantum Science and Technology*, 8(4):045011, jul 2023. DOI: https://doi.org/10.1088/2058-9565/ace64d.

[15] Rui Chao and Ben W Reichardt. Quantum error correction with only two extra qubits. *Physical review letters*, 121(5):050502, 2018. DOI: https://doi.org/10.1103/PhysRevLett.121.050502.

[16] Ben Criger and Barbara Terhal. Noise thresholds for the [4,2,2]-concatenated toric code. *Quantum Info. Comput.*, 16(15–16): 1261–1281, nov 2016. ISSN 1533-7146. DOI: https://doi.org/10.26421/QIC16.15-16-1.

[17] Andrew W Cross, David P DiVincenzo, and Barbara M Terhal. A comparative code study for quantum fault-tolerance. *arXiv preprint arXiv:0711.1556*, 2007. DOI: http://dx.doi.org/10.26421/QIC9.7-8-1.

[18] Poulami Das, Christopher A Pattison, Srilatha

Manne, Douglas M Carmean, Krysta M Svore, Moinuddin Qureshi, and Nicolas Delfosse. Afs: Accurate, fast, and scalable error-decoding for fault-tolerant quantum computers. In *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 259–273. IEEE, 2022. DOI: https://doi.org/10.1109/HPCA53966.2022.00027.

[19] Nicolas Delfosse. Hierarchical decoding to reduce hardware requirements for quantum computing. *arXiv preprint arXiv:2001.11427*, 2020. DOI: https://doi.org/10.48550/arXiv.2001.11427.

[20] Nicolas Delfosse and Naomi H Nickerson. Almost-linear time decoding algorithm for topological codes. *Quantum*, 5:595, December 2021. ISSN 2521-327X. DOI: https://doi.org/10.22331/q-2021-12-02-595.

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

[22] Austin G Fowler, Adam C Whiteside, Angus L McInnes, and Alimohammad Rabbani. Topological code autotune. *Physical Review X*, 2(4):041003, 2012. DOI: https://doi.org/10.1103/PhysRevX.2.041003.

[23] Kosuke Fukui, Akihisa Tomita, Atsushi Okamoto, and Keisuke Fujii. High-threshold fault-tolerant quantum computation with analog quantum error correction. *Physical review X*, 8(2):021054, 2018. DOI: https://doi.org/10.1103/PhysRevX.8.021054.

[24] Iulia M Georgescu, Sahel Ashhab, and Franco Nori. Quantum simulation. *Reviews of Modern Physics*, 86(1):153, 2014. DOI: https://doi.org/10.1103/RevModPhys.86.153.

[25] Daniel Gottesman. Stabilizer Codes and Quantum Error Correction, May 1997.

[26] Daniel Gottesman. Theory of fault-tolerant quantum computation. *Physical Review A*, 57(1):127, 1998. DOI: https://doi.org/10.1103/PhysRevA.57.127.

[27] Daniel Gottesman. Fault-tolerant quantum computation with constant overhead. *Quantum Information & Computation*, 14(15-16):1338–1372, November 2014. ISSN 1533-7146. DOI: https://doi.org/10.1016/S0003-4916(02)00018-0.

[28] Alexei Kitaev. Fault-tolerant quantum computation by anyons. *Annals of Physics*, 303(1):2–30, 2003. ISSN 0003-4916. DOI: https://doi.org/10.1016/S0003-4916(02)00018-0. URL https://www.sciencedirect.com/science/article/pii/S0003491602000180.

[29] Emanuel Knill. Fault-tolerant postselected quantum computation: Schemes. *arXiv preprint quant-ph/0402171*, 2004. DOI: https://doi.org/10.48550/arXiv.quant-ph/0402171.

[30] Emanuel Knill. Fault-tolerant postselected quantum computation: Threshold analysis. *arXiv preprint quant-ph/0404104*, 2004. DOI: https://doi.org/10.48550/arXiv.quant-ph/0404104.

[31] Emanuel Knill. Quantum computing with realistically noisy devices. *Nature*, 434(7029):39–44, 2005. DOI: https://doi.org/10.1038/nature03350.

[32] Vladimir Kolmogorov. Blossom v: a new implementation of a minimum cost perfect matching algorithm. *Mathematical Programming Computation*, 1(1):43–67, 2009. DOI: https://doi.org/10.1007/s12532-009-0002-8.

[33] Aleksander Kubica, Arbel Haim, Yotam Vaknin, Fernando Brandão, and Alex Retzker. Erasure qubits: Overcoming the t_1 limit in superconducting circuits. *arXiv preprint arXiv:2208.05461*, 2022. DOI: https://doi.org/10.48550/arXiv.2208.05461.

[34] Daniel Litinski. Magic State Distillation: Not as Costly as You Think. *Quantum*, 3:205, December 2019. ISSN 2521-327X. DOI: 10.22331/q-2019-12-02-205. URL https://doi.org/10.22331/q-2019-12-02-205.

[35] John Napp and John Preskill. Optimal bacon-shor codes. *arXiv preprint arXiv:1209.0794*, 2012. DOI: https://doi.org/10.48550/arXiv.1209.0794.

[36] Michael Newman, Leonardo A de Castro, and Kenneth R Brown. Generating fault-tolerant cluster states from crystal structures. *Quantum*, 4:295, 2020. DOI: https://doi.org/10.22331/q-2020-07-13-295.

[37] Naomi Nickerson and Héctor Bombín. Measurement based fault tolerance beyond foliation. *arXiv preprint arXiv:1810.09621*, 2018. DOI: https://doi.org/10.48550/arXiv.1810.09621.

[38] Naomi H Nickerson, Ying Li, and Simon C Benjamin. Topological quantum computing with a very noisy network and local error rates approaching one percent. *Nature communications*, 4(1):1–5, 2013. DOI: https://doi.org/10.1038/ncomms2773.

[39] Kyungjoo Noh and Christopher Chamberland. Fault-tolerant bosonic quantum error correction with the surface–gottesman-kitaev-preskill code. *Physical Review A*, 101(1):012316, 2020. DOI: https://doi.org/10.1103/PhysRevA.101.012316.

[40] Kyungjoo Noh, Christopher Chamberland, and Fernando GSL Brandão. Low-overhead fault-tolerant quantum error correction with the surface-gkp code. *PRX Quantum*, 3(1):010315, 2022. DOI: https://doi.org/10.1103/PRXQuantum.3.010315.

[41] David Poulin. Optimal and efficient decoding of concatenated quantum block codes. *Physical Review A*, 74(5), nov 2006. DOI: https://doi.org/10.1103/PhysRevA.74.052333. URL https://doi.org/10.1103%2Fphysreva.74.052333.

[42] Robert Raussendorf and Jim Harrington. Fault-tolerant quantum computation with high threshold in two dimensions. *Physical review letters*, 98(19):190504, 2007. DOI: https://doi.org/10.1103/PhysRevLett.98.190504.

[43] Robert Raussendorf, Sergey Bravyi, and Jim Harrington. Long-range quantum entanglement in noisy cluster states. *Phys. Rev. A*, 71:062313, Jun 2005. DOI: https://doi.org/10.1103/PhysRevA.71.062313. URL https://link.aps.org/doi/10.1103/PhysRevA.71.062313.

[44] Robert Raussendorf, Jim Harrington, and Kovid Goyal. A fault-tolerant one-way quantum computer. *Annals of physics*, 321(9):2242–2270, 2006. DOI: https://doi.org/10.1016/j.aop.2006.01.012.

[45] Robert Raussendorf, Jim Harrington, and Kovid Goyal. Topological fault-tolerance in cluster state quantum computation. *New Journal of Physics*, 9(6):199, 2007. DOI: https://doi.org/10.1088/1367-2630/9/6/199.

[46] Ben W Reichardt. Postselection threshold against biased noise. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 420–428. IEEE, 2006. DOI: https://doi.org/10.1109/FOCS.2006.64.

[47] Ben W Reichardt. Fault-tolerant quantum error correction for steane's seven-qubit color code with few or no extra qubits. *Quantum Science and Technology*, 6(1):015007, 2020. DOI: https://doi.org/10.1088/2058-9565/abc6f4.

[48] Terry Rudolph. Why i am optimistic about the silicon-photonic route to quantum computing. *APL photonics*, 2(3):030901, 2017. DOI: https://doi.org/10.1063/1.4976737.

[49] C Ryan-Anderson, NC Brown, MS Allman, B Arkin, G Asa-Attuah, C Baldwin, J Berg, JG Bohnet, S Braxton, N Burdick, et al. Implementing fault-tolerant entangling gates on the five-qubit code and the color code. *arXiv preprint arXiv:2208.01863*, 2022. DOI: https://doi.org/10.48550/arXiv.2208.01863.

[50] Ciaran Ryan-Anderson, JG Bohnet, Kenny Lee, Daniel Gresh, Aaron Hankin, JP Gaebler, David Francois, Alexander Chernoguzov, Dominic Lucchetti, NC Brown, et al. Realization of real-time fault-tolerant quantum error correction. *Physical Review X*, 11(4):041058, 2021. DOI: https://doi.org/10.1103/PhysRevX.11.041058.

[51] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee, 1994. DOI: https://doi.org/10.1109/SFCS.1994.365700.

[52] Samuel C Smith, Benjamin J Brown, and Stephen D Bartlett. A local predecoder to reduce the bandwidth and latency of quantum error correction. *arXiv preprint arXiv:2208.04660*, 2022. DOI: https://doi.org/10.1103/PhysRevApplied.19.034050.

[53] Federico Spedalieri. Latency in fault-tolerant quantum computing with local, two-dimensional architecture. jun 2007. DOI: https://doi.org/10.48660/07060065. URL https://pirsa.org/07060065. PIRSA:07060065 see, https://pirsa.org.

[54] Thomas M Stace. Loss tolerance in topological quantum codes. In *International Conference on Quantum Information*, page QTuC4. Optica Publishing Group, 2011. DOI: https://doi.org/10.1364/ICQI.2011.QTuC4.

[55] Andrew M. Steane. Efficient fault-tolerant quantum computing. *Nature*, 399(6732):124–126, May 1999. ISSN 1476-4687. DOI: https://doi.org/10.1038/20127.

[56] Andrew M. Steane and Ben Ibinson. Fault-tolerant logical gate networks for calderbank-shor-steane codes. *Phys. Rev. A*, 72:052335, Nov 2005. DOI: https://doi.org/10.1103/PhysRevA.72.052335.

[57] Ashley M Stephens, William J Munro, and Kae Nemoto. High-threshold topological quantum error correction against biased noise. *Physical Review A*, 88(6):060301, 2013. DOI: https://doi.org/10.1103/PhysRevA.88.060301.

[58] Christophe Vuillot, Hamed Asasi, Yang Wang, Leonid P Pryadko, and Barbara M Terhal. Quantum error correction with the toric gottesman-kitaev-preskill code. *Physical Review A*, 99(3):032344, 2019. DOI: https://doi.org/10.1103/PhysRevA.99.032344.

[59] Chenyang Wang, Jim Harrington, and John Preskill. Confinement-higgs transition in a disordered gauge theory and the accuracy threshold for quantum memory. *Annals of Physics*, 303(1):31–58, jan 2003. DOI: 10.1016/s0003-4916(02)00019-2. URL https://doi.org/10.1016%2Fs0003-4916%2802%2900019-2.

[60] David S Wang, Austin G Fowler, and Lloyd CL Hollenberg. Surface code quantum computing with error rates over 1%. *Physical Review A*, 83(2):020302, 2011. DOI: https://doi.org/10.1103/PhysRevA.83.020302.

[61] Yang Wang. Quantum error correction with the gkp code and concatenation with stabilizer codes. *arXiv preprint arXiv:1908.00147*, 2019. DOI: https://doi.org/10.48550/arXiv.1908.00147.

[62] Fern HE Watson and Sean D Barrett. Log-

ical error rate scaling of the toric code. *New Journal of Physics*, 16(9):093045, 2014. DOI: https://doi.org/10.1088/1367-2630/16/9/093045.

[63] Yue Wu, Shimon Kolkowitz, Shruti Puri, and Jeff D Thompson. Erasure conversion for fault-tolerant quantum computing in alkaline earth rydberg atom arrays. *arXiv preprint arXiv:2201.03540*, 2022. DOI: https://doi.org/10.1038/s41467-022-32094-6.

[64] Hayata Yamasaki, Kosuke Fukui, Yuki Takeuchi, Seiichiro Tani, and Masato Koashi. Polylog-overhead highly fault-tolerant measurement-based quantum computation: all-gaussian implementation with gottesman-kitaev-preskill code. *arXiv preprint arXiv:2006.05416*, 2020. DOI: https://doi.org/10.48550/arXiv.2006.05416.

A Effective error model

The effective circuit-level error model is obtained by conjugating all single-qubit and two-qubit error operators to the end of the circuit. Here, the tables show the errors defined by the noise model and the resulting effective error after propagation.

Error	Step	Effective error
X_C	1	Z_W
$X_C Z_W$		
Z_W		Z_W
X_C	2	Z_W
$X_C Z_E$		Z_W, Z_E
Z_E		Z_E
X_C	3	Z_W, Z_E
$X_C Z_N$		Z_W, Z_E, Z_N
Z_N		Z_N
X_C	4	Z_W, Z_E, Z_N
$X_C Z_S$		Z_W, Z_E, Z_N, Z_S
Z_S		Z_S

Table 6: Propagated circuit-level errors for type I codes with the trivial circuit.

Error	Step	Effective error	Step	Effective error
X_C	1	Z_W	5	Z_E, Z_N, Z_S
$X_C Z_W$				Z_W, Z_E, Z_N, Z_S
Z_W		Z_W		Z_W
X_C	2	Z_W	6	Z_N, Z_S
$X_C Z_E$		Z_W, Z_E		Z_E, Z_N, Z_S
Z_E		Z_E		Z_E
X_C	3	Z_W, Z_E	7	Z_S
$X_C Z_N$		Z_W, Z_E, Z_N		Z_N, Z_S
Z_N		Z_N		Z_N
X_C	4	Z_W, Z_E, Z_N	8	Z_S
$X_C Z_S$		Z_W, Z_E, Z_N, Z_S		
Z_S		Z_S		Z_S

Table 7: Propagated circuit-level errors for the $[\![2,1,1]\!]$ scheme. The circuit employs the $[\![2,1,1]\!]$-detectable gate sequence demonstrated in the Fig. 5.

Error	Step	Effective error	Step	Effective error	Step	Effective error
X_C	1	Z_W	5	Z_W, Z_E, Z_N, Z_S	9	Z_E, Z_N, Z_S
$X_C Z_W$				Z_{W12}, Z_E, Z_N, Z_S		Z_W, Z_E, Z_N, Z_S
Z_W		Z_W		Z_W		Z_W
X_C	2	Z_W	6	Z_{W12}, Z_E, Z_N, Z_S	10	Z_S, Z_N
$X_C Z_E$		Z_W, Z_E		$Z_{W12}, Z_{E12}, Z_N, Z_S$		Z_E, Z_N, Z_S
Z_E		Z_E		Z_E		Z_E
X_C	3	Z_W, Z_E	7	Z_W, Z_E, Z_N, Z_{S12}	11	Z_S
$X_C Z_N$		Z_W, Z_E, Z_N		$Z_W, Z_E, Z_{N12}, Z_{S12}$		Z_S, Z_N
Z_N		Z_N		Z_N		Z_N
X_C	4	Z_W, Z_E, Z_N	8	Z_W, Z_E, Z_N, Z_S	12	Z_S
$X_C Z_S$		Z_W, Z_E, Z_N, Z_S		Z_W, Z_E, Z_N, Z_{S12}		
Z_S		Z_S		Z_S		Z_S

Table 8: Circuit-level errors for the $[\![3,1,1]\!]_1$ scheme. The circuit is $[\![3,1,1]\!]_1$-detectable by applying transversal gate sequences for 3 rounds.

B Scaling exponents

The scaling exponent ν obtained from fitting is listed as follows.

Inner codes	Cubic	$[\![4,1,1,2]\!]$	$[\![7,1,3]\!]$	$[\![2,1,1]\!]$	$[\![3,1,1]\!]_1$	$[\![3,1,1]\!]_2$
ν indep.	0.92(1)	1.28(6)	1.13(7)	1.40(9)	1.04(3)	1.21(3)
ν circ.	0.88(1)	1.26(3)	1.06(2)	1.29(7)	1.04(2)	1.04(3)

Table 9: Scaling exponents of the concatenation schemes against the phenomenological and circuit-level noise model.

C Threshold simulations

The raw data obtained from our numerical simulation is plotted in the following diagrams.

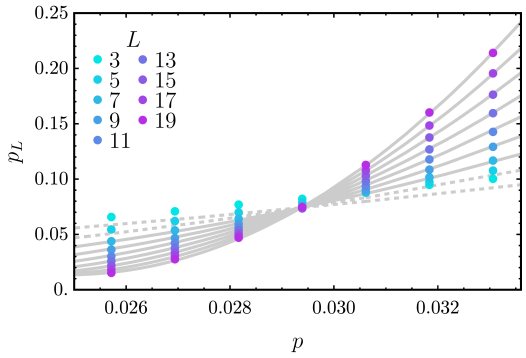


Figure 10: Logical error probability of the bcc scheme under the phenomenological noise model.

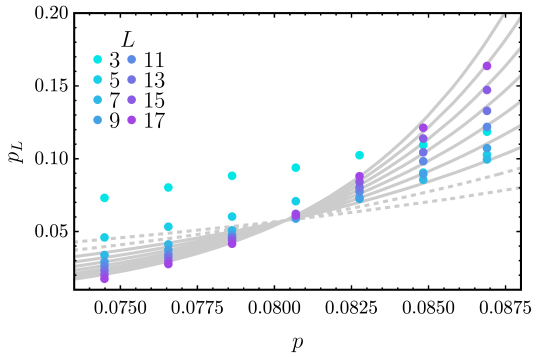


Figure 11: Logical error probability of the $[\![2, 1, 1]\!]$ scheme under the phenomenological noise model.

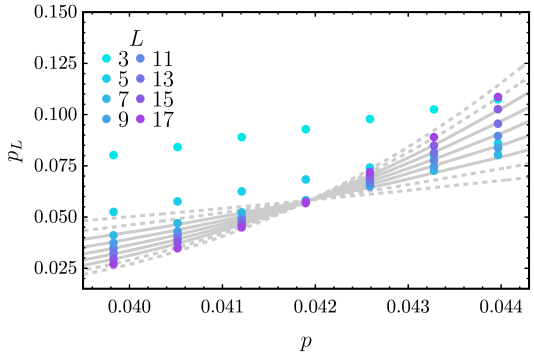


Figure 12: Logical error probability of the $[\![4, 1, 1, 2]\!]$ scheme under the phenomenological noise model.

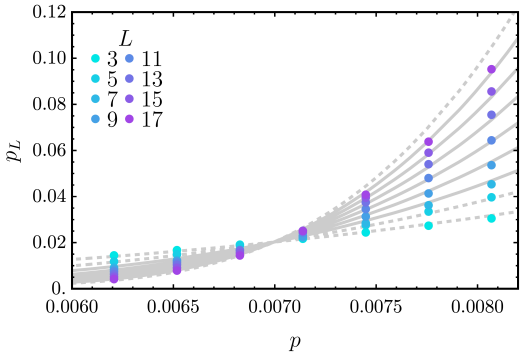


Figure 13: Logical error probability of the $[\![4, 1, 1, 2]\!]$ scheme under the circuit-level noise model.

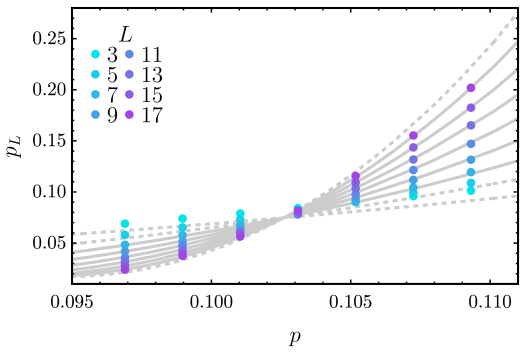


Figure 14: Logical error probability of the $[\![3,1,1]\!]_1$ scheme under the phenomenological noise model.

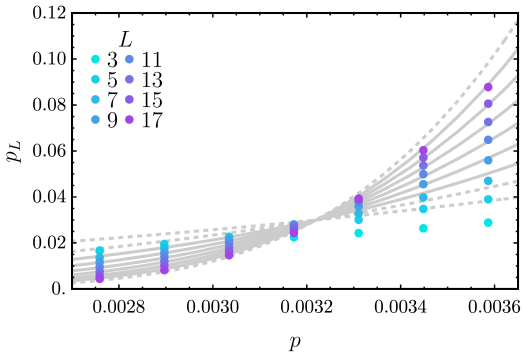


Figure 15: Logical error probability of the $[\![3,1,1]\!]_1$ scheme under the circuit-level noise model.

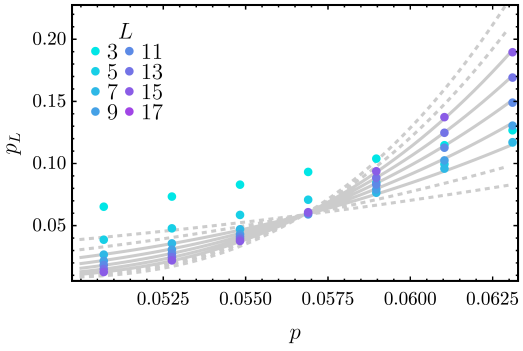


Figure 16: Logical error probability of the $[\![3,1,1]\!]_2$ scheme under the phenomenological noise model.

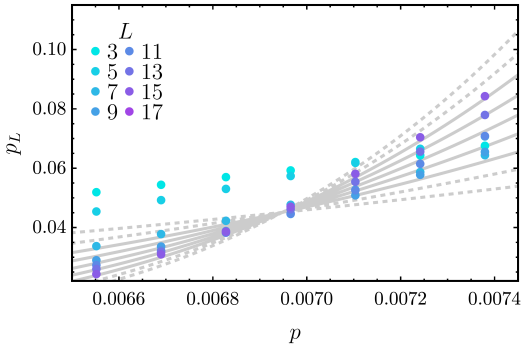


Figure 17: Logical error probability of the $[\![3,1,1]\!]_2$ scheme under the circuit-level noise model.

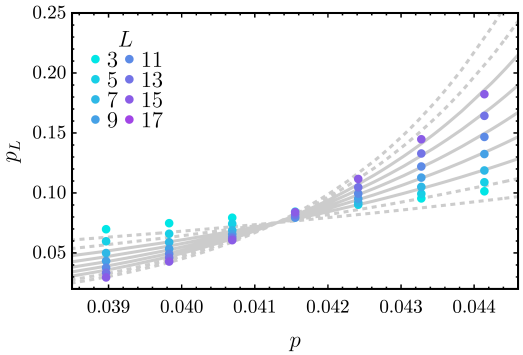


Figure 18: Logical error probability of the $[\![7,1,3]\!]$ scheme under the phenomenological noise model.

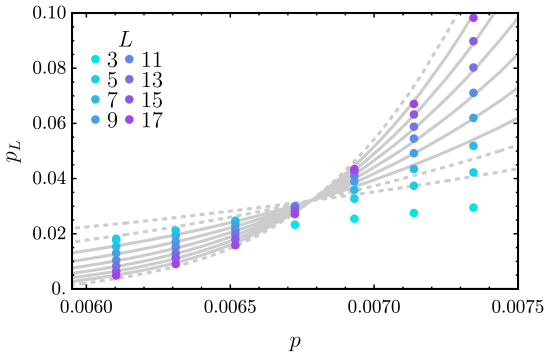


Figure 19: Logical error probability of the $[\![7,1,3]\!]$ scheme under the circuit-level noise model.