Hierarchical memories: Simulating quantum LDPC codes with local gates

Christopher A. Pattison[1], Anirudh Krishna[2,3], and John Preskill[1,4]

[1]Institute for Quantum Information and Matter, California Institute of Technology, Pasadena, CA 91125
[2]Department of Computer Science, Stanford University, Stanford, CA, 94305
[3]Stanford Institute for Theoretical Physics, Stanford University, Stanford, CA, 94305
[4]AWS Center for Quantum Computing, Pasadena CA 91125

April 3, 2025

Constant-rate low-density parity-check (LDPC) codes are promising candidates for constructing efficient fault-tolerant quantum memories. However, if physical gates are subject to geometric-locality constraints, it becomes challenging to realize these codes. In this paper, we construct a new family of $[\![N, K, D]\!]$ codes, referred to as hierarchical codes, that encode a number of logical qubits $K = \Omega(N/\log(N)^2)$. The N^{th} element $\mathcal{H}_N$ of this code family is obtained by concatenating a constant-rate quantum LDPC code with a surface code; nearest-neighbor gates in two dimensions are sufficient to implement the syndrome-extraction circuit $C_N^{\mathcal{H}}$ and achieve a threshold. Below threshold the logical failure rate vanishes superpolynomially as a function of the distance $D(N)$. We present a bilayer architecture for implementing $C_N^{\mathcal{H}}$, and estimate the logical failure rate for this architecture. Under conservative assumptions, we find that the hierarchical code outperforms the basic encoding where all logical qubits are encoded in the surface code.

1 Introduction

Quantum error-correcting codes encode quantum information in entangled states over many qubits. They are defined by a set of operators called stabilizer generators. Errors can accumulate in the state due to imperfect control and interactions with the environment. Stabilizer generators can be measured using syndrome-extraction circuits; the outcome of these measurements are called syndromes, classical information used to infer corrections to these errors. To minimize the probability of corrupting information beyond recovery, it is imperative to minimize the points of failure in the syndrome-extraction circuit. This can be realized by restricting the number of gates that each qubit interacts with and minimizing the total space-time volume of this circuit. The extent to which this can be done depends on the choice of error-correcting code and physical constraints.

Syndrome-extraction circuits are the workhorse of quantum memories, devices that can reliably store qubits for some fixed duration. In this paper, we are concerned with designing memories that can encode a growing number of qubits and simultaneously have a low probability of failure.[1] We focus on their design when qubits are embedded in a two-dimensional lattice and gates are subject to constraints on geometric locality.

Quantum low-density parity-check (LDPC) codes are natural candidates for constructing quantum memories. A quantum LDPC code refers to a family $\{\mathcal{Q}_t\}_t$ of $\{[\![n(t), k(t), d(t), \Delta_q, \Delta_g]\!]\}$ codes. This notation means that the t^{th} element in the family uses $n(t)$ data qubits to encode $k(t)$ logical

[1]We leave fault-tolerant *computation* for future work.

qubits and has distance $d(t)$, i.e. it is robust to $\lfloor (d(t)-1)/2 \rfloor$ Pauli errors. We assume that for all t, $n(t) > n(t-1)$. In what follows, we wish to focus on the dependence of k and d as functions of n. In this case, we implicitly parameterize the family using the size n of the code, i.e. we use the notation $\{\mathcal{Q}_n\}_n$, and we say that we are working with a $[\![n, k(n), d(n), \Delta_q, \Delta_g]\!]$ code family.

A quantum LDPC code is one where, for all codes in the code family, every stabilizer generator only involves at most a constant number Δ_g of qubits, and each qubit is supported within at most a constant number Δ_q of stabilizer generators. Such codes can encode a number of qubits that increases with the code size; simultaneously, the probability of *any* error on the encoded level is suppressed exponentially in the distance $d(n)$.

Furthermore, the syndrome-extraction circuit C_n can be efficient as measured by two figures of merit. The *depth* of the syndrome-extraction circuit is the number of time steps $\mathcal{T}(C_n)$ it takes to implement. The *width* of the syndrome-extraction circuit is the total number of qubits $\mathcal{W}(C_n)$ it uses (including ancilla qubits in addition to data qubits). The size or volume of the circuit is the product of the depth and the width. Building on a result by Kovalev and Pryadko [KP13], Gottesman [Got13] constructed fault-tolerant syndrome-extraction circuits that have volume which is a constant times the volume of the noise-free syndrome-extraction circuit — there exists a threshold q such that if gates fail with fixed probability $p < q$, the probability of the circuit failing falls exponentially in the distance $d(n)$.

However, realizing this architecture in a 2-dimensional layout is challenging. It requires high-fidelity gates acting on qubits that may be far apart. Some architectures might not support such interactions.

It is known that geometric locality severely constrains quantum error-correcting codes in 2 and 3 (Euclidean) dimensions. The most famous codes that are implemented using only geometrically-local gates are surface codes [Kit03; BK98] and color codes [BM06; KB15]. Seminal results by Bravyi and Terhal [BT09], and later by Bravyi, Poulin and Terhal [BPT10] showed that these codes are optimal for quantum LDPC codes defined using geometrically-local stabilizers. Subsequently, it was shown that to implement LDPC codes where the parameters k and d are both strictly better than the surface code, we require a growing amount of long-range connectivity [BK22a; BK22b].

When restricted to using only nearest-neighbor gates in 2 dimensions, Delfosse *et al.* [DBT21] proved the following tradeoff for syndrome-extraction circuits for constant-rate LDPC codes: [2]

$$\mathcal{T}(C_n) = \Omega\left(\frac{n}{\sqrt{\mathcal{W}(C_n)}}\right) , \tag{1}$$

where $\mathcal{T}(C_n)$ is the depth of the syndrome-extraction circuit and $\mathcal{W}(C_n)$ is the total number of qubits, data and ancilla, used in the circuit [3]. In words, this shows that given only nearest-neighbor gates to build a syndrome-extraction circuit for constant-rate LDPC codes, we can choose to minimize either the depth or the width of C_n, but cannot do both.

This sets the stage for presenting the main questions we address in this paper: does the family of circuits saturating Equation (1) still have a threshold? If not, how do we modify the code and associated circuit to achieve a threshold as efficiently as possible? How do we construct the most efficient syndrome-extraction circuits given access to gates whose range is more than merely nearest neighbor? Can we improve on the bound in Equation (1)?

[2] The bound applies to classes of codes that are called *locally expanding*. The exact definition of locally-expanding codes is not relevant; the interested reader is pointed to the paper by Delfosse *et al.* [DBT21]. For our purposes, it includes some important classes of quantum LDPC codes such as hypergraph product codes [TZ14] and good quantum LDPC codes [BE21; PK22; LZ22; LH22].

[3] For an explanation of $O(\cdot)$, $\Theta(\cdot)$ and $\Omega(\cdot)$ notation, please refer to Appendix A.

Our contributions

This paper is centered around the theme of implementing efficient quantum memories. Our main result is that our proposal, called a hierarchical code, has a threshold and that it achieves asymptotically better error suppression than the surface code. As it brings together a few different ideas, we present a short summary of each section and how to navigate the paper. Although these results build on each other, our presentation is modular — readers ought to be able to proceed to their section of choice after reading this overview and Section 2 where we define all the concepts required to formally state our results. (The statements of the main theorems of each section are only presented informally below.)

Section 3 : Permutation routing on graphs Connectivity beyond nearest-neighbor interactions is being explored in many architectures. There is evidence that some architectures can support gates of range R where R can be large [Leu+19; Per+21]. Motivated by these developments, we ask: given nearest-neighbor Clifford gates and SWAP gates of range R, can we reduce the depth of the syndrome-extraction circuit for constant-rate LDPC codes? To this end, we will permute qubits to bring them within range to apply an entangling gate. This is expressed as a *permutation routing*, a task on a graph $G = (V, E)$ specified by a permutation $\alpha : V \to V$. In this task, two vertices labeled u and v connected by an edge (u, v) are allowed to exchange labels within each step. The objective is to ensure that all labels match destinations $\alpha(u)$ while minimizing the total time required. Permuting vertices in parallel is non-trivial—the paths along which one permutes different pairs can overlap and thereby require more time. Section 3.1 reviews a permutation routing algorithm due to Annexstein and Baumslag [AB90]. This algorithm yields a permutation routing on a product of two graphs given permutation routings on each of the input graphs. In Section 3.2, we build on this algorithm to permute vertices on an $L \times L$ lattice where two vertices separated by a distance R are connected by an edge using a sparse subgraph. The main technical result of this section is the following existence result.

Theorem 1.1 (Permutation routing)**.** *For R even, there is an efficient construction of a degree-12 graph $G = (V, E)$ whose vertex set V is identified with an $L \times L$ lattice with edges of length at most R. Given a permutation $\alpha : V \to V$, a permutation routing implementing α can be performed in depth $3L/R + O(\log^2 R)$.*

While it is itself not the main result of our paper, it will be used in service of proving Theorem 1.2 which demonstrates the existence of efficient syndrome-extraction circuits given SWAP gates of range up to R (but not all gates need to have length equal to R). This section is entirely technical and only discusses graph properties and permutation routings.

Section 4 and Section 5: Hierarchical codes & the bilayer architecture Given access to only nearest-neighbor gates, Delfosse *et al.* present some evidence *against* the existence of a threshold if one were to permute qubits to bring them within range to perform a CNOT (see Figure 2 of [DBT21]). In particular, in the setting where $\mathcal{W}(C_n) = \Theta(n)$ and $\mathcal{T}(C_n) = \Theta(\sqrt{n})$, it appears too many errors accumulate before we can complete executing the syndrome-extraction circuit.

We circumvent this problem using code concatenation. We concatenate a constant-rate $[\![n, k, d, \Delta_q, \Delta_g]\!]$ LDPC code $\{\mathcal{Q}_n\}$ with a $[\![d_\ell^2, 1, d_\ell]\!]$ rotated surface code $\mathcal{RS}_\ell$ to obtain the *hierarchical code* $\{\mathcal{H}_N\}$ with parameters denoted $[\![N, K, D]\!]$. This means that each qubit of the syndrome-extraction circuit for the LDPC code $\mathcal{Q}_n$, henceforth referred to as the "outer code", is itself the logical qubit of a rotated surface code $\mathcal{RS}_\ell$, which we refer to as the "inner code" or sometimes as a "tile." As a rotated surface code can suppress errors exponentially in d_ℓ, we can suppress errors long enough to complete syndrome measurements of the outer quantum LDPC code using relatively small inner codes. The lattice length d_ℓ of the inner code only scales logarithmically in the size of the outer LDPC code, i.e. $d_\ell = \Theta(\log(n))$. Here ℓ indexes the qubits in the rotated surface code, $\ell^2 = 2d_\ell^2 - 1$. Section 4 is dedicated to the construction of syndrome-extraction circuits $C_N^{\mathcal{H}}$ corresponding to $\mathcal{H}_N$. The hierarchical code family $\{\mathcal{H}_N\}$ is not LDPC: The stabilizer generators for the outer code

act on a number of physical qubits that scales with the size d_ℓ of the inner code. However, local operations are sufficient to implement the corresponding syndrome-extraction circuit $C_N^{\mathcal{H}}$. The main result of this section is summarized in the following theorem.

Theorem 1.2. *The $[\![N, K, D]\!]$ hierarchical code $\mathcal{H}_N$ is constructed by concatenating an outer code, a constant-rate $[\![n, k, d, \Delta_q, \Delta_g]\!]$ quantum LDPC code $\mathcal{Q}_n$, and an inner code, a rotated surface code $\mathcal{RS}_\ell$ where $d_\ell = \Theta(\log(n))$. Let $\rho > 0$ and $\delta \geq 1/2$, such that $k = \rho \cdot n$ and $d = \Theta(n^\delta)$. The code $\mathcal{H}_N$ has parameters*

$$K(N) = \Omega\left(\frac{N}{\log^2(N)}\right) , \qquad D(N) = \Omega\left(\frac{N^\delta}{\log^{2\delta-1}[N/\log(N)]}\right) .$$

There exists an explicit and efficient construction of an associated family of syndrome-extraction circuits $C_N^{\mathcal{H}}$ constructed using only local Clifford operations and SWAP *gates of range R such that*

$$\mathcal{W}(C_N^{\mathcal{H}}) = O(N) , \qquad \mathcal{T}(C_N^{\mathcal{H}}) = O\left(\frac{\sqrt{N}}{R}\right) .$$

Our construction works for all values of $\delta > 0$; we choose $\delta \geq 1/2$ to make theorem statements simpler. Before describing how the circuit $C_N^{\mathcal{H}}$ is constructed, we motivate why it is interesting—it has a threshold.

We work in a model where errors occur in a stochastic manner. We declare a logical failure if *any* of the K encoded qubits fail. More generally, we declare failure if any logical error occurs on the code space. The main result of Section 5 is the following theorem.

Theorem 1.3 (Informal)**.** *Consider the $[\![N, K, D]\!]$ family of hierarchical codes $\mathcal{H}_N$ and the associated family of syndrome-extraction circuits $C_N^{\mathcal{H}}$. Suppose the outer code $\mathcal{Q}_n$ has constant rate $k = \rho n$ and distance $d(n) = \Theta(n^\delta)$. If we repeat the syndrome-extraction circuit $C_N^{\mathcal{H}}$ for $d(n)$ rounds, then there exists a threshold $q \in (0, 1]$ corresponding to $C_N^{\mathcal{H}}$ such that, if each gate fails with fixed probability $0 < p < q$, then the probability of logical failure under minimum-weight decoding, $p_{\mathcal{H}}(N)$, obeys*

$$p_{\mathcal{H}}(N) \leq \exp\left(-c_{\mathcal{H}} \cdot \frac{N^\delta}{\log^{2\delta}(N)}\right) ,$$

for some positive number $c_{\mathcal{H}}$ independent of N.

The theorem is only stated informally here because we have not yet defined the noise model with respect to which this result holds. We will consider a *locally decaying error model* to account for correlated errors that may occur in a circuit. This error model is defined in Section 2.2. Section 5 is dedicated to a proof of the existence of a threshold. We build on Gottesman's proof of the existence of a threshold for syndrome-extraction circuits (Theorem 4 of [Got13]). The central idea is the requirement that the probability of failure for a qubit *per round of syndrome extraction*, denoted p_{round}, remains a sufficiently small constant. This is reviewed in Section 2.4. Gottesman's result was based on syndrome-extraction circuits for LDPC codes that have constant depth. As Equation (1) highlights, this is not possible when subject to locality constraints. We study the dependence of p_{round} on the circuit depth in Section 5.1. In Section 5.3, we show that $\ell = \Theta(\log(n))$ is sufficient for $C_N^{\mathcal{H}}$ to have a threshold.

As we ask to minimize circuit width $\mathcal{W}(C_N^{\mathcal{H}})$ *and* subject the circuit to locality constraints, we pay a price — in addition to the growing depth, the number of encoded qubits $K(N)$ and distance $D(N)$ are suppressed by polylogarithmic factors in n relative to the outer code $\mathcal{Q}_n$ which has constant rate and distance $d(n) = \Theta(n^\delta)$. Furthermore, for fixed gate error rates $p \in [0, 1]$, the sub-threshold scaling of the logical error rate $p_{\mathcal{H}}(N)$ of $\{\mathcal{H}_N\}$ is subexponential, but superpolynomial, in the distance $D(N)$; for any positive constants α, β, the logical failure probability $p_{\mathcal{H}}(N)$ vanishes faster than any polynomial function $N^{-\beta}$ but slower than any exponential function $\exp(-\alpha \cdot N)$:

$$\frac{p_{\mathcal{H}}(N)}{N^{-\beta}} \xrightarrow{N\to\infty} 0 , \qquad \frac{p_{\mathcal{H}}(N)}{\exp(-\alpha \cdot N)} \xrightarrow{N\to\infty} \infty .$$

Having motivated why we are interested in $\mathcal{H}_N$, we return to the construction of $C_N^{\mathcal{H}}$. In Section 4, we propose a novel bilayer architecture to implement it. We begin the section by presenting the syndrome-extraction circuit C_n for the constant-rate $[\![n, k, d, \Delta_q, \Delta_g]\!]$ LDPC code. Physical qubits are arranged in two parallel layers, each a lattice of side length approximately $L = \Theta(\sqrt{n})$. To obtain the syndrome-extraction circuit $C_N^{\mathcal{H}}$ for the concatenated code, each of the $\mathcal{W}$ qubits in $C_n^{\mathcal{Q}}$ is replaced by a rotated surface code.

In Section 4.2, we describe how to arrange $\mathcal{W} = \mathcal{W}(C_n^{\mathcal{Q}})$ surface codes $\mathcal{RS}_\ell$ in a bilayer architecture. Each layer now has side length approximately $2L\ell$ qubits to accommodate the tiles. An instance of a single layer is shown in Figure 1 (a). We assume access to nearest-neighbor physical Clifford operations and SWAP gates of range R within a layer and Clifford operations between adjacent qubits in different layers. These physical qubits are aggregated into $[\![d_\ell^2, 1, d_\ell]\!]$ codes $\mathcal{RS}_\ell$. See Figure 1 (b). There are $2L^2$ tiles in total. Even though we are only implementing a quantum memory, we still need to understand how to perform a limited set of logical operations on tiles to implement the syndrome-extraction circuit for the outer code. The advantage of the bilayer architecture is that it allows for transversal CNOT to implement logical CNOT. We propose a new technique to perform logical SWAP operations between tiles. This yields all required logical Clifford operations between tiles to perform syndrome-extraction for the outer code.

We note that the existence of a threshold does not depend on using the bilayer architecture. For example, tiles can be arranged in a single layer and Clifford gates can be implemented via lattice surgery [Lit19; HFDM12]. For an alternative implementation in the context of measurement-based quantum computation, see [Bom+21]. Although we do not prove it here, it is possible to show that a threshold exists also in this setting using similar techniques.

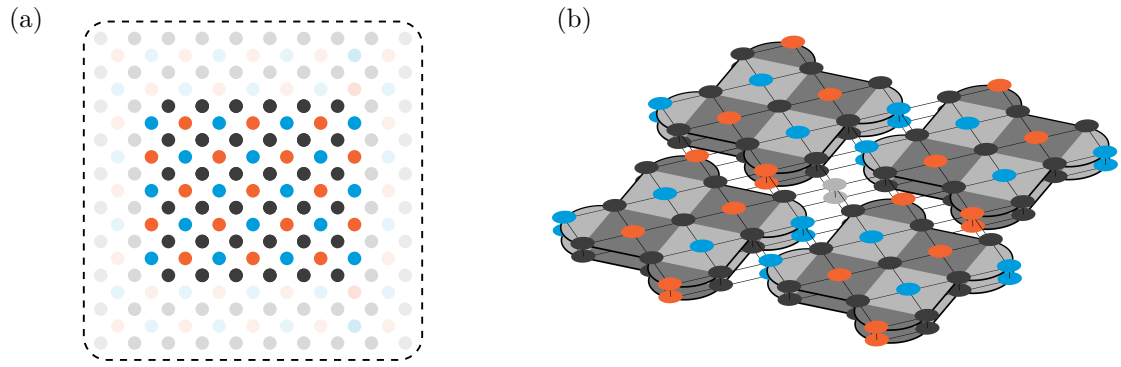

Figure 1: The bilayer architecture used to implement the syndrome-extraction circuit $C_N^{\mathcal{H}}$ for the hierarchical code $\mathcal{H}_N$. (a) represents a single layer of the bilayer architecture. Colored dots represent syndrome qubits and gray dots represent data qubits. Transparent dots represent inactive qubits. At any given time step, the qubits that participate in the circuit are depicted as opaque dots and form a lattice of side length $L\ell$; its location within the larger lattice can shift relative to the second layer. This is used to facilitate logical Clifford operations. (b) represents parallel tiles of distance d_ℓ. Each tile represents an outer qubit of the hierarchical code construction. Light gray dots will be used to facilitate Clifford operations but are not used in the syndrome-extraction circuit for $\mathcal{RS}_\ell$.

The circuits $C_N^{\mathcal{H}}$ are constructed such that each lattice position remains connected to a fixed and constant-sized set of other lattice positions for any $R = \omega(1)$. Furthermore, the connectivity does not change dynamically over the course of the circuit. This way, the wiring can be decided ahead of time.

Section 6 : Comparisons to surface code Finally, we compare the hierarchical memory $\mathcal{H}_N$ with a simple memory that only uses rotated surface codes. At the outset, it may seem unclear whether the use of extra resources to execute the constant-rate LDPC code's syndrome-extraction circuit can be better spent simply building better surface codes which are ideally suited for 2-dimensional local interactions. We let $\{\mathcal{B}_M\}$ refer to the *basic* encoding where each logical qubit is encoded in a separate surface code; for some distance d_M, we let $\mathcal{B}_M$ be the K-fold product of

the surface code, i.e. $\mathcal{B}_M = \mathcal{RS}_{\ell_M}^{\otimes K}$. The index M represents the total number of qubits in this encoding, i.e. $M = \Theta(Kd_M^2)$. To contrast $\mathcal{H}_N$ and $\mathcal{B}_M$, we present both an asymptotic comparison as well as numerical estimates based on some conservative assumptions.

Theorem 1.4 (Informal)**.** *Let $\mathcal{H}_N$ be a specific $[\![N, K, D]\!]$ hierarchical code family such that the (outer) constant-rate LDPC code $\mathcal{Q}_n$ has distance $d = \Theta(n^\delta)$. Let $\mathcal{B}_M$ be the basic encoding $\mathcal{RS}_{\ell_M}^{\otimes K}$ that encodes K qubits in separate rotated surface codes of distance d_M. Let $C_M^{\mathcal{B}}$ be the corresponding family of syndrome-extraction circuits for $\mathcal{B}_M$. Let $p_{\mathcal{B}}(M)$ denote the logical failure probability under minimum-weight decoding for $\mathcal{B}_M$ where we declare failure if* any *logical qubit fails. Suppose the gate error rate p is below the thresholds for both the basic encoding and the hierarchical code. To achieve $p_{\mathcal{B}}(M) < p_{\mathcal{H}}(N)$, we require*

$$\mathcal{W}(C_M^{\mathcal{B}}) = \Omega\left[\left(\frac{N}{\log(N)}\right)^{1+2\delta}\right], \qquad \mathcal{T}(C_M^{\mathcal{B}}) = \Omega\left[\left(\frac{N}{\log^2(N)}\right)^{\delta}\right].$$

We can compare this with parameters for $C_N^{\mathcal{H}}$ from Theorem 1.2. For all $\delta > 0$, the width $\mathcal{W}$ of $C_N^{\mathcal{H}}$ is less than that of $C_M^{\mathcal{B}}$. Furthermore, if the outer code has a single-shot decoder, i.e. if a constant number of applications of $C_N^{\mathcal{H}}$ are sufficient to achieve a threshold, then the depth $\mathcal{T}$ of $C_N^{\mathcal{H}}$ is also less than that of $C_M^{\mathcal{B}}$. Efficient single-shot decoders are known to exist for constant-rate LDPC codes [LTZ15; FGL18a; FGL18b].

Having said this, it is unclear whether this advantage manifests for practically-relevant code sizes and error rates. To make such a comparison, we use numerical estimates. We choose the size $M = M(N)$ such that the syndrome-extraction circuits for the hierarchical scheme $\mathcal{H}_N$ and the basic encoding $\mathcal{B}_M$ use the same number of physical qubits. Fixing the total number of qubits in this manner, we look for a *crossover point*, the gate error rate q_0 at which the hierarchical code achieves a lower logical failure rate than the basic encoding.

We estimate the circuit-level failure rate using some assumptions about the sub-threshold scaling of the logical failure rate for LDPC codes. We assume the threshold of the surface code is 10^{-2} and the threshold for constant-rate LDPC codes under circuit-level noise is 10^{-3}. Our model takes into consideration how the logical failure rate depends on the depth of the circuit $C_N^{\mathcal{H}}$, and how hook errors could reduce the effective distance. Hook errors are harmful errors that spread from the ancilla qubits to the data qubits during syndrome extraction. These are explained in Section 6.2.3.

We offer evidence that against circuit-level depolarizing noise, the crossover happens at a gate error rate as high as 5×10^{-3} depending on the choice of outer code family and inner/outer code sizes. See the left-most plot in Figure 2. These numbers are merely a proof-of-concept and depend on the aforementioned assumptions which are discussed in Section 6.2.

We arrive at these estimates assuming all gates fail with the same probability. While such an assumption is convenient for proofs, in some architectures, it may be possible to perform SWAP operations with higher fidelity than CNOT or CZ [End+16; Bar+16; Blu+22; Hen+06; Kau+17]. For example, in ion trap and neutral atom trap architectures, SWAP gates can be performed by moving the traps. The mechanism is entirely different than that used to perform other two-qubit gates and, in principle, could have much better fidelity. These considerations are especially important to us as the main source of noise in the hierarchical scheme stems from SWAP gates. We present variations of our numerical estimates when the SWAP gates have better fidelity than the CNOT gates. The middle plot and right-most plot in Figure 2 represent estimates for the failure rate when the SWAP gates are $10\times$ and $100\times$ better than entangling gates respectively.

As mentioned, our estimates are predicated on some assumptions. We re-examine these assumptions in Section 6.4 and propose ways to improve the failure rate for hierarchical codes. We show how we can reduce the effect of hook errors by designing noise-biased qubits. A qubit is said to have a noise bias if X and Y errors are suppressed with respect to Z errors. We can introduce a bias on Level-1 qubits using unbiased Level-0 (physical) qubits. As the inner code is a surface code, we can engineer a bias simply by making the surface code longer in one direction of our choosing. See Figure 3 (a). Based on our estimates, we expect this can reduce the size of the code considerably. Figure 3 (b) shows the crossover points for the hierarchical code and the basic encoding with the

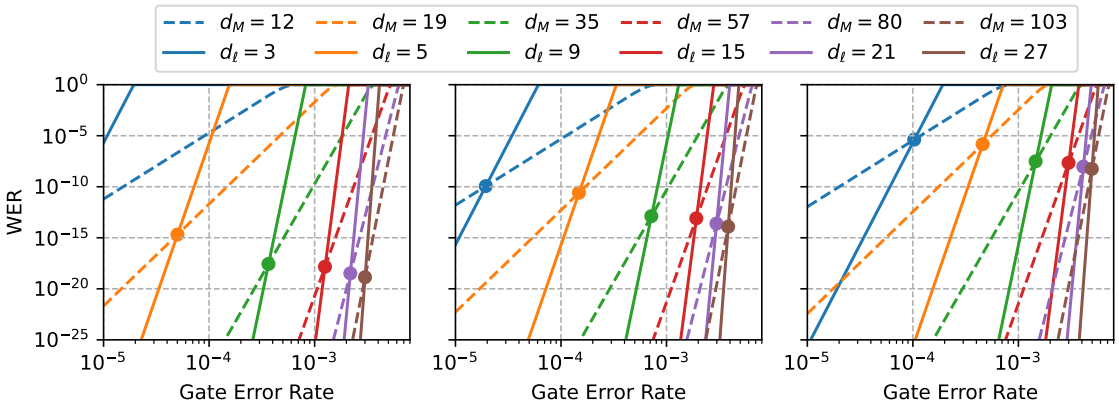

Figure 2: Comparing the logical failure rate for the hierarchical memory versus the logical failure rate for the basic encoding. The outer LDPC code has parameters $[\![1\ 116\ 416, 112\ 896, 119]\!]$. Each color represents an inner code of distance d_ℓ. The solid and dashed lines are estimates for the WER for the hierarchical memory and basic encoding respectively. The legend shows the size of the surface codes in each setting. For example, the solid blue line represents a hierarchical code with inner code lattice length $d_\ell = 3$. The dashed blue line represents a basic encoding that uses surface codes of lattice length 12. The three panels correspond to three different assumptions about the error rate in SWAP gates, as described in the text. In the left-most plot, SWAP gates are assumed to fail at the same rate as entangling gates. In contrast, in the middle and right plots, SWAP gates have a fidelity $10\times$ and $100\times$ better than entangling gates respectively.

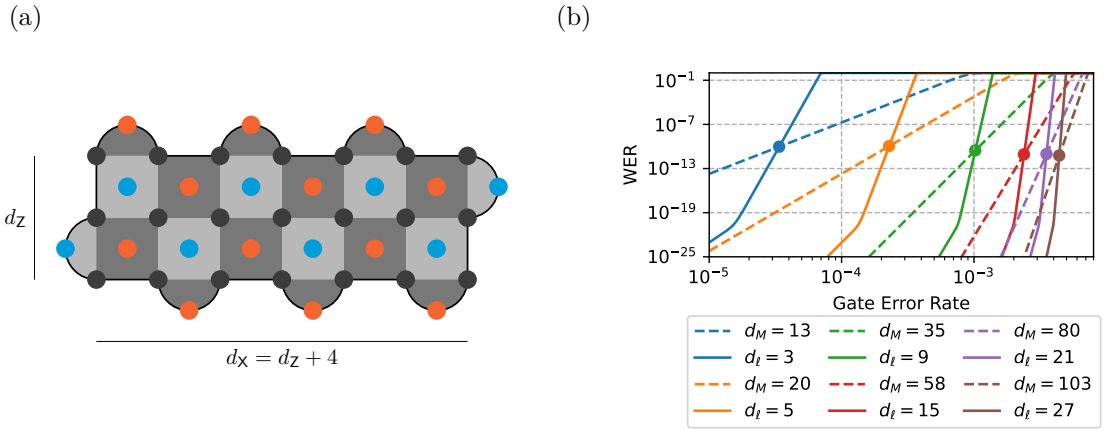

Figure 3: (a) Creating Level-1 qubits such that the probability of logical X failure is less than the probability of Z failure. This is accomplished by changing the aspect ratio of the tiles. (b) Estimating crossover points when SWAP gates are $10\times$ better than entangling gates.

assumption that SWAP gates are 10× better than entangling gates using a much smaller outer code.

Secondly, we believe that decoders for the hierarchical code can take advantage of their concatenated structure. To achieve this, we propose using message-passing decoders between the outer and inner codes. These ideas can be used in soft decoders for the outer code to partially overcome the problems of degeneracy [PC08]. Similar ideas have proved useful in the context of concatenating GKP codes and LDPC codes [Rav+22].

Lastly, we expect the hierarchical scheme to be resistant to *burst* errors. Unlike typical errors which affect only one or two qubits at a time, burst errors can wipe out entire patches of qubits. This can happen when there exists poorly localized error mechanisms such as the absorption of cosmic rays in superconducting circuits [Vep+20; McE+21; Tho+22; Ach+23; Car+23] or blackbody radiation mediated transitions to other Rydberg states in neutral atom platforms [Fes+22; Zei+16]. Large deviations may also occur in single points of failure in the control hardware such as power supplies, local oscillators, lasers, etc [Aru+19; Kra+19; Mad+20; Eba+21; Ma+22]. Protection of surface codes from burst errors was initially studied in [Xu+22] by concatenating a small constant-sized stabilizer code with surface codes. The hierarchical scheme is robust to these errors because each inner surface code represents a qubit of the outer code which we know is resistant to some number of erasure errors.

Related work: Gottesman [Got00] demonstrated that it is possible to find a threshold using only local gates and concatenation. Svore, Divincenzo and Terhal [STD05; SDT06] studied this issue further and established a numerical lower bound on the threshold in a scheme with many layers of concatenation. Yamasaki and Koashi [YK24] show that concatenated codes can be used to achieve constant overhead quantum computation that is also time efficient.

In contrast to these approaches, we consider a qualitatively different setting. In our hierarchical model, the concatenated code has only two layers. The outer LDPC code grows quickly to improve the error rate, while the inner code grows slowly to achieve a threshold. The number of encoded logical qubits in the code therefore increases (sublinearly) with the size of the code. Consequently, the rate of error suppression is significantly better.

Finally, Baspin *et al.* [BFS23] have recently generalized the result of Delfosse *et al.* in another direction. In contrast to the constructive approach in this paper, they approach this problem top-down — given access to *arbitrary* local operations and classical communication (not merely Clifford operations), they study syndrome-extraction circuits for LDPC codes and their ability to suppress stochastic errors. They prove the existence of a tradeoff between the parameters of the syndrome-extraction circuit and the sub-threshold error scaling (See Theorem 28 of [BFS23]). For fixed gate error rate p, suppose we use an $[\![N, K, D]\!]$ code $\mathcal{H}_N$ and desire a sub-threshold scaling of the logical failure rate $p_{\mathcal{H}}(N) = \exp(-f(N))$ for some function $f(N)$. Let C_N be the corresponding family of syndrome-extraction circuits. Assuming $f(N) = O(N)$, we express Theorem 28 of [BFS23] in our notation

$$\frac{\mathcal{W}(C_N)}{K} = \Omega\left(\frac{\sqrt{f(N)}}{\mathcal{T}(C_N)}\right) . \tag{2}$$

To compare with our result, suppose we only use SWAP gates of constant range, i.e. $R = O(1)$. From Theorem 1.2, the syndrome-extraction circuit $C_N^{\mathcal{H}}$ achieves $p_{\mathcal{H}}(N) = \exp(-\Theta(N^\delta/\log^{2\delta}(N))$ with $\mathcal{W}(C_N^{\mathcal{H}}) = \Theta(N)$ and $\mathcal{T}(C_N^{\mathcal{H}}) = O(\sqrt{N})$.

$$\frac{\mathcal{W}(C_N^{\mathcal{H}})}{K} = O(\log(N)) , \qquad \frac{\sqrt{f(N)}}{\mathcal{T}(C_N^{\mathcal{H}})} = O\left(\frac{N^{(\delta-1)/2}}{\log^{\delta}(N)}\right) . \tag{3}$$

Comparing with Equation (2), we can see that the bound is satisfied for any constant $\delta > 0$. Note that such a low logical error rate is only feasible because our syndrome-extraction circuit $C_N^{\mathcal{H}}$ has polynomially growing depth.

2 Background & Notation

In this section, we begin by formally defining concepts needed to state our results. Section 2.1 defines syndrome-extraction circuits. We review gadgets used to construct them and how to use these gadgets to obtain a syndrome-extraction circuit given an error correcting code. Section 2.2 reviews locally decaying distributions that describe errors on states and faults on circuits. These are general error models that can describe the types of correlated errors that we might witness in a circuit. A noise model is parameterized by a failure rate which quantifies the probability of errors. We described how error correcting codes and their associated syndrome-extraction circuits are robust to some amount of errors occurring below a *threshold* failure rate. Section 2.3 reviews syndrome-extraction circuits for concatenated codes. The hierarchical code is constructed by concatenating a constant-rate quantum LDPC code and the surface code. These are defined in Section 2.4 and Section 2.5 respectively. We review Gottesman's requirements [Got13] for the existence of a threshold. This will be an important idea in the proof of the existence of a threshold for hierarchical codes.

2.1 Basic definitions

Let $\mathcal{P} = \langle \mathsf{X}, \mathsf{Z} \rangle / \{\pm i, \pm 1\}$ denote the (projective) single-qubit Pauli group (where we ignore phases); for $n \in \mathbb{N}$, let $\mathcal{P}_n$ denote the n-fold tensor product $\mathcal{P}^{\otimes n}$. For $\mathsf{P} \in \mathcal{P}_n$, $\mathrm{supp}(\mathsf{P}) \subseteq [n]$ denotes the support of P, i.e. the set of qubits on which P acts non-trivially. The *weight* of a Pauli operator P is $|\mathrm{supp}(\mathsf{P})|$, the number of qubits in its support. For brevity, we denote this as $|\mathsf{P}|$.

For $\mathbf{a}, \mathbf{b} \in \{0,1\}^n$, let $\mathsf{X}(\mathbf{a}) = \otimes_i \mathsf{X}^{a_i}$, and $\mathsf{Z}(\mathbf{b}) = \otimes_j \mathsf{Z}^{b_j}$. Any Pauli operator $\mathsf{P} \in \mathcal{P}_n$ can be expressed uniquely as $\mathsf{P} = \mathsf{X}(\mathbf{a})\mathsf{Z}(\mathbf{b})$ for $\mathbf{a}, \mathbf{b} \in \{0,1\}^n$. We use $\mathsf{P}|_{\mathsf{X}}, \mathsf{P}|_{\mathsf{Z}} \in \{0,1\}^n$ to denote the X and Z components of P respectively, i.e. $\mathsf{P}|_{\mathsf{X}} := \mathbf{a}$ and $\mathsf{P}|_{\mathsf{Z}} := \mathbf{b}$.

Stabilizer codes: An n-qubit quantum error correcting code is the simultaneous $+1$-eigenspace of a set of commuting Pauli operators. These Pauli operators form a subgroup $\mathcal{S}$ of the Pauli group called the stabilizer group. The stabilizer group $\mathcal{S}$ is generated by elements $\mathsf{S}_1, ..., \mathsf{S}_m$. The codespace $\mathcal{Q}$ is then defined as

$$\mathcal{Q} = \{|\psi\rangle \in (\mathbb{C}^2)^{\otimes n} |\ \mathsf{S}_i |\psi\rangle = |\psi\rangle \ \ \forall i \in [m]\} \ .$$

The number of encoded qubits k is the base 2 logarithm of the number of linearly-independent vectors in $\mathcal{Q}$. Equivalently, given $\mathcal{S}$, it is simply $k = n - m$ (where we assume the stabilizer generators are independent).

Intuitively, the minimum distance d of the code is the minimum weight Pauli operator such that we can map one element of $\mathcal{Q}$ to a distinct element of $\mathcal{Q}$. Formally, we write

$$d = \min_{\substack{\mathsf{P} \in \mathcal{P}_n \setminus \mathcal{S} \\ [\mathsf{P}, \mathsf{S}_i] = 0}} |\mathsf{P}| \ .$$

We say such a code is an $[\![n, k, d]\!]$ (stabilizer) code.

The code is said to be a CSS code if every generator can be chosen such that it is a tensor product of only X or Z Pauli operators [CS96; Ste96]. We can define the X- and Z-distances d_{X} and d_{Z} of a CSS code as

$$d_{\mathsf{X}} = \min_{\substack{\mathsf{P} \in \{\mathsf{I},\mathsf{Z}\}^{\otimes n} \setminus \mathcal{S} \\ [\mathsf{P}, \mathsf{S}_i] = 0}} |\mathsf{P}| \qquad d_{\mathsf{Z}} = \min_{\substack{\mathsf{P} \in \{\mathsf{I},\mathsf{X}\}^{\otimes n} \setminus \mathcal{S} \\ [\mathsf{P}, \mathsf{S}_i] = 0}} |\mathsf{P}| \ .$$

Let $1 \leq b \leq m_{\mathsf{X}}$ and $1 \leq c \leq m_{\mathsf{Z}}$ index the X-type and Z-type stabilizer generators $\{\mathsf{S}_b^{\mathsf{X}}\}$ and $\{\mathsf{S}_c^{\mathsf{Z}}\}$.

Syndrome-extraction circuits & measurement gadgets: A syndrome-extraction circuit C for a code $\mathcal{Q}$ can be composed of the following elements that are allowed to be classically controlled.

Definition 2.1 (Clifford operations)**.** *Consider a set of qubits arranged in a lattice in* 2 *dimensions. We define the set* $\mathcal{K}$ *of* elementary Clifford operations *as follows:*

1. *Initialization of new qubits in state* $|0\rangle$ *or* $|+\rangle$*,*
2. *Single-qubit Pauli gates,*
3. *Two-qubit Clifford gates* CNOT *between nearest-neighbor qubits,*
4. *Single-qubit Pauli* X *and* Z *measurements,*
5. *Physical* SWAP *operation with range up to* R*.*

At any given time step, a qubit in C can be involved in at most one of these operations. In addition, we assume instantaneous classical communication and access to classical computation for processing measurement data.

To obtain the syndrome, we use gadgets to measure Pauli operators which are described as follows. Consider a CSS code $\mathcal{Q}$ with m_{X} X-type stabilizer generators $\mathcal{S}^{\mathsf{X}} = \{\mathsf{S}_b^{\mathsf{X}}\}_{b=1}^{m_{\mathsf{X}}}$ and m_{Z} Z-type stabilizer generators $\mathcal{S}^{\mathsf{Z}} = \{\mathsf{S}_b^{\mathsf{Z}}\}_{c=1}^{m_{\mathsf{Z}}}$. The entire set of stabilizer generators is $\mathcal{S} = \mathcal{S}^{\mathsf{X}} \cup \mathcal{S}^{\mathsf{Z}}$.

1. **For** $1 \leq b \leq m_{\mathsf{X}}$, measure a product of X operators:
   (a) Initialize the b^{th} ancilla qubit in $|+\rangle_b$.
   (b) Perform a CNOT gate controlled on the b^{th} ancilla qubit and targeted on each qubit in the support of $\mathsf{S}_b^{\mathsf{X}}$.
   (c) Perform a measurement of the b^{th} ancilla in the X basis.
2. **For** $1 \leq c \leq m_{\mathsf{Z}}$**,** measure a product of Z operators:
   (a) Initialize the c^{th} ancilla qubit in $|0\rangle_c$.
   (b) Perform a CNOT gate targeted on the c^{th} ancilla qubit and controlled on each data qubit in the support of $\mathsf{S}_c^{\mathsf{Z}}$.
   (c) Perform a measurement of the c^{th} ancilla in the Z basis.

Figure 4 illustrates gadgets for measuring an X operator of weight 5 and a Z operator of weight 4. Given a circuit C, we use two figures-of-merit to quantify its size:

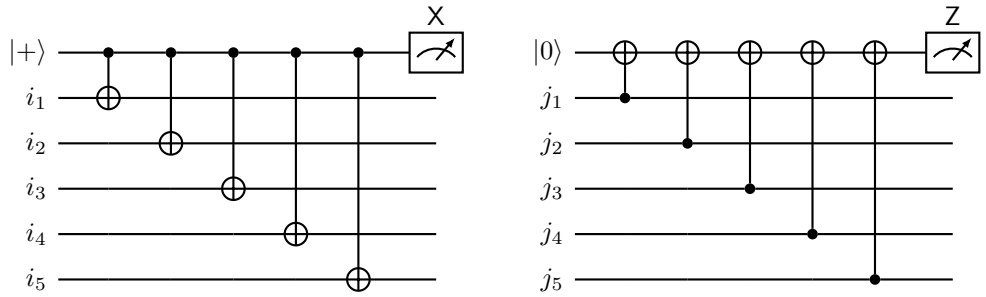

Figure 4: Performing the syndrome extraction corresponding to the operator $\mathsf{X}_{i_1}\mathsf{X}_{i_2}\mathsf{X}_{i_3}\mathsf{X}_{i_4}\mathsf{X}_{i_5}$ on the left and $\mathsf{Z}_{j_1}\mathsf{Z}_{j_2}\mathsf{Z}_{j_3}\mathsf{Z}_{j_4}\mathsf{Z}_{j_5}$ on the right. The measurements are performed on some qubits $\{i_1, ..., i_5, j_1, ..., j_5\} \subseteq [n]$.

1. $\mathcal{W}(C)$: the width of the circuit, i.e. the total number of physical qubits, data and ancilla, used in the circuit.
2. $\mathcal{T}(C)$: the depth of the circuit, i.e. the number of time steps required to measure all syndromes.

We assume operations in $\mathcal{K}$ can be performed in parallel (subject to the constraint that each qubit is only involved in one operation at a time). We shall present one way of using parallel operations to build efficient syndrome-extraction circuits for quantum LDPC codes in Section 3.

2.2 Noise & imperfect syndrome-extraction circuits

In practice, C is imperfect. In general, errors on multiple locations with complicated correlations can arise at the end of a syndrome-extraction circuit. Under the action of a two-qubit gate for instance, single-qubit errors which occur with probability p can transform into two-qubit correlated errors which occur with probability p. Two-qubit gates themselves can fail and introduce errors on both qubits where there were none before. As yet another example, small clusters of qubits that are near each other can also fail together, for example, due to crosstalk, stray magnetic fields, etc. These errors are outside the scope of an i.i.d. errors model and hence, we consider a generalization.

Definition 2.2. *Let $n \in \mathbb{N}$ and $\mathrm{Pow}(n) = \{E : E \subseteq [n]\}$. Consider a probability distribution $\widehat{\Pr} : \mathrm{Pow}(n) \to [0,1]$ and for $E \subseteq [n]$, let $\Pr(E)$ be the total probability*

$$\Pr(E) = \sum_{E' \supseteq E} \widehat{\Pr}(E') \ .$$

We say the distribution $\Pr$ is locally decaying with rate $p \in [0,1]$ if for all $E \subseteq [n]$,

$$\Pr(E) \leq p^{|E|} \ .$$

We first consider general errors on an n-qubit state. We assume every set of qubits has some probability of being corrupted by an arbitrary Pauli error. Consider a Pauli operator $\mathsf{E}' \in \mathcal{P}_n$ such that $\mathsf{E}' = \mathsf{X}(\mathbf{x}')\mathsf{Z}(\mathbf{z}')$. Let $\widehat{\mathcal{E}}(\mathbf{x}', \mathbf{z}')$ be the probability of the error E'. By definition, $\widehat{\mathcal{E}}$ is itself a map from $\mathrm{Pow}(n) \times \mathrm{Pow}(n)$ to $[0,1]$. Let $\mathcal{X}(\mathbf{x}) : \mathrm{Pow}(n) \to \mathbb{R}$ and $\mathcal{Z}(\mathbf{z}) : \mathrm{Pow}(n) \to \mathbb{R}$ denote

$$\mathcal{X}(\mathbf{x}) = \sum_{\mathbf{x}' \supseteq \mathbf{x}} \sum_{\mathbf{z}'} \widehat{\mathcal{E}}(\mathbf{x}', \mathbf{z}') \ , \qquad \mathcal{Z}(\mathbf{z}) = \sum_{\mathbf{x}'} \sum_{\mathbf{z}' \supseteq \mathbf{z}} \widehat{\mathcal{E}}(\mathbf{x}', \mathbf{z}') \ . \tag{4}$$

In other words, $\mathcal{X}$ and $\mathcal{Z}$ denote the probability that a random error E distributed according to $\widehat{\mathcal{E}}$ has X and Z components that contain $\mathbf{x}$ and $\mathbf{z}$ respectively. For brevity, we have used $\mathbf{x}' \supseteq \mathbf{x}$ and $\mathbf{z}' \supseteq \mathbf{z}$ to mean that the supports of $\mathbf{x}$, $\mathbf{z}$ are contained in $\mathbf{x}'$, $\mathbf{z}'$ respectively. Treating $\mathcal{X}$ and $\mathcal{Z}$ separately in this way does not prevent correlations between X and Z errors.

Definition 2.3 (Locally decaying errors on qubits)**.** *Given an n-qubit state with Pauli errors distributed according to $\widehat{\mathcal{E}}$. We say that errors are described by a* locally decaying errors model *to mean that $\mathcal{X}$ and $\mathcal{Z}$ are both locally decaying distributions with failure rate p.*

We want to extend this idea to describe errors caused by faulty circuits. A *location* in a circuit C refers to a one- or two-qubit gate (including identity), single-qubit preparation or single-qubit measurement operation at some time step $1 \leq t \leq \mathcal{T}(C)$. A fault location is a location which performs a random Pauli operation following the desired Clifford operation. We assume that a fault location introduces a Pauli operator on the qubits in its support chosen according to some distribution $\widehat{\mathcal{F}}$. Given a set F of fault locations in C, the support of F is the set $\mathrm{supp}(F) \subseteq [\mathcal{W}(C)]$ of qubits that are in some location in F.

For a set F' of locations, let $\widehat{\mathcal{F}}(F')$ denote the probability of the set of locations F' being faulty. For a set F of fault locations, the total probability $\mathcal{F}(F)$ is

$$\mathcal{F}(F) = \sum_{F' \supseteq F} \widehat{\mathcal{F}}(F') \ . \tag{5}$$

Definition 2.4 (Locally decaying faults on circuits)**.** *Let C be a depth 1 circuit with faults distributed according to $\widehat{\mathcal{F}}$. We say that the faults are described by a* locally decaying faults model *if $\mathcal{F}$ is a locally decaying distribution with failure rate p_{phys}—for all sets of locations F,*

$$\mathcal{F}(F) \leq p_{\mathrm{phys}}^{|F|} \ .$$

Note that the probability of failure falls with the number of locations $|F|$ and not the number of qubits $|\mathrm{supp}(F)|$.

In practice, different locations may have different failure rates. To prove the existence of a threshold, we assume that p_{phys} is the maximum failure probability across all gates. We return to this assumption in Section 6, where we discuss how the logical failure rate behaves if gates have different failure rates.

Definition 2.4 pertains to circuits of depth 1—we assume faults in successive time steps are independent. In a more general model for faults, we could include arbitrary fault patterns for a circuit of growing depth so long as the probability of a particular fault path falls exponentially with the size of the fault path.

As a state undergoes circuit operations, errors can spread and accumulate. Consider a CNOT gate acting on two qubits. Figure 5 illustrates how a generating set of 2-qubit Pauli operators $\{\mathsf{XI}, \mathsf{IZ}, \mathsf{IX}, \mathsf{ZI}\}$ on these two qubits evolve under ideal CNOT. The error doubles in size in the worst-case scenario. As shorthand, we say that Pauli operators 'flow' within circuits to refer to this spreading. X operators flow down a CNOT and Z operators flow up.

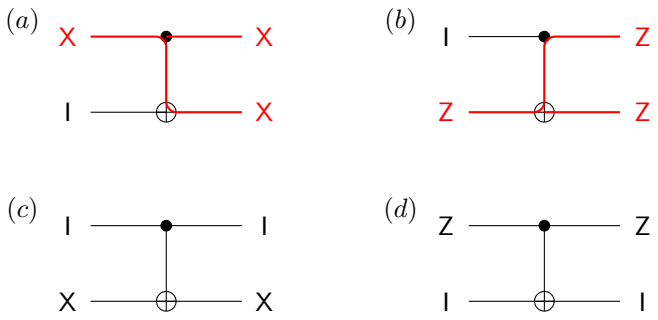

Figure 5: Evolution of Pauli errors under the action of CNOT. The first qubit is the control qubit and the second qubit is the target. The operators $\mathsf{X} \otimes \mathsf{I}$ and $\mathsf{I} \otimes \mathsf{Z}$ double in size. The red paths show how X 'flows down' a CNOT gate and Z 'flows up' a CNOT gate.

The structure of syndrome-extraction circuits is special. For $\mathsf{P} \in \{\mathsf{X}, \mathsf{Z}\}$, a controlled-P gate within the syndrome-extraction circuit uses ancilla qubits as control qubits and data qubits as target qubits (See Figure 4). This means that errors only flow in limited ways—for example, X errors always flow from ancilla qubits to data qubits, and Z errors flow from data qubits to ancilla qubits when CNOT gates are applied.

Implementing the imperfect circuit C, we obtain an imperfect syndrome. To overcome this problem, we repeat the syndrome-measurement circuit for r *rounds*. Let $\boldsymbol{\sigma} = (\boldsymbol{\sigma}_{\mathsf{X}}^{(1)}, \boldsymbol{\sigma}_{\mathsf{Z}}^{(1)}, ..., \boldsymbol{\sigma}_{\mathsf{X}}^{(r)}, \boldsymbol{\sigma}_{\mathsf{Z}}^{(r)})$ be the r faulty syndromes.

Failure rate per round:

Consider a corrupted code state $\mathsf{E}\,|\psi\rangle$ where ψ is a code state and $\mathsf{E} = \mathsf{X}(\mathbf{e}_x)\mathsf{Z}(\mathbf{e}_z)$ is some Pauli operator. If the syndrome-extraction circuit C has no faults, the joint state of the data and ancilla qubits after one round of syndrome extraction is described by

$$\mathsf{E}\,|\psi\rangle \otimes \mathsf{Z}(\boldsymbol{\sigma})\,|+\rangle^{\otimes m} \ , \tag{6}$$

where $\boldsymbol{\sigma}$ represent the ideal syndromes for X- and Z-type stabilizer generators.

However, because of faults in the circuit, the state after the circuit is

$$(\mathsf{D} \otimes \mathsf{A})(\mathsf{E}\,|\psi\rangle \otimes \mathsf{Z}(\boldsymbol{\sigma})\,|+\rangle^{\otimes m}) \ , \tag{7}$$

where D and A represent errors on the data and ancilla qubits respectively caused by faults in C that then spread.

Let $\widehat{\mathcal{E}}'(\mathsf{D} \otimes \mathsf{A})$ denote the probability of errors *per round* on the qubits. Let $\mathcal{X}'$, $\mathcal{Z}'$ denote the induced distributions for errors on data and ancilla qubits of X and Z type respectively.

Definition 2.5 (Probability of errors per round)**.** *We say that the probability of errors per round is locally decaying with failure rate* $p_{\text{round}} \in [0,1]$ *such that* $\mathcal{X}'$*,* $\mathcal{Z}'$ *are locally decaying distributions with failure rates* p_{round} *respectively.*

Definition 2.5 thus considers one round of syndrome extraction not as individual operations, but in aggregate; it then associates a failure probability p_{round} with the entire round associated with the probability of witnessing X and Z errors. Thus, p_{round} can be a function of the code size N, as well as other details of the implementation such as the specific syndrome-extraction circuit used.

A priori, $\widehat{\mathcal{E}}'$ can depend on r and the input error E. However, as entangling gates restrict the direction of error propagation, errors do not propagate from one data qubit to another or from one ancilla qubit to another. In Section 5.1, we use this to show that p_{round} does not depend on how many prior rounds of the syndrome-extraction circuit have already been applied. We show that p_{round} is a function of p_{phys} of the form $a \cdot p_{\text{phys}}^b$, where a is a function of the depth $\mathcal{T}(C)$ and b is a function of the degrees Δ_q and Δ_g.

Recovering the state: After performing r rounds of syndrome extraction, a *decoding algorithm* $\texttt{dec} : (\mathbb{F}_2^m)^{\times r} \to \mathcal{P}_n$ maps the observed syndrome $\boldsymbol{\sigma}$ to a deduced error.

The applied correction may not completely correct all errors due to faults in the syndrome extraction circuit. We declare success if, after applying the correction, the final state is 'not too far' from the desired output of the ideal circuit C. To this end, we consider the ideal recovery map $\mathcal{R}$ [AGP06]—a fictitious quantum channel that is not subject to geometric constraints or noise. We gauge the accuracy of the circuit $\widetilde{C}$ using the logical failure probability $p_{\mathcal{Q}}$, which is the probability that the residual error is correctable by the ideal recovery map. To be precise, $p_{\mathcal{Q}}$ is the probability that *any* logical qubit fails in one round of error correction. The probability $p_{\mathcal{Q}}$ also referred to as the *Word Error Rate* (WER).

Ideal recovery map & Thresholds: To understand whether a scheme is scalable, we are interested in properties of a *family* of codes $\{\mathcal{Q}_n\}$ to process an ever increasing number n of qubits. Consider a code family $\{\mathcal{Q}_n\}$ and suppose errors are described by a locally decaying distribution $\mathcal{E}$ with failure rate p_{in}. Let $\{C_n\}$ be the corresponding set of syndrome-extraction circuits to $\{\mathcal{Q}_n\}$, where faults are described by $\mathcal{F}$, a locally decaying distribution with failure rate $p_{\text{phys}} \in [0,1]$. We can compute p_{round} as a function of p_{phys} as shown in Section 5.1.

For our purposes, we say that the family has a *threshold* with respect to the noise model and decoding algorithm if there exists a pair q_{in}, $q_{\text{round}} \in (0,1]$ such that if

$$p_{\text{in}} < q_{\text{in}} \ , \qquad p_{\text{round}} < q_{\text{round}} \ , \tag{8}$$

the probability of logical failure $p_{\mathcal{Q}} \to 0$ as the size of the code $n \to \infty$. The logical probability of failure is defined with respect to family of ideal recovery maps. It depends on p_{in} and p_{phys} and the thresholds.

Whether a threshold exists with respect to a given noise model, the exact value of the threshold, as well as how quickly the logical failure probability decreases as a function of n (e.g. polynomially or exponentially), depend not only on the choice of quantum error-correcting code $\mathcal{Q}_n$, but also the implementation of the syndrome-extraction circuit C_n and the decoding algorithm. In our construction, the code family is a concatenated code where the syndrome-extraction circuit is subject to constraints on geometric locality.

While the state after error correction is 'close enough' to the codespace, undoing the deduced error may not correct all errors. The remaining errors on the state are described by $\mathcal{E}_{\text{res}}$ that is a locally decaying distribution with failure rate p_{res}. We can perform another round of error correction and thereby keep the state alive for arbitrary duration if $p_{\text{res}} < p_{\text{in}}$. For this reason, we will specify the residual failure rate after error correction in addition to the logical failure probability $p_{\mathcal{Q}}$.

2.3 Concatenated codes

A concatenated code is a quantum code obtained via the composition of two codes, an inner code $\mathcal{Q}_0$ and an outer code $\mathcal{Q}$. We consider the simple case of a $[\![n_0, 1, d_0]\!]$ code $\mathcal{Q}_0$ that only encodes 1 qubit and a suitable $[\![n, k, d]\!]$ code $\mathcal{Q}$.

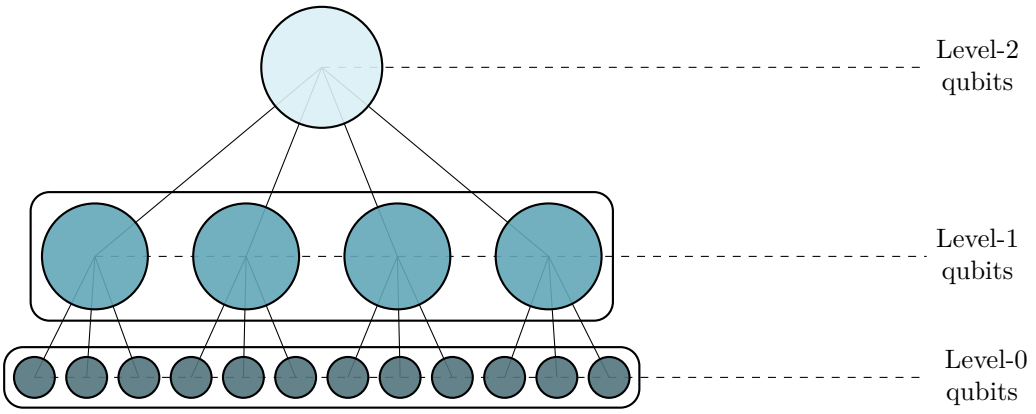

Figure 6: Visualizing a concatenated code $\mathcal{H}$.

Code parameters: The concatenated code, denoted $\mathcal{H}$ with parameters $[\![N, K, D]\!]$, is constructed by replacing each qubit of the code $\mathcal{Q}$ by a copy of $\mathcal{Q}_0$, resulting in n copies of the inner code $\mathcal{Q}_0$. The benefit of this construction is that the distance D of the code $\mathcal{H}$ is amplified with respect to the constituent codes. To be precise,

$$N = n \cdot n_0 \ , \qquad K = k \ , \qquad D = d \cdot d_0 \ .$$

The physical qubits are referred to as Level-0 qubits, the logical qubits of $\mathcal{Q}_0$ which form the block $\mathcal{Q}$ are referred to as Level-1 qubits, and the logical qubits of $\mathcal{H}$ are referred to as Level-2 qubits. See the schematic in Figure 6.

When errors on qubits are distributed in an i.i.d. manner, the advantage of concatenation becomes apparent when we "coarse grain" details of the concatenated code. Consider a simple setting where qubits are subject to independent X and Z errors. Suppose we use the code $\mathcal{Q}$ without concatenation. By assumption, the probability of failure of each of the physical qubits is p. However, after concatenation, the probability of failure of the Level-1 qubits is suppressed—it fails with probability proportional to $p^{d_0/2}$. This is because at least $d_0/2$ errors are required to cause a logical error for $\mathcal{Q}_0$. The inner code thus adds an extra layer of protection and consequently, the logical failure rate for the outer code is that much lower. As we shall see, we have to be more careful when making this sort of argument in the context of circuits.

Syndrome-extraction circuit: Let $C^{\mathcal{Q}_0}$ and $C^{\mathcal{Q}}$ denote the syndrome-extraction circuits for $\mathcal{Q}_0$ and $\mathcal{Q}$ respectively such that both can be implemented in 2 dimensions using $\mathcal{K}$, the set of local Clifford operations and R-local SWAP gates. To implement a CNOT between distant qubits, we may need to permute qubits using SWAP gates to bring them within range of a two-qubit gate. We discuss to how to design such a permutation in Section 3.

A syndrome-extraction circuit $C^{\mathcal{H}}$ for the concatenated code $\mathcal{H}$ can be expressed in terms of the syndrome-extraction circuits $C^{\mathcal{Q}_0}$ and $C^{\mathcal{Q}}$. Each data and ancilla qubit in the syndrome-extraction circuit for $C^{\mathcal{Q}}$ is now replaced with a copy of $\mathcal{Q}_0$. Each gate in $C^{\mathcal{Q}}$ is replaced by the corresponding logical Clifford gate between Level-1 qubits. Thus, even for constructing a quantum memory, we need to understand how to perform a restricted set of inner code logical operations in a fault-tolerant manner. We perform error correction either after the logical gate or in an interleaved manner. We discuss this in the context of our explicit architecture in Section 4.

The ideal recovery map $\mathcal{R}_{\mathcal{H}}$ for $\mathcal{H}$ is obtained by first decoding the n copies of the inner code $\mathcal{Q}_0$ using $\mathcal{R}_0$ and then decoding the outer code using $\mathcal{R}_{\mathcal{Q}}$. Here $\mathcal{R}_{\mathcal{Q}_0}$ and $\mathcal{R}_{\mathcal{Q}}$ refer to the ideal recovery maps for $\mathcal{Q}_0$ and $\mathcal{Q}$ respectively. Thus $\mathcal{R}_{\mathcal{H}} = \mathcal{R}_{\mathcal{Q}} \circ (\mathcal{R}_{\mathcal{Q}_0})^{\otimes n}$.

We generalize the notion of location in the context of circuits. A Level-1 location refers to a Level-1 gate, including the error correction rounds. The location is faulty if it implements the incorrect logical operation on the Level-1 qubits in its support. In the context of $C^{\mathcal{Q}}$, a single Level-1 location in the circuit could refer to a SWAP gate or an entangling gate or a preparation or measurement of a logical qubit of $\mathcal{Q}_0$.

When "coarse graining" circuits for concatenated codes, more care is needed than the i.i.d. errors setting. We illustrate using the following examples.

Problem # 1: Level-1 failure rates are not additive

Consider a n_0-qubit code state of the inner code $\rho_{\text{in}}^{(0)}$ with Level-0 errors E_{in}. The error E_{in} is not catastrophic—the ideal decoder $\mathcal{R}_0$ can correct it. The state is therefore correctable.

Consider the syndrome-extraction circuit $C^{\mathcal{Q}_0}$ with Level-0 faulty locations F. Suppose there is some error supported on $\text{supp}(F)$ but that this error is not a logical error. We may then be tempted to extend the notion of correctability to include circuits and declare the circuit $C^{\mathcal{Q}_0}$ correctable. However, this is misleading as a correctable circuit acting on a correctable input state need not produce a correctable output state.

Let $\rho_{\text{out}}^{(0)}$ denote the output state and $\mathsf{E}_{\text{out}}^{(0)}$ denote the errors on this state. Suppose the faulty locations F result in an error E_F. The product $\mathsf{E}_{\text{in}} \cdot \mathsf{E}_F$ might not be correctable. In addition, the errors E_{in} and E_F can spread in unpredictable ways within the circuit. We therefore cannot calculate the Level-1 output failure probability by merely knowing the input state and the faults individually resulted in correctable errors. We need additional structure.

Problem # 2: Level-0 failure rate is not always sustainable

Secondly, the thresholds are decoder dependent. By definition, the ideal decoder $\mathcal{R}_0$ has no faults; if the errors on the state $\rho_{\text{out}}^{(0)}$ are correctable, then $\mathcal{R}_{\mathcal{Q}_0}$ is successful. On the other hand, $C^{\mathcal{Q}_0}$ can contain faults and may be unable to deal with as many errors as the ideal decoder $\mathcal{R}_{\mathcal{Q}_0}$. This can result in instances where the output state $\rho_{\text{out}}^{(0)}$ will be correctable by $\mathcal{R}_0$; by our criteria for success, the output state is decodable. However, the number of residual errors may be above the threshold for error correction. In other words, as error correction is itself faulty, these faults can combine with existing errors to cause a logical failure.

In our construction, we address these problems in Section 5.2. We shall show that for sufficiently low failure rates, we can indeed ignore dealing with the syndrome-extraction circuit for the outer code assuming a failure rate that depends on the inner code. This statement relies on the structure of LDPC codes and surface codes. We now proceed to review these codes.

2.4 Constant-rate LDPC codes

An $[\![n, k, d]\!]$ code family $\{\mathcal{Q}_n\}$ is said to be a low-density parity-check code if

1. each stabilizer generator S_i, $i \in [m]$, only acts non-trivially on at most a constant number Δ_g of qubits for all elements in $\{\mathcal{Q}_n\}$.
2. each qubit only participates in at most a constant number Δ_q of stabilizer generators for all elements in $\{\mathcal{Q}_n\}$.

To include the degree of stabilizer generators and qubits, we shall say that a code family $\{\mathcal{Q}_n\}$ is an $[\![n, k, d, \Delta_q, \Delta_g]\!]$ LDPC family.

We will choose the outer code to be a code with constant rate, i.e. $k = \Theta(n)$. Constructing a constant-rate LDPC code is a non-trivial task because there is a conflict between the constraints on stabilizer generators. On one hand, all stabilizer generators need to commute with each other

to form a well-defined stabilizer code; on the other hand, the stabilizer generators need to have weight at most Δ_g. Despite these difficulties, there exists constant-rate quantum LDPC codes, i.e. $k(n) = \Theta(n)$, with distance $d(n) = \Theta(n^\delta)$ for $0 < \delta \leq 1$.

LDPC codes have a threshold [KP13; Got13] *if* operations in $\mathcal{K}$ are not subject to any locality constraints. In this setting, we can construct syndrome-extraction circuits where each qubit is involved in a constant number of two-qubit gates.

Consider a family of $[\![n, k, d, \Delta_q, \Delta_g]\!]$ quantum LDPC codes $\{\mathcal{Q}_n\}$ where $k = \rho \cdot n$ for some constant $\rho > 0$ and distance $d = \Theta(n^\delta)$ for some $\delta > 0$. Suppose qubits are subject to the following errors:

1. the input state is subject to locally decaying errors with failure rate per qubit p_{in}.
2. the syndrome-extraction circuit is subject to locally decaying faults with failure rate per gate p_{phys}.

We restate a result from Gottesman [Got13] (Theorem 4) which guarantees the existence of a threshold for arbitrary LDPC codes. In this construction, we require $r = d(n)$ rounds of syndrome extraction. After syndrome extraction, the (imperfect) syndromes are processed by a minimum-weight decoder `dec`. We do not describe the decoder in detail here and merely note that it exists. For generic LDPC codes, the minimum-weight decoder is not necessarily efficient.

There exist q_{in}, q_{round} in the interval $(0, 1]$ such that when

$$p_{\text{in}} \leq q_{\text{in}} \ , \qquad p_{\text{round}} \leq q_{\text{round}} \ , \tag{9}$$

the following is true.

The minimum-weight decoder `dec` yields a correction such that:

1. the final state is recoverable by an ideal recovery operator $\mathcal{R}_{\mathcal{Q}}$ with probability at least to $1 - p_{\mathcal{Q}}(n)$ where $p_{\mathcal{Q}}(n) := \exp[-\Theta(d(n))]$. To be precise, $p_{\mathcal{Q}}(n)$ is the probability of failure *per round of syndrome extraction.*
2. the physical qubits have residual errors that are described by a locally decaying error model with failure rate at most p_{round}.

The first condition guarantees that the probability of logical failure falls exponentially with the distance of the code. It is worth noting that we declare a logical failure if *any* logical qubit fails. This is *qualitatively* different from codes that only encode a constant number of qubits.

The second condition on the residual error is not what is in the theorem statement of Theorem 4 of [Got13]; however, the proof implies it. For sufficiently low values of p_{phys}, it guarantees that we can continue to perform error correction for arbitrarily many rounds (conditioned on no logical errors). In other words, we require $p_{\text{round}} < p_{\text{in}}$.

We highlight that this result applies to arbitrary LDPC codes, i.e. it is independent of the rate of the code. In particular, it applies to the surface code.

We note that the threshold is stated in terms of p_{round}, and not directly in terms of p_{phys}. This is for two reasons: (1) this is how Theorem 4 of [Got13] is itself stated, and (2) in our construction, the dependence of p_{round} on p_{phys} can change depending on the depth of the syndrome-extraction circuit. Stating the thresholds in this manner will allow us to derive the functional dependence between p_{round} and the depth of the syndrome-extraction circuit. In Gottesman's construction [Got13], the syndrome-extraction circuit is constant depth and therefore p_{round} is also a constant. In contrast, our construction is more complicated because of constraints on geometric locality.

It is known that codes defined by geometrically-local stabilizer generators in 2 dimensions cannot achieve both constant rate and growing distance [BT09; BPT10]. To achieve a constant rate and distance $d = \Theta(n^\delta)$ with fixed degrees Δ_q and Δ_g, the amount of non-locality scales with the parameters k and d [BK22a; BK22b]. In other words, there exist $\Theta(n)$ stabilizer generators such that qubits in their support cannot be close to each other in the 2-dimensional lattice. In the

context of syndrome-extraction circuits, the result by Delfosse *et al.* [DBT21] states that the depth of the syndrome-extraction circuit will grow when we only have geometrically-local gates and a limited number of ancilla qubits. (Recall Equation (1).)

In Section 5, we show that p_{round} grows if the syndrome-extraction circuit C is constrained by geometric locality. In other words, it is *not* constant and we need an approach different than Gottesman's to prove the existence of a threshold. In our alternative approach using the hierarchical code, the growth of the inner code suppresses Level-1 logical errors sufficiently to ensure that the Level-2 logical failure rate drops rapidly as the outer LDPC code scales up.

Finally, we discuss the choice of quantum LDPC code. While the result above applies to generic quantum LDPC codes, more is known about specific constructions. Quantum expander codes are one family of constant-rate quantum LDPC codes for which $d = \Theta(\sqrt{n})$ [TZ14; LTZ15]. It has been rigorously proven that these codes can be equipped with an *efficient* decoder called `small-set-flip` [FGL18a; FGL18b]. Furthermore, it was shown that the decoder is *single shot* meaning that it only requires a constant number of rounds of syndrome measurements for the decoder to function even when the syndrome is noisy. Similar to Gottesman's requirements for the existence of a threshold, all that is needed in Fawzi *et al.* [FGL18a] is for p_{round} to remain constant. However, unlike Gottesman's construction, it was shown that these codes have an efficient decoding algorithm that only require a constant number of rounds of syndrome measurement. Thus, if we wish to implement a quantum expander code, we can use the same machinery presented in this paper to justify an efficient single-shot decoder for the outer code.

We refer to LDPC codes with distance scaling as $d = \Theta(n)$ as good codes. For nearly 2 decades, it was unclear whether good codes even existed. Following a series of breakthroughs, [PK20; EKZ20; KT21; HHO21; BE21], this impasse was famously crossed first by Panteleev & Kalachev [PK22] and later by Leverrier & Zémor [LZ22]. Furthermore, these codes have the single-shot property (albeit with an inefficient decoder) as guaranteed by Quintavalle *et al.* [QVRC21].

In this paper, we do not place any constraints on the outer LDPC code other than it have constant rate $\rho > 0$ and distance $d = \Theta(n^\delta)$ for $1/2 \leq \delta \leq 1$. Our construction works for all δ, but we choose $\delta \geq 1/2$ to simplify some theorem statements.

2.5 Surface codes

We consider the rotated surface code [BK98; HFDM12], arguably the simplest code that can be laid out on a 2-dimensional lattice. The surface code is an LDPC code, albeit with vanishing asymptotic rate.

An example is shown in Figure 7. The code is implemented on a *rotated* lattice, i.e. the points of the lattice correspond to the vertices of squares that run in 45 degree angles relative to the x and y axes. The points of the lattice are labeled (a, b) where $a, b \in \mathbb{Z}/2$. Each black dot represents a data qubit; these are located on integer points, i.e. on points (a, b) where $(a, b) \in \mathbb{Z}$. Each colored dot represents a syndrome qubit; these are located on half-integer points, i.e. on points $(a + 1/2, b + 1/2)$ where $(a, b) \in \mathbb{Z}$. Corresponding to each blue face, we define an X-type stabilizer generator that jointly measures $\mathsf{X}^{\otimes 4}$ on adjacent data qubits. Similarly, corresponding to each red face, we define a Z-type stabilizer generator that jointly measures $\mathsf{Z}^{\otimes 4}$ on adjacent qubits. The semi-circles represent stabilizers that only act on two qubits in their support, i.e. they measure $\mathsf{X}^{\otimes 2}$ or $\mathsf{Z}^{\otimes 2}$ jointly.

The rotated surface code $\mathcal{RS}_\ell$ encodes exactly one qubit and has distance d_ℓ. It uses d_ℓ^2 data qubits and $d_\ell^2 - 1$ syndrome qubits. The total number of qubits is thus $\ell^2 = 2d_\ell^2 - 1$. We use ℓ to parameterize the code family. We also refer to each code as a tile.

Using operations $\mathcal{K}$, the syndrome-extraction circuit for the surface code has depth 6.

Thresholds for error correction: To motivate our noise model, we consider a simple setting where $n = 1$, i.e. we have a single tile $\mathcal{RS}_\ell$. Suppose we are given physical qubits, each qubit in some fixed computational-basis state, and use the syndrome-measurement circuit to *project* this state

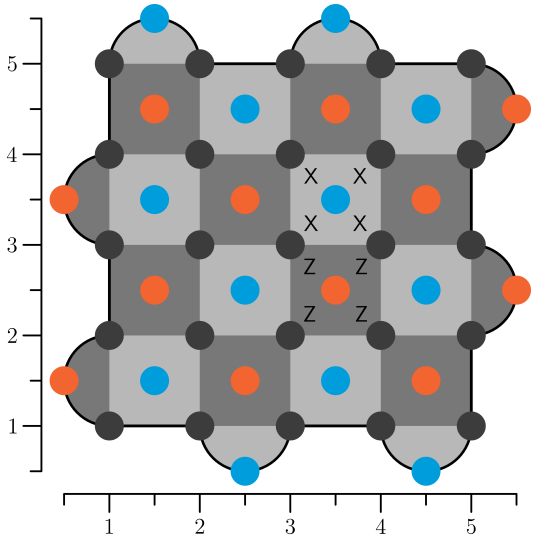

Figure 7: A surface code of distance $d_\ell = 5$. Each dark gray dot represents a data qubit. Light faces correspond to X checks and dark faces correspond to Z checks. They are measured using the qubit represented as a blue or orange dot respectively in the center of each face. Note that data qubits reside at integer points $\mathbb{Z}^2$ and ancilla qubits reside at the points of this lattice shifted by $(1/2, 1/2)$.

onto a (fixed) code state of the surface code. If the physical qubits are subject to locally decaying errors at failure rate $p^{(0)}_{\text{phys}}$, we can derive the Level-1 probability of failure $p^{(1)}_{\mathcal{RS}}(\ell)$ for the surface code. The superscripts denote the noise on Level-1 and Level-0 qubits respectively.

Contrast this with the scenario where we obtain the surface code from another party. Upon receipt, we are only informed that the tile has already failed with failure rate $p^{(1)}_{\text{in}}$; we do not have additional information, such as syndrome histories from prior rounds of error correction. If the code has not *already* failed, then we are guaranteed that the physical failure rate is $p^{(0)}_{\text{phys}}$. Failure after error correction can thus result in two ways: either the tile fails prior to us receiving the state with probability $p^{(1)}_{\text{in}}$ *or* conditioned on it being correct, it fails because of error correction with probability $p^{(1)}_{\mathcal{RS}}(\ell) = \exp(-c_{\text{EC}} \cdot \ell)$ for some positive number c_{EC} that does not depend on ℓ. The Level-1 failure rate after error correction is thus $p^{(1)}_{\text{in}} + p^{(1)}_{\mathcal{RS}}(\ell)$.

When performing *repeated* rounds of error correction, we require the Level-1 failure rate $p^{(1)}_{\text{in}}$ to bound the probability that the code has already failed in prior rounds.

Surface codes will form the inner code in our concatenated construction. Consider the syndrome-extraction circuit C for the constant-rate LDPC code $\mathcal{Q}_n$. Suppose $\mathcal{W} = \mathcal{W}(C)$ is the width of the circuit. We require an arrangement of $\mathcal{RS}_\ell^{\otimes \mathcal{W}}$ in two dimensions. In Section 4, we introduce a *bilayer* architecture for arranging tiles in two parallel layers. We return to the explicit description of this layout in Section 4.

Consider an input state of the code $\mathcal{RS}_\ell^{\otimes \mathcal{W}}$. The errors are distributed in the following manner:

1. Level-1 errors are described by $\mathcal{E}^{(1)}$, a locally decaying distribution with failure rate $p^{(1)}_{\text{in}}$.
2. Level-0 errors are described by $\mathcal{E}^{(0)}$, a locally decaying distribution with failure rate $p^{(0)}_{\text{in}}$.

Faults in the syndrome-extraction circuit are described by $\mathcal{F}^{(0)}$, a Level-0 locally decaying distribution with failure rate $p^{(0)}_{\text{phys}}$.

The code $\mathcal{RS}_\ell^{\otimes \mathcal{W}}$ is itself an LDPC code (with vanishing rate asymptotically), and therefore, we can apply Theorem 4 from [Got13]. We note that although the original theorem is itself is not stated in this way, the proof implies the following.

There exist thresholds $q_{\text{in}}^{(0)}$, $q_{\text{round}}^{(0)}$ on Level-0 failure rates such that, below threshold, the probability of logical failure after error correction is described by a locally decaying Level-1 error $p_{\text{in}}^{(1)} + p_{\mathcal{RS}}^{(1)}(\ell)$ where $p_{\mathcal{RS}}^{(1)}(\ell) = \exp(-c_{\text{EC}} \cdot \ell)$ for some positive number c_{EC} that is independent of ℓ.

In addition, the state after error correction is described by locally decaying errors with failure rate proportional to $p_{\text{round}}^{(0)}$. This guarantees that if we are sufficiently below threshold, then the number of residual errors is low enough such that we can apply another round of error correction.

Unlike the case for general LDPC codes, surface codes possess a minimum weight decoder that runs in poly(n) time by mapping the decoding problem to a minimum-weight perfect matching problem.

Logical Clifford operations: As highlighted in the subsection on concatenated codes, we need to implement logical Clifford operations for the surface code to be able to use it within a concatenated construction. Extending our notation from Definition 2.1, we let $\mathcal{K}_0$ denote the physical geometrically-local Clifford gates and R-local SWAP operation on the physical qubits. Let C_0 be the syndrome-extraction circuit for $\mathcal{RS}_\ell$ constructed using $\mathcal{K}_0$.

Let $\mathcal{K}_1$ denote the corresponding logical operations on the surface code. Single-tile operations in $\mathcal{K}_1$ — state preparation in a fixed stabilizer state, (destructive) measurement of logical Pauli operators and applying Pauli corrections — can be performed using operations in $\mathcal{K}_0$ regardless of how two-tile gates are implemented. The only Clifford operations we require are two-tile operations: CNOT and R-local SWAP. These are discussed in Section 4.

3 Permutation routings on sparse graphs in two dimensions

In this section, we prove Theorem 1.1, restated here for convenience.

Theorem. *For R even, there is an efficient construction of a degree-12 graph $G = (V, E)$ whose vertex set V is identified with an $L \times L$ lattice with edges of length at most R. Any permutation $\alpha : V \to V$ can be performed in depth $3L/R + O(\log^2 R)$.*

We use this result in the next section to construct syndrome-extraction circuits for quantum LDPC codes. We shall study permutation routings on graphs and focus on $\text{NN}_2(L, R)$, the $L \times L$ lattice in 2 dimensions where two vertices share an edge if they are separated by a distance of at most R. Based on the idea of a permutation routing on product graphs, we demonstrate that we can implement an arbitrary permutation in depth $O(L/R)$. For the special case of the 2D lattice, we can make heavy use of sorting networks to find implementations of target permutations.

Using sorting networks to implement long-range connectivity is itself not a new idea [Bea+13]. For instance, it was used in Delfosse *et al.* [DBT21] to construct syndrome-extraction circuits for quantum expander codes to match the bound in Equation (1). The results in this section generalize this idea to arbitrary syndrome-extraction circuits with constant spatial overhead. To the best of our knowledge, this is the first work to construct sparse syndrome-extraction circuits when R can scale as a function of L.

3.1 Permutation Routing on product graphs

A *permutation routing* is sometimes explained in terms of a pebble-exchange game, where pebbles are placed on the vertices of an (undirected) connected graph $G = (V, E)$. The pebble on vertex $u \in V$ has an address $\alpha(u)$. The addresses of all the pebbles together specify a permutation α on the vertices of G. We are allowed to swap any two pebbles along an edge of G. Formally, every vertex has a label and for every edge $(u, v) =: e \in E$, we are equipped with an edge permutation $\pi(e)$ that exchanges the labels of u and v. Edge permutations can be performed in parallel as long as every pebble is involved in at most one edge permutation in one time step. We say β is a simple permutation on G if it is the product of edge permutations $\{\pi(e)\}_e$ that commute. The

objective of the pebble-exchange game is to find a minimum sequence of simple permutations so that the pebble that began at u is located on the vertex $\alpha(u)$ afterwards. In other words, we wish to find the smallest sequence of simple permutations $\beta_1, ..., \beta_{\mathcal{T}(\alpha)}$ such that $\alpha = \beta_{\mathcal{T}(\alpha)} \circ \cdots \circ \beta_1$. Here $\mathcal{T}(\alpha)$ denotes the minimum number of simple permutations required to perform α. We represent permutations using the one-line notation [Wik22] where $\alpha = \begin{pmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_n \end{pmatrix}$ means 1 is mapped to α_1, 2 is mapped to α_2 and so on. Given any permutation α, the permutation α^{-1} can be computed efficiently by applying the permutation to a list of consecutive integers $[n]$.

The R-nearest-neighbor graph: The R-nearest-neighbor graph in 1 dimension of length L is denoted $\mathrm{NN}_1(L, R) = (V, E)$ where

$$V = \{1, ..., L\}, \qquad E = \{(u, v) : |u - v|_2 \leq R\} \ , \tag{10}$$

where $|\cdot|_2$ represents the standard 2-norm. This is the graph for which the vertices are simply the positive integers up to L and two vertices are connected by an edge if their difference is less than R. In particular, consider the graph $\mathrm{NN}_1(L, 1)$ which corresponds to the path graph.

Fact: We can perform an arbitrary permutation α of pebbles placed on the vertices of the path graph $\mathrm{NN}_1(L, 1)$ in depth $L-1$ [Knu97]. The explicit permutation routing algorithm `Path-Routing` is presented in Algorithm 1.

Algorithm 1 `Path-Routing`(α)

Input: Permutation α.
Output: simple permutations $\beta_1, ..., \beta_{L-1}$ such that $\alpha = \beta_{L-1} \circ \cdots \circ \beta_1$.

`labels` $\leftarrow \{\alpha^{-1}(1), ..., \alpha^{-1}(L)\}$
$t = 1$.
while $t \leq L - 1$ **do**
  $\beta_t \leftarrow \{1, \ldots, L\}$
  for $i \in \{1, ..., \lfloor L/2 \rfloor\}$ **do**
    $a \leftarrow 2i - 1$ if t is even else $2i$
    $b \leftarrow 2i$ if t is even else $2i + 1$
    if `label`$(a) <$ `label`(b) **then** ▷ *Swap*
      $\beta_t(a) \leftarrow b$
      $\beta_t(b) \leftarrow a$
      Exchange `label`(a) and `label`(b).
  $t \leftarrow t + 1$.
return $\beta_1, ..., \beta_{L-1}$.

To illustrate, we consider a permutation α on the path graph on 8 vertices in Figure 8. Here, $\alpha = \begin{pmatrix} 6 & 7 & 2 & 5 & 3 & 4 & 8 & 1 \end{pmatrix}$.

We can generalize this concept and define the R-nearest-neighbor graph in 2 dimensions which we denote by $\mathrm{NN}_2(L, R)$. It has vertices $\{\mathbf{u} : \mathbf{u} = (u_x, u_y) \in [L] \times [L]\}$; two vertices $\mathbf{u}$, $\mathbf{u}'$ share an edge if $|\mathbf{u} - \mathbf{u}'|_2 \leq R$. Our objective is to build up to a routing algorithm on this graph. Before considering this general case, we study the case where $R = 1$. The idea used there will be used again for general R in the next subsection.

Routing on graph products: The main idea we present in this subsection are techniques due to Annexstein and Baumslag [AB90] to route on Cartesian products of graphs. They showed that we can derive routing algorithms for the Cartesian product $G_1 \times G_2$ using routing algorithms for graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$.

The general routing algorithm `Product-Routing` presented in Algorithm 2 below applies to any two graphs G_1 and G_2 for which routing routines are known. Each $v_1 \in V_1$, defines a "row"

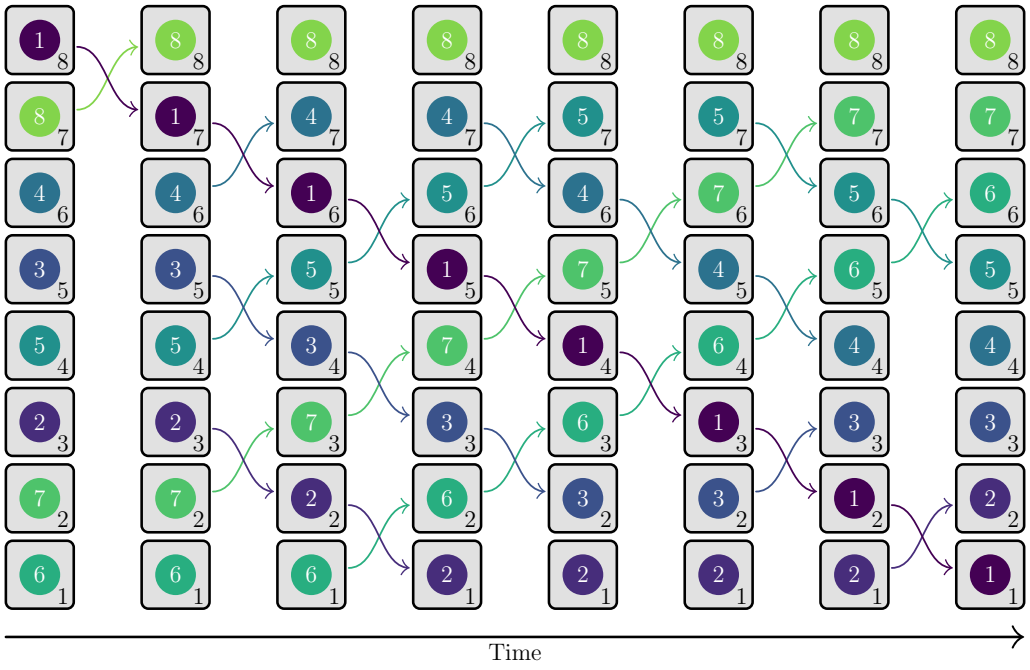

Figure 8: Example: implementing the permutation $\alpha = \begin{pmatrix} 6 & 7 & 2 & 5 & 3 & 4 & 8 & 1 \end{pmatrix}$ with nearest-neighbor swaps using Algorithm 1.

$\mathcal{R}_{v_1} = \{v_1\} \times V_2$, and each $u_2 \in V_2$ defines a "column" $\mathcal{C}_{u_2} = V_1 \times \{u_2\}$ [4]. We call a permutation α of the vertices of $G_1 \times G_2$ a *row permutation* if the permutation respects a decomposition into rows i.e. for all $v_1 \in V_1$, $\alpha : \mathcal{R}_{v_1} \to \mathcal{R}_{v_1}$. Likewise, for a *column permutation*, we have that for all $u_2 \in V_2$, $\alpha : \mathcal{C}_{u_2} \to \mathcal{C}_{u_2}$. A row or column permutation can be implemented using routing routines for G_2 or G_1 by applying the routine to each copy in the Cartesian product.

Lemma 3.1 (Annexstein & Baumslag [AB90])**.** *For any routing α_{12} on $G_1 \times G_2$, there exist column permutations α_1, α_1' and row permutations α_2 on G_2 such that: $\alpha_{12} = \alpha_1 \circ \alpha_2 \circ \alpha_1'$. These permutations can all be computed in polynomial time. If the permutations α_1 and α_1' require depth at most $\mathfrak{T}_1$ and α_2 requires depth at most $\mathfrak{T}_2$. Then α_{12} requires depth at most $2\mathfrak{T}_1 + \mathfrak{T}_2$.*

We provide some intuition for this lemma. At first glance, one might expect that a row permutation α_1 followed by a column permutation α_2 ought to suffice. However, this will not always work—if two pebbles in a row share the same destination column, then no row permutation will be able to send both the pebbles to the correct column.

To avoid collisions, we start the procedure with an additional step. We first send pebbles to rows in which no other pebbles shares the same destination column, so that the routing procedure performs a column permutation, a row permutation, and finally a column permutation. This problem will be rephrased as an edge-coloring problem where each color corresponds to the intermediate row that qubits will be routed through.

We first construct a bipartite multigraph B over the vertices $(V_2 \sqcup V_2')$ with left and right vertex sets both copies of V_2 [5]. To each pebble, we associate an edge between the initial column on the left and the destination column on the right. B is bipartite and has degree at most $|V_1|$, so there exists an efficiently computable edge coloring with $|V_1|$ colors [Sch+03] i.e. a decomposition into $|\mathcal{R}|$ disjoint matchings.

To each color, $\tau \in [V_1]$, we will assign an arbitrary row. For each pebble (edge), we will first pre-route it to the assigned row (color) before completing a final routing along the rows then columns.

[4] This notation is inspired by thinking about the vertices arranged as a matrix of size $V_1 \times V_2$.

[5] A multigraph is a generalization of a graph where two vertices are allowed to share multiple edges

In a valid coloring, no two edges (pebbles) of the same color (intermediate row) are incident to the same vertex (column). In the first step, this means that, for every column, each pebble has a unique intermediate destination row. Further, in the row permutation step, for every row, each pebble has a unique destination column. Finally, in the last column permutation step, each pebble is in its destination column, so, for each column, each pebble has a unique destination row.

We assume we are given blackbox access to an efficient edge coloring algorithm for bipartite graphs [Sch+03] and call it via a subroutine `Edge-Coloring` in Algorithm 2.

Algorithm 2 `Product-Routing`(α)

Input: Permutation $\alpha\colon V_1 \times V_2 \to V_1 \times V_2$
Output: Row permutations α_1, α_1' and a column permutation α_2 such that $\alpha = \alpha_1 \circ \alpha_2 \circ \alpha_1'$.

Initialize bipartite graph $B \leftarrow (V_2 \sqcup V_2', \varnothing)$ with no edges.
for Every $(v_1, u_2) \in V_1 \times V_2$ **do**
    Draw an edge between $u_2 \in V_2$ and $u_2' \in V_2'$ if $\alpha(v_1, u_2) = (v_1', u_2')$.
$\tau \leftarrow$ `Edge-Coloring`(B)
for $(v_1, u_2) \in E$ **do**
    $\alpha_1'(v_1, u_2) \leftarrow (\tau(e), u_2)$
    $\alpha_2(\tau(e), u_2) \leftarrow (\tau(e), u_2')$
    $\alpha_1(\tau(e), u_2') \leftarrow (v_1', u_2')$

To illustrate Algorithm 2, we describe how to obtain the permutation routing on the nearest-neighbor graph in 2-dimensions $\mathrm{NN}_2(L, 1)$. Noting that $\mathrm{NN}_2(L, 1) \cong \mathrm{NN}_1(L, 1) \times \mathrm{NN}_1(L, 1)$, Lemma 3.1 implies that an arbitrary permutation on $\mathrm{NN}_2(L, 1)$ can be implemented using a product of permutations on the components $\mathrm{NN}_1(L, 1)$. Recalling that any permutation on the path graph $\mathrm{NN}_1(L, 1)$ can be done in depth $L - 1$ implies the following corollary.

Corollary 3.2. *Any permutation α on $\mathrm{NN}_2(L, 1)$ can be performed in depth $3L - 3$.*

Proof. $\mathrm{NN}_2(L, 1) \cong \mathrm{NN}_1(L, 1) \times \mathrm{NN}_1(L, 1)$, and we can route on $\mathrm{NN}_1(L, 1)$ using an even-odd sorting network (Algorithm 1). Using Algorithm 2, we have that the total number of steps is $3(L - 1)$. ■

In a different context, the above claim was also made Thompson & Kung [TK77].

Figure 9 shows an example of a permutation of such a lattice using Algorithm 2.

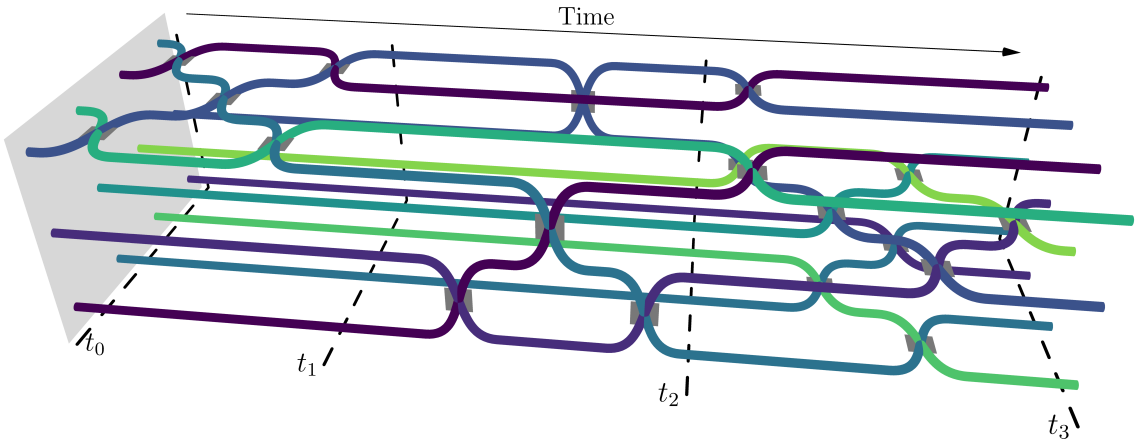

Figure 9: Visualizing the routing algorithm via the space-time path of individual qubits. For each swap location, a gray rectangle indicated the plane of the swap is drawn for visualization purposes. Note that in the intervals (t_0, t_1) and (t_2, t_3) there are only row swaps, and in the interval (t_1, t_2) there are only column swaps. Each of the swaps within a single row or column we obtain by a 1D even-odd sorting network (Figure 8).

3.2 Permutation routing given long-range gates

In this subsection, we show how to route on $\mathrm{NN}_2(L, R)$. We do this by finding graph approximations – subgraphs of our original graph that we can route on nearly as well. We will approximate $\mathrm{NN}_2(L, R)$ using a two-step approach—first we show that the complete graph times a 2-dimensional nearest-neighbor graph approximates $\mathrm{NN}_2(L, R)$ well; we then show that a sparse graph approximates the complete graph well. Together, this will result in a circuit with sparse connectivity that exploits long-range connectivity of range R.

The complete graph & sparse approximations: The complete graph K_m is a graph on m vertices with edges between every pair of vertices. Any permutation α on K_m can trivially be accomplished in depth 2. Using a complete graph will simplify some of the analysis in this section. However, K_m is a dense graph; in turn, the corresponding syndrome-extraction circuit we construct from it will require that qubits are involved in a super constant number of two-qubit gates. To avoid this problem, we replace K_m by a sparse graph in exchange for a modest increasing in the depth of permutations. We state some facts about sparse approximations to the complete graph K_m.

Definition 3.3 (Spectral Expander)**.** *Let G be a d-regular graph on m vertices where all eigenvalues of the adjacency matrix except for the largest $\{\lambda_i\}_{i=2}^m$ satisfy $|\lambda_i| \leq \lambda < d$. G is said to be an (m, d, λ)*-spectral expander*.*

Fact 3.4 ([Fri03])**.** *For even $d \geq 4$ and any $\varepsilon > 0$, there is an efficient randomized*[6] *algorithm that returns a random d-regular graph such that it is an (m, d, λ)-spectral expander for $\lambda = 2\sqrt{d-1}+\varepsilon$.*

This fact establishes that a random regular graph is a good spectral expander with high probability. For $d = 4$, such a graph can be defined on any even number of vertices, so this family is extremely flexible. For convenience, we will set $d = 4$ and m even. We will call a random 4-regular graph picked in this way on m vertices $\mathcal{E}_m$.

The next fact concerns routing on spectral expanders in an efficiently computable manner.

Fact 3.5 ([ACG94])**.** *Let G be an (m, d, λ)-spectral expander. Then, any permutation $\sigma : [m] \to [m]$ can be performed in depth $O\left(\frac{d^2}{(d-\lambda)^2}\log^2(m)\right)$.*

Take together, we can replace K_m by a random 4-regular subgraph $\mathcal{E}_m$ on which we can route in depth $O(\log^2(m))$. For our purposes, we assume that the routing algorithm for these sparse graphs can be accessed in a black-box manner.

Now we are prepared to move on to implementing permutations on $\mathrm{NN}_2(L, R)$. The depth we will find is nearly optimal, even when $R > 1$. For convenience, let us assume that L/R is an integer. First, note that at distances shorter than R, $\mathrm{NN}_2(L, R)$ locally "looks" like a complete graph on R-vertices. We can leverage this to find a spanning subgraph[7] of $\mathrm{NN}_2(L, R)$ that is a product of graphs that we know how to route on: K_R and $\mathrm{NN}_2(L/R, 1)$.

Lemma 3.6. *If R divides L, $\mathrm{NN}_1(L/R, 1) \times K_R$ is a spanning subgraph of $\mathrm{NN}_1(L, R)$.*

Proof. Using the coordinates $[L/R] \times [R]$ for $\mathrm{NN}_1(L/R, 1) \times K_R$ and $[L]$ for $\mathrm{NN}_1(L, R)$, we can map the vertices of $\mathrm{NN}_1(L/R, 1) \times K_R$ to those of $\mathrm{NN}_1(L, R)$ using the bijection $\eta\colon [L/R] \times [R] \to [L]$

$$(a, b) \xrightarrow{\eta} (a-1)R + b \ .$$

Away from the boundary, the neighbors of an arbitrary vertex (a, b) of $\mathrm{NN}_1(L/R, 1) \times K_R$ are $(a \pm 1, b)$ and $(a, [R] \setminus \{b\})$. A vertex (a, b) at a boundary is adjacent to the vertices $(a, [R] \setminus \{b\})$

[6] We note that the result in [Fri03] only shows that a regular random graph will be an expander with high probability. However, spectral expansion is efficiently checkable, so this process may be repeated until success.

[7] A subgraph H of a graph G is said to be a spanning subgraph if all vertices of G are contained in H.

and one of $(a-1,b)$ or $(a+1,b)$; whichever is in the graph. All elements of these sets are at most a distance R from (a,b) under η, so it is a valid edge in $\mathrm{NN}_1(L,R)$. ■

Clearly, $\mathrm{NN}_1(L,R)\times\mathrm{NN}_1(L,R)$ is a spanning subgraph of $\mathrm{NN}_2(L,R)$ given by retaining only those edges connecting vertices within a single row or column.

Corollary 3.7. *Any permutation on* $\mathrm{NN}_2(L,R)$ *can be performed in depth* $3L/R+9$.

Proof. Denote $\mathrm{NN}_1(L/R,1)\times K_R$ by H. By Lemma 3.6, H is a spanning subgraph of $\mathrm{NN}_1(L,R)$, and $\mathrm{NN}_1(L,R)\times\mathrm{NN}_1(L,R)$ is a spanning subgraph of $\mathrm{NN}_2(L,R)$. It follows that $H\times H$ is a spanning subgraph of $\mathrm{NN}_2(L,R)$, so any simple permutation for $H\times H$ is a simple permutation for $\mathrm{NN}_2(L,R)$.

Permutations on K_R and $G_2=\mathrm{NN}_1(L/R,1)$ can be implemented in depth 2 and $L/R-1$, respectively. Setting $G_1=K_R$ and $G_2=\mathrm{NN}_1(L/R,1)$ in Lemma 3.1, any permutation on H can therefore be implemented in depth $(L/R-1)+2\cdot 2=L/R+3$. Invoking Lemma 3.1 again with $G_1=G_2=H$, we can implement any permutation on $H\times H$ and hence, $\mathrm{NN}_2(L,R)$ in depth $3L/R+9$. ■

Note even though there are many edges that we do not use, the lowest depth routing can be no better than the graph diameter of $\mathrm{NN}_2(L,R)$ which is roughly $\sqrt{2}L/R$, so this is nearly optimal. Furthermore, owing to the translation invariance of $\mathrm{NN}_2(L/R,1)$, the embedding of $K_R\times K_R\times\mathrm{NN}_2(L/R,1)$ is also translation invariant—far from the boundaries, the graph remains the same locally under translation of R units in the vertical and horizontal directions.

Now, $\mathrm{NN}_2(L,R)$ is not sparse: the degree of each vertex grows as R^2. For practical purposes, it would be convenient if only a sparse subgraph of $\mathrm{NN}_2(L,R)$ were used in the routing routine. In Corollary 3.7, we used only the edges contained in a subgraph $\mathrm{NN}_1(L/R,1)\times K_R\times\mathrm{NN}_1(L/R,1)\times K_R$. $\mathrm{NN}_1(L/R,1)$ is sparse, but K_R is not. However, we can replace K_R by a sparse expander graph so that the subgraph we use is sparse. Fact 3.5 supplies such a family of graphs and a depth $O(\log^2 R)$ routing subroutine.

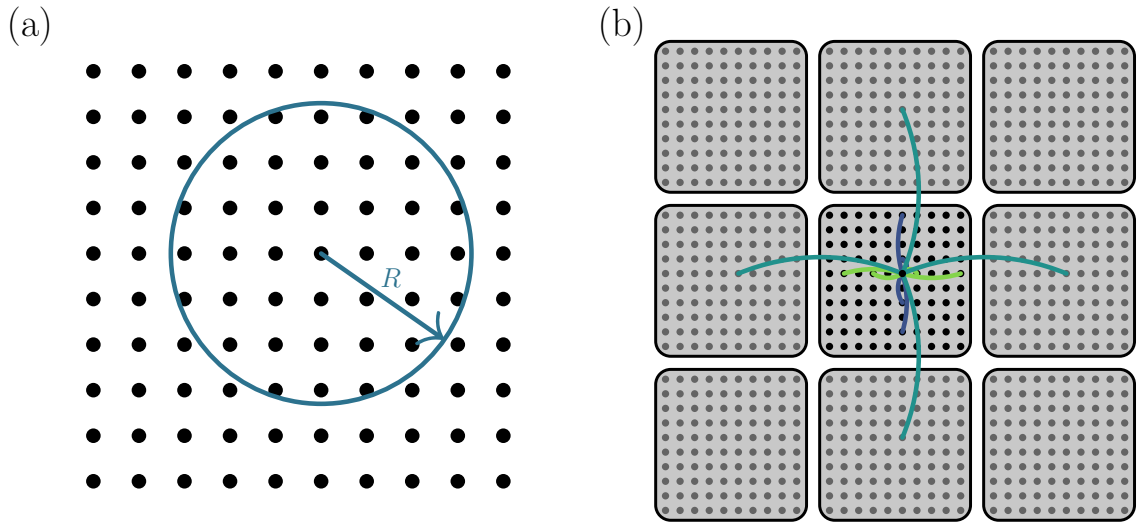

Figure 10: (a) A square lattice $\mathrm{NN}_2(L,R)$ with a qubit on each lattice point. The blue circle of radius R denotes the interaction radius for a qubit in the lattice. Such a circle exists around each qubit; we only draw one for clarity. (b) Approximating the lattice using the sparse product graph $\mathcal{E}_R\times\mathcal{E}_R\times\mathrm{NN}_2(L/R,1)$. Different colored edges come from different factors in the product.

We now bring these ideas together formally in the following corollary.

Corollary 3.8. *For R even, there is an efficiently constructable degree-12 spanning subgraph of 2-dimensional R-nearest-neighbor lattice $\mathrm{NN}_2(L,R)$ on which any permutation can be performed in depth $3L/R + O(\log^2 R)$.*

Proof. We will use replace the use of the fully connected graph in Corollary 3.7 with a sparse expander. Consider a 4-regular random graph $\mathcal{E}_R$ generated according to fact 3.4. By fact 3.5, we can route on $\mathcal{E}_R$ in depth $O(\log^2 R)$.

Now consider the graph $H = \mathcal{E}_R \times \mathrm{NN}_1(L/R, 1)$. $\mathcal{E}_R$ is a spanning subgraph of K_R, so it follows by lemma 3.6 that H is a spanning subgraph of $\mathrm{NN}_1(L/R, 1)$. Further, using Lemma 3.1, we can implement any permutation on H in depth $L/R + O(\log^2 R)$, so we can also implement any permutation on $H \times H$ in depth $3L/R + O(\log^2 R)$.

By an identical argument to Corollary 3.7, we have that $H \times H$ is a spanning subgraph of $\mathrm{NN}_2(L, R)$. Further, $H \times H$ has vertex degree 12[8] since the max vertex degree of the product of graphs is the sum of the max vertex degrees of the factors. ■

This subgraph is illustrated in figure 10 (b). Later, we will use the contents of the corollary to construct syndrome-extraction circuits with two-qubit gates of range at most R and where each qubit only needs to interact with a constant number of other qubits.

4 Bilayer implementation of hierarchical codes

In this section, we will prove Theorem 1.2. Note that R can be taken to be any number (such as 1) and the required connectivity is always sparse. We state Theorem 1.2 here in two parts and present the proof for each part in turn.

Theorem 4.1 (Theorem 1.2, Part 1)**.** *The $[\![N, K, D]\!]$ hierarchical code $\mathcal{H}_N$ is constructed by concatenating an outer code, a constant-rate $[\![n, k, d, \Delta_q, \Delta_g]\!]$ quantum LDPC code $\mathcal{Q}_n$ and an inner code, a rotated surface code $\mathcal{RS}_\ell$ where $d_\ell = \Theta(\log(n))$. Let $\rho > 0$ and $\delta \geq 1/2$, such that $k = \rho \cdot n$ and $d = \Theta(n^\delta)$. The code $\mathcal{H}_N$ has parameters*

$$K(N) = \Theta\left(\frac{N}{\log(N)^2}\right) , \qquad D(N) = \Omega\left(N^\delta / \log^{2\delta-1}\left[\frac{N}{\log(N)}\right]\right) .$$

Proof. We first define the hierarchical code family $\{\mathcal{H}_N\}$ and corresponding syndrome-extraction circuits $\{C_N^{\mathcal{H}}\}$. The element of this family indexed by $N = N(n) = n \cdot d_\ell^2$ is created by concatenating an outer LDPC code $\mathcal{Q}_n$ with an inner surface code $\mathcal{RS}_\ell$, where $\ell = \Theta(\log(n))$. Recall $\ell = 2d_\ell^2 - 1$ is the total number of qubits used to construct the rotated surface code $\mathcal{RS}_\ell$. The justification for this choice of ℓ will follow in the next section.

To express n in terms of N, the following bounds will be useful:

$$n = O\left(\frac{N}{\log(N)}\right) , \qquad n = \Omega\left(\frac{N}{\log^2(N)}\right) . \tag{11}$$

It follows from the definition of a concatenated code that the number of encoded qubits is $K = k(n)$, the code distance is $D = d(n) \cdot d_\ell$ (See Section 2.3). As $\delta \geq 1/2$, $1 - 2\delta \leq 0$. Using Equation (11), we can write

$$K = \Omega\left(\frac{N}{\log^2(N)}\right) , \qquad D(N) = \Omega\left(N^\delta \ \log^{1-2\delta}\left[\frac{N}{\log(N)}\right]\right) . \tag{12}$$

This completes the proof. ■

[8] Degree-8 can be achieved by replacing the $\mathcal{E}_R \times \mathcal{E}_R$ factor in the decomposition with a single expander graph after some modification of parameters.

The second portion of Theorem 1.2, stated below, guarantees the existence of a syndrome-extraction circuit for the hierarchical code constructed in Theorem 4.1.

Theorem 4.2 (Theorem 1.2, Part 2)**.** *There exists an explicit and efficient construction of an associated family of syndrome-extraction circuits $C_N^{\mathcal{H}}$ using only local Clifford operations and* SWAP *gates of range R such that*

$$\mathcal{W}(C_N^{\mathcal{H}}) = \Theta(N) \ , \qquad \mathcal{T}(C_N^{\mathcal{H}}) = O\left(\frac{\sqrt{N}}{R}\right) \ .$$

The rest of this section is dedicated to the proof of Theorem 4.2. We construct the syndrome-extraction circuit $C_N^{\mathcal{H}}$ for the concatenated code with the stated parameters. This circuit is constructed in a bilayer architecture and is described in detail below. A bilayer construction of the syndrome-extraction circuit $C_n^{\mathcal{Q}}$ is described in Section 4.1. To obtain $C_N^{\mathcal{H}}$, each outer qubit in the syndrome-extraction circuit $C_n^{\mathcal{Q}}$ is replaced by a copy of the inner code $\mathcal{RS}_\ell$ as described in Section 2.3.

If $C_n^{\mathcal{Q}}$ requires $\mathcal{W} = \mathcal{W}(C_n^{\mathcal{Q}})$ qubits, then we need a layout for $\mathcal{W}$ surface codes in 2 dimensions. In Section 4.2, we propose an implementation of $\mathcal{RS}_\ell^{\otimes \mathcal{W}}$ using a bilayer 2-dimensional architecture. The advantage of this architecture is that entangling gates between codes can be performed in a transversal manner which reduces the number of extra ancilla qubits. In Section 4.3, we describe a novel implementation of SWAP gates for this architecture. This completes the set of logical Clifford operations $\mathcal{K}_1$. We will bring these elements together to construct the syndrome-extraction circuit $C_N^{\mathcal{H}}$ in Section 4.5.

Before doing so, we take a brief detour in Section 4.4 to design Level-1 qubits with noise bias. We will return to this construction in Section 6 to deal with hook errors.

4.1 Syndrome-extraction circuit $C_n^{\mathcal{Q}}$ for the outer code

In this section, we design a family of syndrome-extraction circuits $\{C_n^{\mathcal{Q}}\}$ for a constant-rate $[\![n, k, d, \Delta_q, \Delta_g]\!]$ LDPC code $\{\mathcal{Q}_n\}$. We assume $k = \rho \cdot n$ for $\rho > 0$ and that $m = n - k$ is the number of stabilizer generators. In Section 2.1, we described measurement gadgets to measure each stabilizer generator. We now describe how these gadgets can be implemented in parallel subject to constraints on geometric locality. We first state the existence of an *ideal* circuit $(C_n^{\mathcal{Q}})^{\text{ideal}}$ which is not constrained by geometric locality. While we include the proof of this construction for the sake of completeness, we note that the idea and the proof itself have been used before—for example, see [DBT21]. For this reason, the proof is relegated to Appendix B.

Define constants Δ and m_0 such that

$$\Delta := \max(\Delta_q, \Delta_g) \ , \qquad m_0 := \max(m_{\mathsf{X}}, m_{\mathsf{Z}}) \ .$$

The circuit $(C_n^{\mathcal{Q}})^{\text{ideal}}$ is divided into two phases, where in each phase we measure either X or Z syndromes. Each phase requires at most $(\Delta + 2)$ stages. It satisfies

$$\mathcal{W} := \mathcal{W}[(C_n^{\mathcal{Q}})^{\text{ideal}}] = n + m_0 \ , \qquad s := \mathcal{T}[(C_n^{\mathcal{Q}})^{\text{ideal}}] = 2(\Delta + 2) \ . \tag{13}$$

The first phase proceeds as follows:

1. All m_{X} ancilla qubits are prepared in the state $|+\rangle$.
2. In each intermediate step $1 < t \leq \Delta + 1$, there is a subset P_t of all $\mathcal{W}$ qubits such that P_t is a disjoint union of m_{X} pairs of qubits, where each pair contains one ancilla and one data qubit respectively. These pairs correspond to control and target qubits, respectively, for CNOT.
3. All m_{X} ancilla qubits are measured in the X basis.

The second phase is structurally similar with minor modifications:

1. All m_{Z} ancilla qubits are prepared in the state $|0\rangle$.
2. In each intermediate step $1 < t \leq \Delta + 1$, there is a subset P_t of all $\mathcal{W}$ qubits such that P_t is a disjoint union of m_{Z} pairs of qubits, where each pair contains one ancilla and one data qubit respectively. These pairs correspond to target and control qubits, respectively, for CNOT.
3. All m_{Z} ancilla qubits are measured in the Z basis.

We now use this circuit to construct the syndrome-extraction circuit $C_n^{\mathcal{Q}}$ that is constrained by geometric locality. It will have the same space footprint $\mathcal{W}$, but its depth will be different.

Setup: Qubits are arranged in two parallel layers where each layer is a grid of dimensions $L \times L$. We assume we have access to Clifford operations $\mathcal{K}$ where, in addition to nearest-neighbor gates between two qubits in the same layer, we can perform nearest-neighbor gates between two qubits that are adjacent but in different layers. We also assume that SWAP gates of range $R > 1$ are restricted to a single layer.

Initialization: To accommodate all $\mathcal{W}$ qubits required for $(C_n^{\mathcal{Q}})^{\text{ideal}}$, it is sufficient to choose the smallest integer L that satisfies $2L^2 \geq \mathcal{W}$. Initially, data qubits and syndrome qubits are distributed arbitrarily. While further optimization is likely possible, this will not affect the asymptotics, and certainly results in an upper bound on the circuit volume.

Partition $C_n^{\mathcal{Q}}$ into stages: The circuit $C_n^{\mathcal{Q}}$ will be partitioned into s *stages*, where in each stage, we prepare and measure ancilla qubits or simulate long-range entangling gates between pairs of qubits specified by P_t. To simulate a long-range entangling gate, we use a series of SWAP gates which bring each pair specified by P_t close together, followed by the desired entangling gate when they are sufficiently close. The preparation and measurement stages are straightforward. We now describe how to perform the long-range entangling gate.

Simulating long-range: In each simulation stage, qubits are arranged such that each pair of P_t are adjacent but in different layers. In the first step of each stage, we apply a permutation to the ancilla qubit in each pair of P_t to ensure that both qubits in the pair are not in the same layer. Qubits that are not in P_t remain stationary. For simplicity, we only permute qubits in the top layer and keep the bottom layer stationary. This specifies a permutation α_t on the top layer. As this layer is an $L \times L$ lattice, we can use Algorithm 2 to design a circuit so that SWAP operations can be performed in parallel. Using Theorem 1.1, we can construct a sparse spanning subgraph of $\mathrm{NN}_2(L, R)$ with vertex degree 12 such that any permutation α_t can be accomplished in depth at most $3L/R + O(\log^2(R))$. This is followed by nearest-neighbor entangling gates as specified by P_t. The simulation stages have depth

$$3\frac{L}{R} + O(\log^2(R)) + 1 \ . \tag{14}$$

Accounting for preparation and measurement steps in each phase, the circuit $C_n^{\mathcal{Q}}$ has parameters

$$\mathcal{W}(C_n^{\mathcal{Q}}) = \mathcal{W} = n + m_0 \ , \qquad \mathcal{T}(C_n^{\mathcal{Q}}) \leq 2\Delta \left(3\frac{L}{R} + O(\log^2(R)) + 1 \right) + 4 \ . \tag{15}$$

We note that if $R = o(L)$, then the depth is $O(\sqrt{n}/R)$ (as $L = O(\sqrt{n})$). We shall assume that this is the case for the rest of the paper. We include the bound in Equation (15) with constants (ex. the 3 preceding L/R) and the dependence on the degree Δ to highlight that, for $R = 1$, this is not merely an asymptotic result and can actually be executed in practice.

The bounds in Equation (15) represent an achievability result—any quantum LDPC code can be simulated in depth $O(\sqrt{n}/R)$ as stated in the theorem above. However, it is not asymptotically tight for all code families (for example, consider the surface code). We expect future versions of this bound to depend on k and d, and how they scale as functions of n.

Circuit connectivity: In addition to providing a bound on the depth of permutations, Theorem 1.1 guarantees that each lattice position interacts with at most 12 other locations all within a range R. This implies that the connectivity of the circuit $C_n^{\mathcal{Q}}$ we have constructed can be 'static'—once qubits have been connected by wires of length at most R, we do not change it afterwards.

4.2 Implementation of the inner code

We have shown that $C_n^{\mathcal{Q}}$ is constructed using two parallel layers of qubits, where each layer is a lattice of dimensions $L \times L$. Here L is the smallest integer such that $2L^2 \geq \mathcal{W}$, where $\mathcal{W}$ is the number of qubits used by $C_n^{\mathcal{Q}}$. To construct the syndrome-extraction circuit $C_N^{\mathcal{H}}$, we use two parallel *rotated* lattices. Each qubit in $C_n^{\mathcal{Q}}$ is replaced by a rotated surface code $\mathcal{RS}_\ell$ where $\ell = \Theta(\log(n))$. As each tile uses $\ell^2 = 2d_\ell^2 - 1$ physical qubits, the circuit $C_N^{\mathcal{H}}$ requires at least $2L^2 \cdot \ell^2$ physical qubits. We also use additional physical qubits which we refer to as *buffer* qubits to facilitate logical Clifford operations between tiles. See Figure 11.

Buffer qubits are either placed along the periphery of each lattice or between tiles:

1. First, we include a thin band of "buffer" qubits along the perimeter of each layer for reasons that we will explain shortly. The band has thickness $(\ell + 1)/2$ and therefore this adds at most $2L(\ell+1)^2$ ancilla qubits per layer. These are denoted as transparent dots in Figure 11 (a).
2. Second, for each surface code, we have d_ℓ^2 data qubits, $d_\ell^2 - 1$ ancilla qubits, and 1 extra buffer qubit (light gray) for later convenience. These are denoted using dark gray, orange/blue and light gray respectively in Figure 11 (b).

In total, accounting for both qubits used in tiles as well as buffer qubits, we use at most $2(\ell + 1)^2(L + 1)^2$ physical qubits. These are arranged in two parallel (rotated) lattices of side length $(L + 1) \cdot (\ell + 1)$[9].

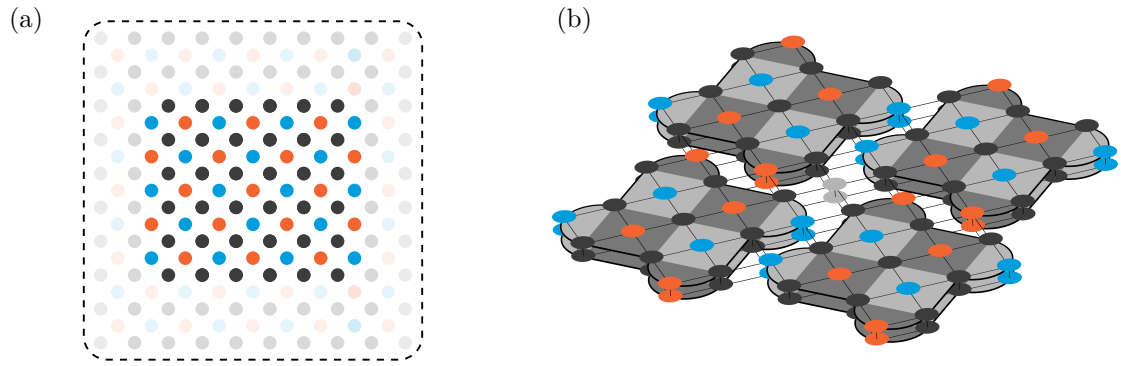

Figure 11: (a) A top-down view of one layer of the physical layout. At any given time, only some of the qubits are active; these are denoted using dots with solid color. In contrast, there are qubits that are inactive that can be used for performing logical operations; these are denoted using dots that are transparent. (b) A small $2 \times 2 \times 2$ unit cell of the physical layout containing 8 distance-3 rotated surface code tiles. Physical qubits are drawn as dark gray dots. X- and Z-type stabilizer generators within a tile are indicated by a light or dark gray region with the colored dot used as an ancilla. Thin lines indicate gate connectivity: Each qubit has 5 neighbors, 4 in-plane and 1 out-of-plane. The light gray qubit in the center is unused when the tiles are idle. Note that this layout does not contain additional ancilla qubits between tiles for lattice surgery: all operations will be performed transversally.

Physical gates can act either on two neighboring qubits in the same layer, or on adjacent qubits in different layers. Using only $\mathcal{K}_0$ operations, we construct the necessary primitives to implement the syndrome-extraction circuits $C_N^{\mathcal{H}}$.

As described in Section 4.1, the circuit $C_n^{\mathcal{Q}}$ is divided into s stages. Implementing the syndrome-extraction circuit for the outer code requires logical Clifford operations $\mathcal{K}_1$.

At the outset, both data and ancilla tiles are arranged arbitrarily. Syndromes are measured in two phases—first we measure the X-type syndromes and then the Z-type syndromes. The first and last stages of each phase correspond to single-tile logical state preparation and single-tile logical

[9]Note that there are $\sqrt{2}L\ell$ qubits to a side due to the rotated lattice.

measurements. Single-tile logical operations of state preparation and measurements can be done using only nearest-neighbor gates. If Level-0 qubits can be prepared in $|0\rangle$ or $|+\rangle$, then we can prepare Level-1 $|\overline{0}\rangle$ and $|\overline{+}\rangle$ simply by performing the syndrome-extraction circuit which projects the state into the code space. Similarly, we can perform destructive measurements of logical Pauli operators $\overline{\mathsf{X}}$ and $\overline{\mathsf{Z}}$ using single-qubit measurements of X and Z.

For each stage where we simulate long-range entangling gates, there exists a partition of outer qubits P_t; Here, P_t is the set of $m_0 = \max(m_{\mathsf{X}}, m_{\mathsf{Z}})$ pairs, where each pair has one outer ancilla qubit and one outer data qubit respectively. Depending on whether we are measuring X or Z syndromes, we perform the logical CNOT gate using the outer data qubit as target or the outer ancilla qubit as target. Data and syndrome tiles that are involved in entangling gates are always arranged such that the tiles are adjacent but in different layers. As the rotated surface code is a CSS code, we can perform CNOT gates between surface code blocks corresponding to data and ancilla qubits using transversal operations. For all these operations — single-tile preparation, single-tile measurement, and transversal CNOT — we perform surface code error correction after the operation.

To complete the description of the Level-1 syndrome extraction circuit, it only remains to explain how the logical SWAP operation is implemented. We propose a novel way to perform this gate when $R < \ell$; this is the focus of Section 4.3. We show that an arbitrary permutation requires $O(L \cdot d_\ell)$ steps. Error correction for SWAP gates is performed in an *interleaved* manner—we perform a single round of error correction after each step as described below.

We conclude Section 4.2 by discussing connectivity requirements. As mentioned in the introduction, we construct syndrome-extraction circuits such that once two lattice positions have been connected, this does not need to change dynamically over the course of the circuit. Furthermore, each lattice position only ever interacts with a constant-sized set of other lattice locations. For both error correction and logical operations, lattice positions (that store a physical qubit) will be involved in CNOT gates. It would be preferable if the pairs of lattice positions that need to interact did not change dynamically over the course of the circuit, and instead could be chosen ahead of time.

If the circuit $C_n^{\mathcal{Q}}$ is implemented such that each lattice position requires only connectivity to a constant sized set of other lattice positions, then the entire syndrome-extraction circuit $C_N^{\mathcal{H}}$ for the hierarchical code $\mathcal{H}_N$ will only use sparse connectivity of two-qubit physical gates. The proof of this claim is straightforward for single-tile logical state preparation and measurement — these are accomplished using local physical operations. Secondly, by construction, logical entangling gates are implemented transversally and therefore the connectivity does not change. Finally, we will show in Section 4.3 that for a given L and R, the connectivity required to implement an arbitrary permutation of tiles will be chosen ahead of time and will not change dynamically.

4.3 SWAP Gate

As discussed above, we can perform logical CNOT transversally. To complete $\mathcal{K}_1$, the final ingredient we need are SWAP gates. We first focus on the special case $R = 1$, and then generalize the construction to arbitrary R. To implement the permutation returned by Algorithm 2, it suffices to perform SWAP gates only along one orientation of the lattice at a time, either vertically or horizontally. This restriction is utilized to create a resource efficient SWAP operation that requires no additional ancilla qubits. The key insight is that movement of individual tiles may be accomplished by moving *all* the tiles within a single layer. By performing the transversal SWAP after movement and then moving back, we can accomplish a SWAP operation between two surface codes that are not directly on top of each other.

4.3.1 Nearest-neighbor logical SWAP gates

High-level overview: We first provide a high-level overview of the SWAP operation and refer to Figure 12. Consider a two parallel rows of tiles in the bilayer architecture as shown in Figure 12

(a). The tiles in the top row are labeled a_1,..., a_4 and the tiles in the bottom row are labeled b_1,...,b_4. The tiles labeled $\varnothing$ are buffer qubits along the periphery. For ease of visualization, the picture depicts a single-tile width of buffer qubits along the top layer. In practice, we use two half-tile width of buffer qubits in both layers. In this example, we swap tiles a_2 and a_3; however, this process can be generalized to swap tiles in parallel. This is accomplished in 5 steps:

1. The logical SWAP operation begins by exchanging alternate tiles between two rows. In the bilayer architecture, this exchange is performed in a checkerboard pattern, i.e. we swap alternate tiles along both rows and columns. This can be accomplished using what we call the *staggered* SWAP *primitive*.
2. We can then slide the entire top layer one tile width to the left. In the bilayer architecture, this will be accomplished using what we call the *walking primitive* that we describe below. The top layer will shift a half-tile width in one direction while the bottom layer will move a half-tile width in the other. This is the reason we use buffer qubits along the periphery—to accommodate tiles after the walk step.
3. Pairs of tiles that we wish to exchange are now adjacent in different layers. For each pair that we wish to swap, we perform a swap operation using nearest-neighbor SWAP gates between adjacent layers.
4. The last two steps are the inverse of the first two steps—we apply the walk primitive and then perform a swap operation between layers on alternate tiles.

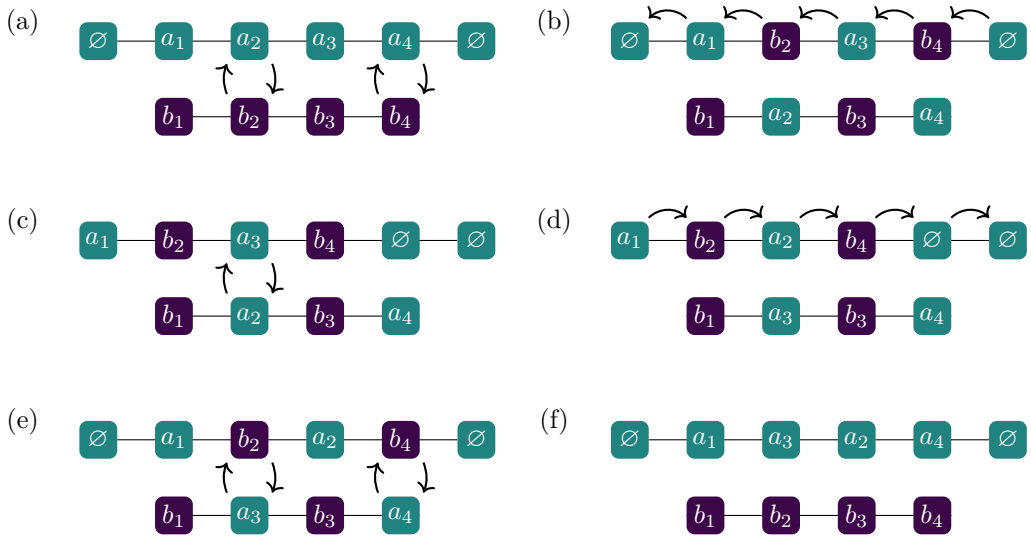

Figure 12: Consider two parallel rows of tiles $a_1, ..., a_4$ and $b_1, ..., b_4$ in different layers, one on top of another. Tiles are depicted as squares. The tiles with the label $\varnothing$ represent a tile width of buffer qubits on the periphery of the top layer. This example demonstrates how to swap two tiles a_2 and a_3. To begin, we swap alternate tiles in each row as shown in (a). We then use the walk operation to move tiles one unit as shown in (b). For every pair of tiles we wish to exchange, we perform a inter-layer SWAP as shown in (c). In this example, we only wish to swap tiles a_2 and a_3 so the other tiles remain stationary. We then undo the transformation by reversing the walk in (d) and undoing the alternate exchange in (e). The final panel (f) is the desired state.

Walking primitive: By placing a 1/2-tile wide strip of buffer qubits on the periphery of the lattice, we can "walk" the entire memory by swapping the physical-level ancilla qubits and surface code data qubits (Figure 14).[10] Using this walking primitive, we can move an entire layer an entire tile width in depth $2d_\ell$ using only SWAP-gates. This is a global operation. We will use this primitive in two ways: First, we implement a transversal SWAP between two tiles that are in different layers in the staggered-SWAP primitive (explained below). Second, we will use this repeatedly to move an entire layer half-a-tile width in some direction.

[10]Not to be confused with the extra buffer qubit per tile.

Staggered SWAP primitive: When possible, we would like to avoid applying gates directly between data qubits of surface code blocks as this would introduce (small) extra correlations in the logical failure probability of tiles. Instead of a direct transversal swap between data blocks, the vertical SWAP can instead be performed between data qubits in one layer and syndrome qubits in the other layer. This is accomplished via the staggered SWAP primitive. See Figure 13.

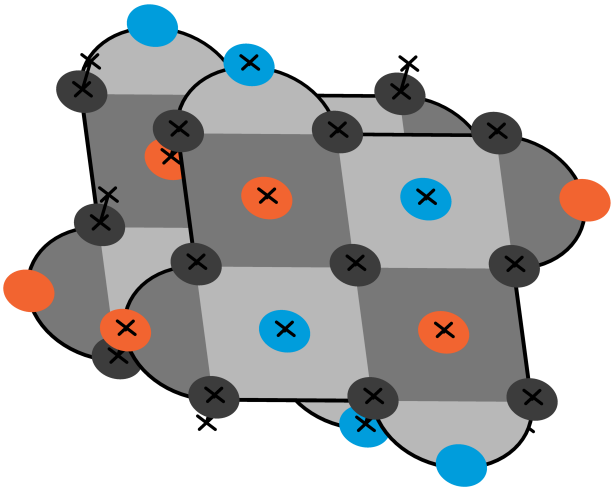

Figure 13: Performing a staggered SWAP operation between layers. Physical qubits are arranged such that data (ancilla) qubits in the top layer are adjacent to ancilla (data) qubits in the bottom layer. Each qubit in the top layer is swapped with the qubit immediately below it. This allows us to exchange tiles between layers without two data qubits directly interacting with each other.

By default, the qubits in the two layers are positioned such that a data qubit in the top layer is above a data qubit in the bottom layer. This facilitates performing logical CNOT gates via transversal physical CNOT gates. We can perform a stagger operation using the walking primitive. This positions data qubits in the top layer above ancilla qubits in the bottom layer. We can then apply a transversal SWAP between layers, and undo the stagger operation if need be. Over the course of the Level-1 logical SWAP, however, we only need to undo the stagger operation at the very end. Throughout the logical SWAP, the two layers remain staggered. If syndrome qubits are reset before use in a syndrome-extraction round, the surface code blocks have undergone a somewhat complicated idle operation with no correlated errors generated between the two surface code blocks.

Level-1 logical SWAP: We are ready to describe the logical SWAP operation.

We begin by exchanging every other tile. This procedure is illustrated in Figure 15; this can be compared to Figure 12. Steps 1 and 3 (half step shifts) are performed globally with step 2A (vertical swap) or 2B (half step shifts) performed whether or not a logical swap or logical identity is scheduled for a given tile. The step 2B is necessary for the logical identity gate, because step 2A would otherwise lead to data qubits of adjacent tiles directly next to each other instead of separated by an ancilla qubit. In this way, syndrome extraction can optionally be performed after every layer of SWAP gates.

To summarize, our SWAP-gate is implemented as follows where we account for the depth of each operation:

1. Use the staggered SWAP on every other tile in a checkerboard pattern to put them in different layers (depth-3).
2. Use the walking primitive to translate the top and bottom layers d_ℓ lattice sites in opposite directions so that originally adjacent tiles are now stacked (depth-d_ℓ).
3. Optionally perform a staggered SWAP (depth-3).
4. Translate the top and bottom layers d_ℓ lattice sites back (depth-d_ℓ)
5. Bring the tiles back to the same layer by undoing a staggered SWAP (depth-3).

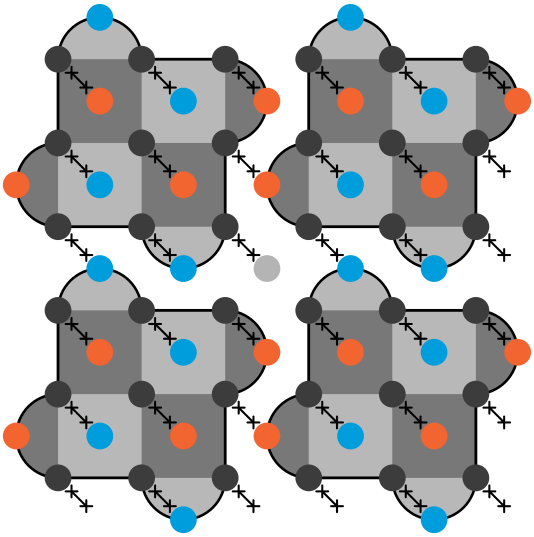

Figure 14: Walking primitive used in the SWAP implementation with qubits outside of the 2×2 unit cell not drawn. Swaps of Level-0 qubits are drawn as black lines with crosses. Data qubits become ancilla qubits and ancilla qubits become data qubits, so that syndrome extraction is possible at every step. A full 1-unit step to the right of the data qubits can be accomplished following this $1/2$ unit step by swapping each (initially) syndrome qubit with the data qubit up and to the right. Later, we will increase the speed at which the top and the bottom layers are shifted relative to each other by shifting the top layer in one direction and the bottom layer in the other.

At every step, we have the necessary ancilla qubits to perform surface code syndrome extraction. At first, we might think to perform d_ℓ rounds of error correction after each of the 5 steps above. However, because we are working with transversal SWAP operations, we only perform a *single* round of error correction after each step. Ideal SWAP gates do not spread errors, and therefore, the SWAP operation can be seen as a syndrome-extraction circuit on each tile with a higher failure rate. While performing just a single round of error correction may reduce the threshold, we expect this change to be minimal. Furthermore, it reduces the depth of the logical SWAP operation, so our tile SWAP-gate takes $2d_\ell + 9$ steps of physical SWAP-gates.

Recall from Section 4.1 that the syndrome-extraction circuit $C_n^{\mathcal{Q}}$ is split into $s = 2\Delta + 4$ stages. Besides the preparation and measurement stages, we simulate a long-range CNOT between pairs of data and ancilla qubits in each stage. The stage begins by picking an element of each pair and ensuring that they are in different layers. We can then use the logical SWAP described here to permute tiles.

Note that the SWAP operations can be performed in parallel between tiles on the same layer; the SWAP operations exchange tiles in the same row or column as required by the routing algorithm presented in Algorithm 2. We can use Lemma 3.1 to show that any permutation of tiles on an $L \times L$ lattice can be accomplished in $3L - 3$ steps. This allows any desired permutation on the $L \times L \times 2$ lattice of tiles to be accomplished in depth $(2d_\ell + 9)(3L - 3)$.

In fact, we can optimize this further to avoid repeating redundant operations. For all but the first and last swap operations: 1) Steps 1 and 5 can be omitted. 2) Within step 3, we may omit the walking step that offsets the upper and lower layers by a half lattice site, so the staggered SWAP becomes a simple transversal swap. Using these optimizations, any permutation on the $L \times L \times 2$ lattice of tiles can be accomplished in depth t_{route} where

$$t_{\text{route}} := (2d_\ell + 1)(3L - 3) + 8 \ . \tag{16}$$

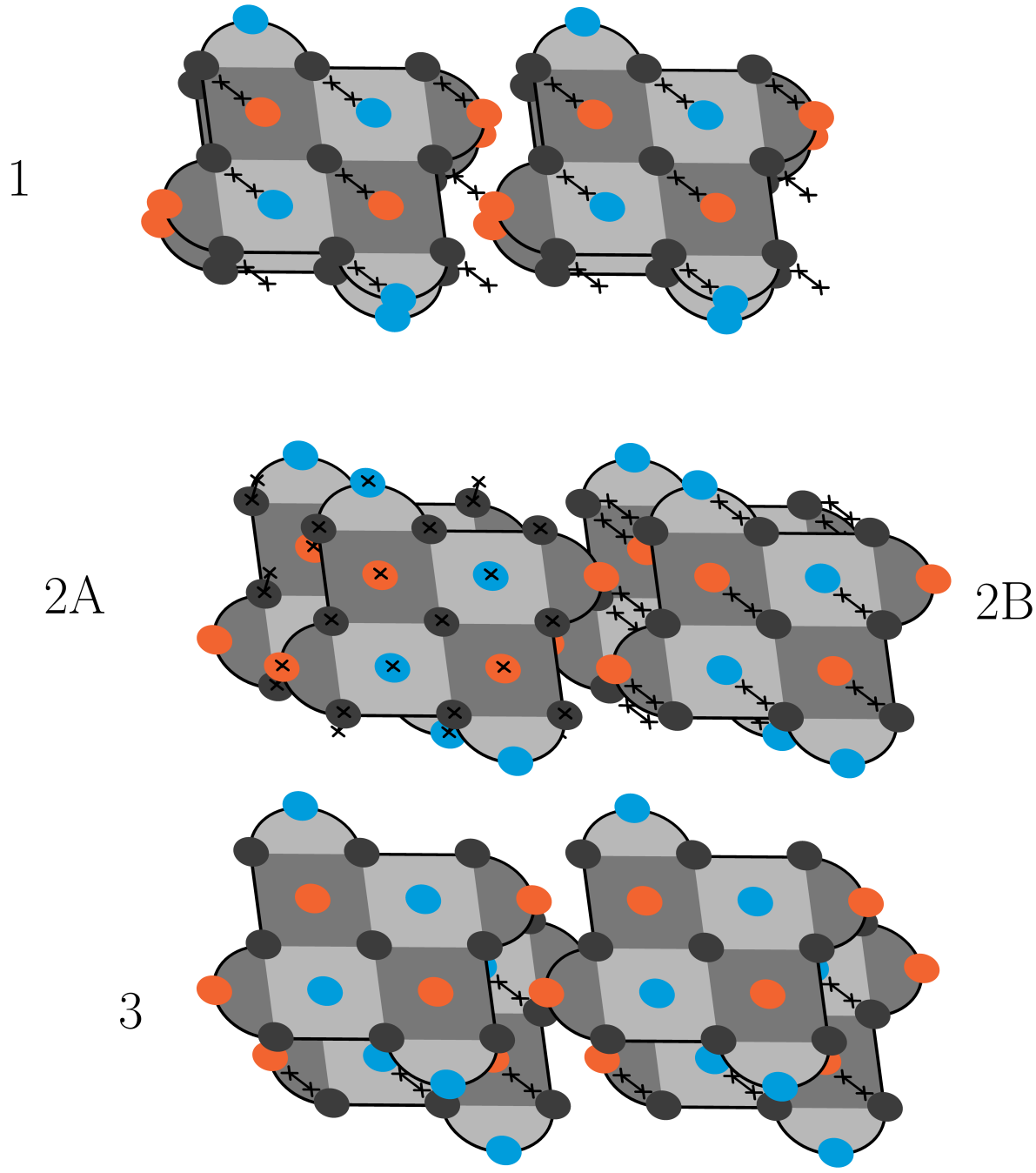

Figure 15: 3-step transversal swap implementation between two stacked syndrome tiles that avoids directly swapping data qubits. The pair of tiles on the right undergoes an identity operation while the pair of tiles on the left are swapped. In step 1, the top layer is shifted by a half-unit using SWAP gates on the top layer. In step 2, the pairs of tiles are either swapped using vertical SWAP gates (2A) or shifted using horizontal SWAP gates to keep alignment (2B). Finally, in step 3, the lower layer is shifted back by a half-unit using SWAP gates in the bottom layer. This operation has the property that syndrome extraction can be performed in all three timesteps. The perspective is inclined slightly to show both layers.

4.3.2 Logical permutation routings

We restrict our attention to range R SWAP gates in a single layer. Interlayer operations are strictly nearest-neighbor gates, and can be accomplished using the primitives discussed in Section 4.3.1. In the following lemma, we show that an arbitrary permutation routing of tiles can be accomplished in depth $O(L\ell/R)$.

Lemma 4.3. *Consider access to physical* SWAP *operations with range $R = o(L \cdot \ell)$. We can implement any arbitrary permutation of Level-1 qubits in the bilayer architecture in depth*

$$O\left(L\ell/R\right).$$

Proof. We proceed in two cases, $R \geq \ell$ and $R < \ell$.

Case 1: $R \geq \ell$

Level-0 SWAP gates of range R can be used to implement Level-1 transversal gates of range $R_1 = \lfloor R/\ell \rfloor$. We can route on the Level-1 lattice $\mathrm{NN}_2(L, R_1)$ using Corollary 3.8 with Level-1 tiles swapped transversally. This guarantees that the depth of any permutation routing of tiles is $O(L/R_1)$. Each transversal SWAP is followed by a single round of syndrome extraction of the rotated surface code; this requires constant depth and does not affect the depth of permutation routing.

Case 2: $R < \ell$

The range R Level-0 SWAP gate can be used to speed up the walking primitive presented in Section 4.3.1. Parallelized Level-1 nearest-neighbor SWAP gates implemented in this way take time $\Theta(\ell/R)$. Combined with the Corollary 3.2, we have that routing takes time $O(L\ell/R)$. ■

4.4 Biased-noise qubits

In Section 6.4.1, we will be interested in suppressing certain kinds of Pauli errors. When a qubit experiences X or Z errors with an asymmetric rate, it is said to be noise-biased. In this section, we will explain how to introduce such a noise bias on Level-1 qubits by modifying the bilayer architecture.

Let $\eta \geq 1$ be the desired noise bias of the Level-1 qubits, and suppose Z errors occur with a probability p and are η-times more likely to occur than X or Y errors. We can introduce a noise bias $\eta > 1$ on Level-1 qubits by elongating the surface code into rectangular regions where the minimum-weight X logical operator is longer than the minimum weight Z logical operator. See Figure 16. Suppose we have a rectangular surface code patch of dimensions d_X by d_Z such that the minimum weight X logical operator has weight d_X and the minimum weight Z logical operator has weight d_Z. Considering the minimum weight logical operators, we expect a failure rate $\propto p^{\lceil d_\mathsf{X}/2 \rceil}$ in the X basis and $\propto p^{\lceil d_\mathsf{Z}/2 \rceil}$ in the Z basis. By taking $d_\mathsf{X} > d_\mathsf{Z}$, we can introduce a noise bias.

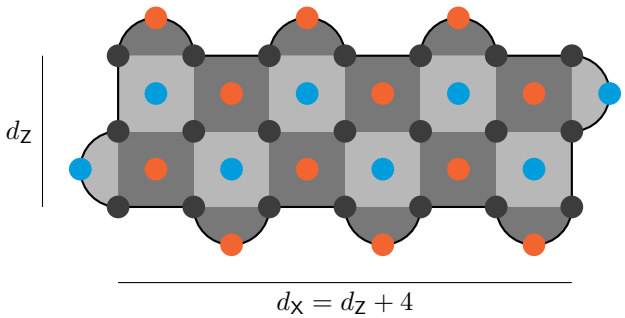

Figure 16: Creating a biased Level-1 qubit by using a rectangular surface code. In this picture $d_\mathsf{X} = 7$ and $d_\mathsf{Z} = 3$. The bias is therefore $\eta = O(p^{-2})$.

For simplicity, we assume $d_{\mathsf{X}} = d_{\mathsf{Z}} + \lceil \log(\eta)/\log(p) \rceil$ to guarantee a bias of *at least* η. We also assume that all Level-1 qubits, data and ancilla both, have been biased. We update the bilayer architecture to use two $L_{\mathsf{X}} \times L_{\mathsf{Z}}$ grids of rotated surface code tiles with X and Z distances d_{X} and d_{Z} respectively. Consider a constant-rate LDPC code $\mathcal{Q}_n$ with rate ρ. To accommodate $\mathcal{W}$ outer qubits, we let L be defined by $2L^2 = \mathcal{W}$. We then define L_{Z} and L_{X} to be the smallest integers satisfying $L_{\mathsf{Z}} \geq L \cdot \sqrt{d_{\mathsf{X}}/d_{\mathsf{Z}}}$ and $L_{\mathsf{X}} \geq L \cdot \sqrt{d_{\mathsf{Z}}/d_{\mathsf{X}}}$.

It is not sufficient that the qubits themselves are noise biased. In addition, we also require that all gates preserve this bias. We consider each Clifford operation in turn:

1. **Preparation & Measurement:** Within the syndrome-extraction circuit for the outer code where all measurement ancilla qubits are prepared in the $|\overline{+}\rangle$ state (Section 2.1), X errors on the ancilla qubits are suppressed by a factor of η.
2. **Entangling gates:** In the bilayer architecture, entangling gates are performed transversally. Transversal gates are naturally bias preserving.
3. SWAP **gates:** The way we perform SWAP operations does not change as all tiles have the same dimensions. The buffer region on the periphery of the lattice must be slightly increased to accommodate the elongated surface code tiles during the walking operation. SWAP operations themselves do not spread errors and are also bias preserving. [11]

Together, this completes the requirements for implementing logical Clifford operations $\mathcal{K}_1$.

4.5 Syndrome-extraction circuits for hierarchical codes

The syndrome-extraction circuit $C_N^{\mathcal{H}}$ for $\mathcal{H}_N$ is the circuit $C_n^{\mathcal{Q}}$ where we replace each outer qubit by a tile $\mathcal{RS}_\ell$. In the circuit $C_N^{\mathcal{H}}$, each gate of $C_n^{\mathcal{Q}}$ from the set $\mathcal{K}$ is replaced by the corresponding element in $\mathcal{K}_1$ followed by surface code error correction on each outer qubit. Recall that in Section 4.2, we discussed how to perform preparation, measurement and logical entangling gates. In Section 4.3, we discussed how to perform SWAP gates.

Theorem 4.4. *Each element $\mathcal{H}_N$ has an associated 2-dimensional syndrome-extraction circuit $C_N^{\mathcal{H}}$ with the following properties:*

$$\mathcal{W}(C_N^{\mathcal{H}}) = \Theta(N) \ , \qquad \mathcal{T}(C_N^{\mathcal{H}}) = O\left(\frac{\sqrt{N}}{R}\right) \ .$$

Further, each lattice position in $C_N^{\mathcal{H}}$ only interacts with a fixed set of other lattice positions whose size is independent of N.

Proof. Consider the family of $[\![n, k, d, \Delta_q, \Delta_g]\!]$ quantum LDPC codes $\{\mathcal{Q}_n\}$ of constant rate $\rho > 0$. From Section 4.1, $\mathcal{W}(C_n^{\mathcal{Q}}) = \Theta(n)$. To construct $C_N^{\mathcal{H}}$, each qubit in $C_n^{\mathcal{Q}}$ is replaced by a surface code $\mathcal{RS}_\ell$. It follows that

$$\mathcal{W}(C_N^{\mathcal{H}}) = \Theta(\ell)^2 \cdot \Theta(n) = \Theta(N) \ . \tag{17}$$

Secondly, each $\mathcal{K}_1$ operation in $C_N^{\mathcal{H}}$ requires depth $\Theta(\ell)$ for error correction. This is because entangling gates are implemented transversally followed by d_ℓ rounds of error correction and, per Lemma 4.3, the Level-1 logical SWAP operation requires depth $O(L \cdot \ell/R) = O(\sqrt{N}/R)$.

This implies that

$$\mathcal{T}(C_N^{\mathcal{H}}) = O\left(\frac{\sqrt{N}}{R}\right) \ . \tag{18}$$

[11] Recall that for two single qubit operators A and B, $\mathsf{SWAP}(A \otimes B) = (B \otimes A)\mathsf{SWAP}$.

By assumption, logical two-tile operations in $\mathcal{K}_1$ can be implemented such that the set of lattice positions that interact with each other remain fixed. Furthermore, the construction of Section 4.1 guarantees that each of the outer positions in $C_N^{\mathcal{H}}$ only interact with a fixed and constant-sized set of other outer positions. Therefore, all of the physical positions of $C_N^{\mathcal{H}}$ need only interact with a fixed and constant-sized set of positions in $C_N^{\mathcal{H}}$.

This completes the proof. ■

5 Overhead, threshold and asymptotics

In this section, we prove that the hierarchical code has a threshold if we use the syndrome-extraction circuits $C_N^{\mathcal{H}}$ presented in Section 4. We present a formal version of Theorem 1.3.

We recall the bound by Delfosse *et al.* in Equation (1)

$$\mathcal{T}(C_n^{\mathcal{Q}}) = \Omega\left(\frac{n}{\sqrt{\mathcal{W}(C_n^{\mathcal{Q}})}}\right) . \tag{1}$$

According to this bound, the physical circuit $C_n^{\mathcal{Q}}$ cannot have constant depth and space footprints simultaneously. This blowup in the volume of the circuit introduces additional failure modes. Consequently, saturating these bounds by no means ensures the existence of a threshold. This then seems to have defeated the purpose of simulating a non-local circuit using local operations.

In this section, we present a geometrically-local construction of a circuit $C_N^{\mathcal{H}}$ that encodes a growing number of encoded qubits *and* guarantees that a threshold exists. We use code concatenation to define a family $\{\mathcal{H}_N\}$ that we call *hierarchical* codes. The N^{th} element of this family is obtained by concatenating an LDPC code $\mathcal{Q}_n$ and a rotated surface code $\mathcal{RS}_\ell$ of size $\Theta(\ell^2)$. Here $N = \Theta(n \cdot \ell^2)$.

In Section 5.1, we study the failure rate per round p_{round} and establish its dependence on p_{phys} for a syndrome-extraction circuit for any $[\![n, k, d, \Delta_q, \Delta_g]\!]$ quantum LDPC code. The circuits are themselves faulty and are described by a locally decaying faults model. We show in Section 5.2 that logical gates in the bilayer architecture guarantee that Level-1 logical errors in surface code blocks are suppressed exponentially following logical Clifford operations. This allows us to deal with the Level-1 syndrome-extraction circuit for the outer code directly without having to keep track of Level-0 failure probabilities. Permuting tiles can introduce Level-1 correlated errors among the tiles. In Section 5.3, we show that there exists a choice of ℓ such that the $p_{\text{round}}^{(1)}$ is an arbitrarily small constant. We can then invoke Gottesman's threshold theorem which we discussed in Section 2.4 to prove the existence of a threshold.

We conclude this overview by highlighting some features of this construction.

1. We show that $\ell = \Theta(\log(n))$ is sufficient to achieve a threshold for the outer code. If the code $\mathcal{Q}_n$ has constant rate, then the code $\mathcal{H}_N$ has rate $O(\log(n)^{-2}) \to 0$ as $n \to \infty$.
2. Although outer and inner codes are both LDPC codes, $\mathcal{H}_N$ is itself no longer an LDPC code—it uses stabilizer measurements that have weight $\log(n)$ as the side length of the inner surface codes is $\ell = \Theta(\log(n))$. However, the logical operators of the surface code can be measured (destructively) using only single-qubit measurements.
3. There is a cost to locality—the sub-threshold scaling of the logical failure rate is qualitatively different from the typical exponential error suppression as a function of the distance. Instead, we see a strictly sub-exponential, but still superpolynomial, suppression of the logical failure rate with the distance.

5.1 Evolution of errors in syndrome-extraction circuit $C_n^{\mathcal{Q}}$

Consider an $[\![n, k, d, \Delta_q, \Delta_g]\!]$ LDPC code family $\{\mathcal{Q}_n\}$ with constant rate, i.e. $k = \rho \cdot n$ for some constant $\rho \in (0, 1)$ and distance $d = \Theta(n^\delta)$ for some constant $\delta \in (0, 1]$. In Section 2.4, we discussed

how Gottesman's proof for the existence of a threshold for LDPC codes depends on the number of errors *per round* of syndrome extraction on both data and ancilla qubits. In this section, we shall study how the probability of errors per round depends on the circuit $C_n^{\mathcal{Q}}$.

Recall the circuit $C_n^{\mathcal{Q}}$ described in Section 4.1. Qubits are arranged on two parallel lattices where each lattice has dimensions $L \times L$. Here, L is the smallest natural number that satisfies $2L^2 \geq \mathcal{W}$. In total the circuit has $s = 2\Delta + 4$ stages which can be broken down as follows. The syndrome-measurement circuit $C_n^{\mathcal{Q}}$ is divided into two phases, one for each of X- and Z-type syndrome measurements. Each phase is further divided into $\Delta + 2$ stages, where $\Delta = \max(\Delta_q, \Delta_g)$. In addition to one stage each to prepare and measure ancillas, there are Δ stages where we simulate long-range entangling gates. In each such stage, we permute qubits in the lattice using SWAP gates. Given access to SWAP operations of range R, the permutation has bounded depth $\mathcal{T}_{\text{perm}} = O(L/R)$. This is then followed by nearest-neighbor entangling gates.

Our first result is technical and allows one to compose locally decaying distributions.

Lemma 5.1. *Let* $\Pr_1$ *and* $\Pr_2$ *be two independent and locally decaying distributions on* $[n]$ *with rates* p_1 *and* p_2*. Consider the distribution* $\widehat{\Pr} : \mathrm{Pow}(n) \to \mathbb{R}$ *defined as*

$$\widehat{\Pr}(E) = \sum_{\substack{E_1, E_2 \subseteq [n] \\ E \subseteq E_1 \cup E_2}} \widehat{\Pr}_2(E_1) \cdot \widehat{\Pr}_1(E_2) \ . \tag{19}$$

Then Pr *is a locally decaying distribution with rate* $p = p_1 + p_2$*, i.e. for all* $E \subseteq [n]$*,*

$$\Pr(E) \leq (p_1 + p_2)^{|E|} \ . \tag{20}$$

Proof. For $E \subseteq [n]$, we can write

$$\begin{aligned}
\Pr(E) &= \sum_{\substack{E_1, E_2 \subseteq [n] \\ E \subseteq E_1 \cup E_2}} \widehat{\Pr}_1(E_1) \widehat{\Pr}_2(E_2) && (21) \\
&\leq \sum_{\substack{E_1, E_2 \subseteq E \\ E = E_1 \sqcup E_2}} \Pr_1(E_1) \Pr_2(E_2) && (22) \\
&\leq \sum_{w=0}^{|E|} \binom{|E|}{w} p_1^w p_2^{|E|-w} && (23) \\
&= (p_1 + p_2)^{|E|} && (24)
\end{aligned}$$

The result follows. ∎

Lemma 5.2. *Let* $\mathbf{e} \in \mathbb{F}_2^n$ *be a random binary vector such that* $E = \mathrm{supp}(\mathbf{e})$ *is distributed according to a locally decaying distribution with rate* p*. Let* $\mathbf{M} \in \mathbb{F}_2^{m \times n}$ *with row and column weight at most* Δ *and* $\mathbf{f} = \mathbf{M} \cdot \mathbf{e}$ *be a random variable induced from* $\mathbf{e}$*. Then* $F = \mathrm{supp}(\mathbf{f})$ *is distributed according to a* $2^{\Delta} p^{1/\Delta}$ *locally decaying distribution.*

Proof. For a set of bits $A \subseteq [n]$, denote its image under $\mathbf{M}$ by $\mathbf{M}(A)$ defined as the union of columns $\mathbf{M}_i$ of $\mathbf{M}$ in A. I.e. $\mathbf{M}(A) := \bigcup_{i \in A} \mathrm{supp}\, \mathbf{M}_i$. The probability that an error set $F \subseteq [m]$ on the output occurs

$$\Pr(F) = \sum_{\substack{E \subseteq [n] \\ F \subseteq \mathbf{M}(E)}} \widehat{\Pr}(E) \ , \tag{25}$$

i.e. in order for F to have occurred, there must be a set of errors on the input such that F is in the image. We can rewrite this sum in terms of the *largest subset* of the powerset $\mathcal{I} \subseteq \mathrm{Pow}(n)$ such that for any single set $E \in \mathcal{I}$:

1. We have that $F \subseteq \mathbf{M}(E)$.

2. For all non-empty subsets $G \subset E$, $F \nsubseteq \mathbf{M}(E \setminus G)$.

Each element of $\mathcal{I}$ is minimal in the sense that it is a subset of no other element of $\mathcal{I}$ (Assumption 2) while still having F in its image (Assumption 1). The second condition will allow us to replace the $\widehat{\Pr}(\cdot)$ with $\Pr(\cdot)$ in the sum without loosening the upper bound. Additionally, the column weight of $\mathbf{M}$ is at most Δ, so the size of each element $E \in \mathcal{I}$ is at least $|F|/\Delta$. Using the locally decaying distribution assumption yields

$$\Pr(F) \leq \sum_{E \in \mathcal{I}} \Pr(E) \tag{26}$$

$$\leq |\mathcal{I}| \cdot p^{|F|/\Delta} \ . \tag{27}$$

It now remains to count the number of elements of $\mathcal{I}$: Let $J \subseteq [m]$ be the preimage of F in the sense that for every element e in J, the intersection of $\mathbf{M}(\{e\})$ with F is not empty. The row weight of $\mathbf{M}$ is at most Δ, so J is no larger than $|F|\Delta$. Every set E in $\mathcal{I}$ must satisfy $E \subseteq J$ or else there would be some element $a \in E$ such that $F \subseteq \mathbf{M}(E \setminus \{a\})$ (contradicting Assumption 2 on $\mathcal{I}$). Finally, there are $2^{|J|}$ subsets of J, so $|\mathcal{I}| \leq 2^{|J|} \leq 2^{|F|\Delta}$. Continuing with the bound, we have that

$$\Pr(F) \leq |\mathcal{I}| \cdot p^{|F|/\Delta} \tag{28}$$

$$\leq 2^{|F|\Delta} \cdot p^{|F|/\Delta} \tag{29}$$

$$= \left(2^{\Delta} \cdot p^{1/\Delta}\right)^{|F|} \ . \tag{30}$$

The result follows. ■

The syndrome-extraction circuit $C_n^{\mathcal{Q}}$ has a special structure—errors do not spread from one data qubit to another or from one ancilla qubit to another. We show that this implies that D and A are distributed according to a locally decaying distribution. Before doing so, we review the *symplectic representation* formalism which we use in the following proofs.

The symplectic representation: For $\mathcal{W} \in \mathbb{N}$, consider any Pauli operator $\mathsf{P} \in \mathcal{P}_{\mathcal{W}}$ and suppose it is expressed as $\mathsf{P} = \mathsf{X}(\mathbf{p}_x)\mathsf{Z}(\mathbf{p}_z)$ for $\mathbf{p}_x, \mathbf{p}_z \in \mathbb{F}_2^{\mathcal{W} \times 1}$. Clifford unitary operators U map Pauli operators to Pauli operators under conjugation, i.e. $U\mathsf{P}U^\dagger$ is a Pauli operator. Equivalently, this can be represented as a linear map on $\mathbf{p}_x$ and $\mathbf{p}_z$. Corresponding to U, there exists a matrix $\mathbf{M} \in \mathbb{F}_2^{2\mathcal{W} \times 2\mathcal{W}}$[12] such that the action of U on P can equivalently can be expressed as

$$\begin{pmatrix} \mathbf{p}_x \\ \mathbf{p}_z \end{pmatrix} \to \mathbf{M} \cdot \begin{pmatrix} \mathbf{p}_x \\ \mathbf{p}_z \end{pmatrix} \pmod 2 \ . \tag{31}$$

All arithmetic on symplectic vectors is performed modulo 2; we drop the 'mod 2' suffix in the equations that follow.

Recall that the $\mathcal{W} = \mathcal{W}(C_n^{\mathcal{Q}})$ qubits in $C_n^{\mathcal{Q}}$ are partitioned into data qubits and ancilla qubits respectively. Controlled-P gates for $\mathsf{P} \in \mathcal{P}$ only use the ancilla qubits as control and data qubits as target. Let $\mathbf{d}_x$, $\mathbf{d}_z \in \mathbb{F}_2^{n \times 1}$ and $\mathbf{a}_x$, $\mathbf{a}_z \in \mathbb{F}_2^{m_0 \times 1}$ represent the Pauli operators D and A on data and ancilla qubits respectively. For the purposes of understanding how errors accumulate over one round of syndrome measurements, we are not interested in the physical locations of the qubits. As far as their action on D and A are concerned, we treat SWAP gates as (noisy) idle gates [13].

In any given time step of $C_n^{\mathcal{Q}}$ where we apply entangling gates, all qubits interact with the same type of gate (CNOT or CZ) or remain idle. The corresponding symplectic matrices have a very special form. We can write the joint evolution of D and A under the Clifford transformation acting on the X- and Z-components separately:

[12] The matrix $\mathbf{M}$ has additional structure—it is symplectic [Got97], but this is not relevant for this proof.

[13] Colloquially, SWAP gates change the locations of qubits in physical space, not in 'math' space. For instance, suppose each qubit has a label $1, ..., n$ and we choose to represent the vector $\mathbf{p}_x$ as $(\mathbf{p}_x(1), ..., \mathbf{p}_x(n))$, where $\mathbf{p}_x(i)$ represents the Pauli operator on the i^{th} qubit. Then moving qubits around in physical space using SWAP gates does not affect the i^{th} component $\mathbf{p}_x(i)$. For this reason, we ignore the action of SWAP on $\mathbf{p}_x$ and $\mathbf{p}_z$.

1. If qubits are only involved in CNOT operations that use ancilla qubits as control qubits and data qubits as target qubits, then there exists a matrix $\mathbf{M} \in \mathbb{F}_2^{m_X \times n}$ such that

$$\begin{pmatrix} \mathbf{d}_x \\ \mathbf{a}_x \end{pmatrix} \to \begin{pmatrix} (\mathbf{M})^T \cdot \mathbf{a}_x + \mathbf{d}_x \\ \mathbf{a}_x \end{pmatrix} , \qquad \begin{pmatrix} \mathbf{d}_z \\ \mathbf{a}_z \end{pmatrix} \to \begin{pmatrix} \mathbf{d}_z \\ \mathbf{a}_z + \mathbf{M} \cdot \mathbf{d}_z \end{pmatrix} . \tag{32}$$

   For every pair of qubits indexed by $i \in [m_0]$ and $j \in [n]$ that are the control and target of a CNOT, the (i, j) entry of $\mathbf{M}$ is 1. The other entries are 0. In this setting, we note that $\mathbf{a}_x$ and $\mathbf{d}_z$ remain invariant.
2. If qubits are only involved in CZ operations that use ancilla qubits as control qubits and data qubits as target qubits, then there exists a matrix $\mathbf{N} \in \mathbb{F}_2^{m_Z \times n}$ such that

$$\begin{pmatrix} \mathbf{d}_x \\ \mathbf{a}_x \end{pmatrix} \to \begin{pmatrix} \mathbf{d}_x \\ \mathbf{a}_x \end{pmatrix} , \qquad \begin{pmatrix} \mathbf{d}_z \\ \mathbf{a}_z \end{pmatrix} \to \begin{pmatrix} \mathbf{d}_z + \mathbf{N}^T \cdot \mathbf{a}_x \\ \mathbf{a}_z + \mathbf{N} \cdot \mathbf{d}_x \end{pmatrix} . \tag{33}$$

   For every pair of qubits indexed by $i \in [m_Z]$ and $j \in [n]$ that are the control and target of a CZ, the (i, j) entry of $\mathbf{N}$ is 1. The other entries are 0. In this setting, we note that $\mathbf{d}_x$ and $\mathbf{a}_x$ remain invariant.

In the symplectic representation, we can see that the structure of a syndrome-extraction circuit is special because in each phase where we measure either X or Z syndromes, there is always an invariant subspace (for example, $\mathbf{d}_x$, $\mathbf{a}_z$ when measuring X-type syndromes).

The induced error model: In the symplectic representation, a faulty Clifford operation can be expressed as an affine map—there exists random variables $\mathbf{b}_x$, $\mathbf{b}_z \in \mathbb{F}_2^{\mathcal{W} \times 1}$ such that the noisy operation can be expressed as

$$\begin{pmatrix} \mathbf{q}_x \\ \mathbf{q}_z \end{pmatrix} = \mathbf{M} \cdot \begin{pmatrix} \mathbf{p}_x \\ \mathbf{p}_z \end{pmatrix} + \begin{pmatrix} \mathbf{b}_x \\ \mathbf{b}_z \end{pmatrix} . \tag{34}$$

The errors $\mathbf{b}_x$, $\mathbf{b}_z$ are caused by faults. The faults are themselves are distributed according to the locally decaying distribution $\mathcal{F}$ with failure rate p_{phys}. Let $\mathcal{X}$, $\mathcal{Z}$ be the induced distributions over $\mathbf{b}_x$ and $\mathbf{b}_z$. For example, $\mathcal{X}(\mathbf{b}_x)$ represents the sum of the probabilities over all events where the error is $(\mathbf{b}'_x, \mathbf{b}'_z)$ such that $\text{supp}(\mathbf{b}_x) \subseteq \text{supp}(\mathbf{b}'_x)$. In other words, it represents the total probability that the error has a non-trivial X component on $\text{supp}(\mathbf{b}_x)$.

When the circuit C is composed of elements from $\mathcal{K}$, we can say more about the induced distributions $\mathcal{X}$ and $\mathcal{Z}$.

Lemma 5.3. *Consider a Clifford circuit of depth* 1 *composed of elements from* $\mathcal{K}$*. The induced total probabilities* $\mathcal{X}$*,* $\mathcal{Z}$ *are locally decaying distributions with failure rate* $\sqrt{p_{\text{phys}}}$*.*

Proof. We shall prove this statement for the distribution $\mathcal{X}$; the proof for the distribution $\mathcal{Z}$ is identical. Fix an arbitrary vector $\mathbf{b}_x \in \{0, 1\}^{\mathcal{W}}$.

Suppose a fault F results in some error $\mathbf{b}'_x$ such that $\text{supp}(\mathbf{b}'_x) \supseteq \text{supp}(\mathbf{b}_x)$. This implies that F must obey $\text{supp}(F) \supseteq \text{supp}(\mathbf{b}_x)$. Let F be the smallest set of fault locations such that $\text{supp}(F) \supseteq \text{supp}(\mathbf{b}_x)$. Because the circuit C has depth 1 and is composed entirely of only 1- and 2-qubit gates, this implies that $|F| \leq |\mathbf{b}_x| \leq 2|F|$.

By definition, the total probability of the fault F is $\mathcal{F}(F)$ a locally decaying distribution with failure rate p_{phys}.

$$\mathcal{X}(\mathbf{b}_x) \leq \mathcal{F}(F) \leq (p_{\text{phys}})^{|F|} \tag{35}$$

$$\leq (\sqrt{p_{\text{phys}}})^{|\mathbf{b}_x|} . \tag{36}$$

The result follows. ∎

We are now ready to study p_{round} and its dependence on $C_n^{\mathcal{Q}}$. To set the stage, we first consider ideal syndrome extraction in the absence of circuit faults. We focus our attention on the extraction of X-type syndromes and note that the analysis for the Z-type syndromes is identical.

Consider a corrupted code state $\mathsf{E}|\psi\rangle$ where ψ is a code state and $\mathsf{E} = \mathsf{X}(\mathbf{e}_x)\mathsf{Z}(\mathbf{e}_z)$ is some Pauli operator. If the syndrome-extraction circuit $C_n^{\mathcal{Q}}$ has no faults, the joint state of the data and ancilla qubits after the circuit is described by[14]

$$\mathsf{E}|\psi\rangle \otimes \mathsf{Z}(\boldsymbol{\sigma}_{\mathsf{X}})|+\rangle^{\otimes m_{\mathsf{X}}} \ , \tag{37}$$

where $\boldsymbol{\sigma}_{\mathsf{X}}$ represent the ideal syndromes for X-type stabilizer generators.

In this setting, we can use Equation (32) to update X- and Z-components of Pauli operators under the action of CNOT. Initially, the X and Z components of the state $\mathsf{E}|\psi\rangle \otimes |+\rangle^{\otimes m_{\mathsf{X}}}$ can be expressed as $(\mathbf{e}_x|\mathbf{0})$, $(\mathbf{e}_z|\mathbf{0})$, where $\mathbf{0}$ is the all zeros vector of length m_{X}. The vector $\mathbf{0}$ means that we assume that the input to the circuit is the state $\mathsf{E}|\psi\rangle \otimes |+\rangle^{\otimes m_{\mathsf{X}}}$; preparation faults on the ancilla occur in the first time step. For $1 < t < \Delta + 2$, we apply CNOT gates specified by a matrix $\mathbf{M}^{(t)} \in \mathbb{F}_2^{m_{\mathsf{X}} \times n}$. If we do not apply an entangling gate (i.e. when we SWAP qubits), then $\mathbf{M}^{(t)}$ is a matrix of zeros. Otherwise, the (i, j) entry of $\mathbf{M}^{(t)}$ is 1 if and only if the i^{th} syndrome qubit and the j^{th} data qubit are involved in a CNOT gate in the t^{th} time step. In the absence of circuit faults, the X components of the error $(\mathbf{e}_x|\mathbf{0})$ are left unaffected during the phase where we measure X-type syndromes. On the other hand, the Z-components transform as

$$\begin{pmatrix} \mathbf{e}_z \\ \mathbf{0} \end{pmatrix} \mapsto \begin{pmatrix} \mathbf{e}_z \\ \sum_t \mathbf{M}^{(t)} \cdot \mathbf{e}_z \end{pmatrix} \ . \tag{38}$$

The vector $\sum_t \mathbf{M}^{(t)} \cdot \mathbf{e}_z$ is the X-type syndrome $\boldsymbol{\sigma}_{\mathsf{X}}$. In other words, $\sum_t \mathbf{M}^{(t)} =: \mathbf{H}_{\mathsf{X}}$ is the symplectic representation of the X-type stabilizer generators. Note that $\mathbf{H}_{\mathsf{X}}$ is a sparse matrix with at most Δ_q ones per row and Δ_g ones per column.

Next, we move on to the setting where circuit components are faulty. The final state of the data and ancilla qubits is different from Equation (37) because of circuit faults. We express it as

$$(\mathsf{D} \otimes \mathsf{A}) \left(\mathsf{E}|\psi\rangle \otimes \mathsf{Z}(\boldsymbol{\sigma}_{\mathsf{X}})|+\rangle^{\otimes m_{\mathsf{X}}} \right) \ , \tag{39}$$

where, D and A represent errors due to faults in the circuit $C_n^{\mathcal{Q}}$ on the data qubits and ancilla qubits respectively. The symplectic representation of the final Pauli operator on the data and ancilla qubits is

$$\begin{pmatrix} \mathbf{e}_x + \mathbf{d}_x \\ \mathbf{a}_x \end{pmatrix} \ , \qquad \begin{pmatrix} \mathbf{e}_z + \mathbf{d}_z \\ \boldsymbol{\sigma}_{\mathsf{X}} + \mathbf{a}_z \end{pmatrix} \ . \tag{40}$$

The probability of errors per round, p_{round}, is the maximum failure rate for the distributions describing $\mathbf{d}_x, \mathbf{d}_z, \mathbf{a}_x$, and $\mathbf{a}_z$.

Theorem 5.4. *The induced distributions $\mathcal{X}$ and $\mathcal{Z}$ that govern the errors $\mathsf{D}\otimes\mathsf{A}$ are locally decaying distributions with failure rate p_{round}, where*

$$p_{\text{round}} \leq 2^{\Delta+1} \cdot \mathcal{T}(C_n^{\mathcal{Q}}) \cdot (p_{\text{phys}})^{1/(2\Delta+2)} \ ,$$

where $\Delta = \max(\Delta_q, \Delta_g)$ is the number of stages in the circuit $C_n^{\mathcal{Q}}$.

Proof. Recall that the circuit $C_n^{\mathcal{Q}}$ proceeds in two phases, with the first phase used to measure X-type syndromes and the second phase used to measure Z-type syndromes. For brevity, we allow $\mathcal{T}_{\mathsf{X}}$ and $\mathcal{T}_{\mathsf{Z}}$ to be the depth of the circuit $C_n^{\mathcal{Q}}$ corresponding to each phase; this means $\mathcal{T}(C_n^{\mathcal{Q}}) = \mathcal{T}_{\mathsf{X}} + \mathcal{T}_{\mathsf{Z}}$. Here, we focus on the first phase of $C_n^{\mathcal{Q}}$ which is used to measure X-type syndromes and study the evolution of Z-type errors; the proof of the remaining three cases is identical and for this reason we omit them.

Let $\mathbf{b}_x^{(t)}$, $\mathbf{b}_z^{(t)} \in \{0,1\}^n$ and $\mathbf{c}_x^{(t)}$, $\mathbf{c}_z^{(t)} \in \{0,1\}^{m_{\mathsf{X}}}$ be the errors on data and ancilla qubits induced by faults caused at time t. In turn, these errors can spread to other qubits and interact with errors

[14] The ancillas are disentangled from the data block by measurement.

at later times. Using Equation (32) repeatedly, we can write the final error $\mathbf{e}_z + \mathbf{d}_z$, $\boldsymbol{\sigma}_{\mathsf{X}} + \mathbf{a}_z$ in terms of the errors at each step as follows:

$$\begin{pmatrix} \mathbf{e}_z + \mathbf{d}_z \\ \boldsymbol{\sigma}_{\mathsf{X}} + \mathbf{a}_z \end{pmatrix} = \begin{pmatrix} \mathbf{e}_z + \sum_t \mathbf{b}_z^{(t)} \\ \sum_t \mathbf{M}^{(t)} \cdot \mathbf{e}_z + \sum_t \mathbf{M}^{(t)} \cdot \sum_{t'<t} \mathbf{b}_z^{(t')} + \sum_t \mathbf{c}_z^{(t)} \end{pmatrix} . \tag{41}$$

As we are only measuring X-type syndromes, all sums are over time steps t in the first phase of the circuit. For time steps t where we do not apply a CNOT, all entries of $\mathbf{M}^{(t)}$ are 0.

We can simplify Equation (41) by eliminating $\mathbf{e}_z$ and $\boldsymbol{\sigma}_{\mathsf{X}} = \sum_t \mathbf{M}^{(t)} \cdot \mathbf{e}_z$:

$$\begin{pmatrix} \mathbf{d}_z \\ \mathbf{a}_z \end{pmatrix} = \begin{pmatrix} \sum_t \mathbf{b}_z^{(t)} \\ \sum_t \mathbf{M}^{(t)} \cdot \sum_{t'<t} \mathbf{b}_z^{(t')} + \sum_t \mathbf{c}_z^{(t)} \end{pmatrix} . \tag{42}$$

While it is a straightforward consequence of the linear evolution under symplectic transformations, being able to write $\mathbf{d}_z$ and $\mathbf{a}_z$ without $\mathbf{e}_x$ and $\mathbf{e}_z$ means that the Z-components of the errors $\mathbf{d}_z$ and $\mathbf{a}_z$ do not depend on the input error E.

Furthermore, the special structure of the syndrome-extraction circuit is reflected here — $\mathbf{d}_z$ is simply the sum of the errors $\mathbf{b}_z^{(t)}$ caused by faulty gates at each step. In other words, Z errors on data qubits are not affected by Z errors on ancilla qubits.

We simplify this further using two observations. First, we will find it useful to reorder the sums within this equation as follows:

$$\sum_t \mathbf{M}^{(t)} \cdot \sum_{t'<t} \mathbf{b}_z^{(t')} = \sum_t \sum_{t'} \mathbb{1}[t' < t]\, \mathbf{M}^{(t)} \cdot \mathbf{b}_z^{(t')} = \sum_{t'} \left(\sum_{t>t'} \mathbf{M}^{(t)} \right) \cdot \mathbf{b}_z^{(t')} . \tag{43}$$

where $\mathbb{1}[t' < t]$ is the indicator function, i.e. it is 1 when $t' < t$ and 0 otherwise. The terms on the right-hand sides of these equations have a natural interpretation — for example, the error $\mathbf{b}_z^{(t')}$ that occurs on data qubits at time t' can propagate to ancilla qubits at times $t > t'$.

Second, it is difficult to directly deal with sums of random vectors modulo 2 that appear in Equation (42). Instead, we re-write Equation (42) in terms of the *support* of the vectors. To this end, we note that

$$\operatorname{supp}\left(\sum_{t'>t} \mathbf{M}^{(t')} \right) \subseteq \operatorname{supp}\left(\sum_t \mathbf{M}^{(t)} \right) = \operatorname{supp}(\mathbf{H}_{\mathsf{X}}) . \tag{44}$$

Together, these observations mean we can rewrite Equation (42) as

$$\operatorname{supp}\begin{pmatrix} \mathbf{d}_z \\ \mathbf{a}_z \end{pmatrix} \subseteq \operatorname{supp}\begin{pmatrix} \sum_t \mathbf{b}_z^{(t)} \\ \mathbf{H}_{\mathsf{X}} \cdot \sum_t \mathbf{b}_z^{(t)} + \sum_t \mathbf{c}_z^{(t)} \end{pmatrix} \tag{45}$$

$$\subseteq \bigcup_t \operatorname{supp}\begin{pmatrix} \mathbf{b}_z^{(t)} \\ \mathbf{H}_{\mathsf{X}} \cdot \mathbf{b}_z^{(t)} + \mathbf{c}_z^{(t)} \end{pmatrix} \tag{46}$$

$$\subseteq \bigcup_t \operatorname{supp}\left[\begin{pmatrix} \mathbf{I}_n & \mathbf{0} \\ \mathbf{H}_{\mathsf{X}} & \mathbf{I}_{m_{\mathsf{X}}} \end{pmatrix} \begin{pmatrix} \mathbf{b}_z^{(t)} \\ \mathbf{c}_z^{(t)} \end{pmatrix} \right] . \tag{47}$$

where $\mathbf{I}_n$ and $\mathbf{I}_{m_{\mathsf{X}}}$ are identity matrices of dimensions n and m_{X} respectively. We pause to explain the two simplifications in words. Substituting $\sum_{t'>t} \mathbf{M}^{(t')}$ with $\mathbf{H}_{\mathsf{X}}$ corresponds to a worst-case

setting—an error on an ancilla qubit can propagate to *all* data qubits in its support *regardless* of when the error on the ancilla qubit occurs. Second, by dealing with the union of the supports of the vectors instead of the vectors themselves, we upper bound the maximum size of the final error. Evaluating the probability of this event allows us to upper bound the probability of a final error $\mathsf{D} \otimes \mathsf{A}$.

We can bound the probabilities of the terms in Equation (47). The errors at time t,

$$\begin{pmatrix} \mathbf{b}_z^{(t)} \\ \mathbf{c}_z^{(t)} \end{pmatrix} ,$$

are independent of errors occurring at $t' \neq t$ — induced errors at different time steps are independent because faults occurring at different time steps are independent. As shown in Lemma 5.3, the induced distributions over errors at each time step are locally decaying distributions with failure rate $\sqrt{p_{\text{phys}}}$.

Next, Lemma 5.2 describes how the distribution is transformed when errors undergo linear transformations. Consider the terms in Equation (47):

$$\begin{pmatrix} \mathbf{I}_n & \mathbf{0} \\ \mathbf{H}_\mathsf{X} & \mathbf{I}_{m_\mathsf{X}} \end{pmatrix} \begin{pmatrix} \mathbf{b}_z^{(t)} \\ \mathbf{c}_z^{(t)} \end{pmatrix} , \tag{48}$$

$\mathbf{H}_\mathsf{X}$ has row and column weight at most Δ, so the block matrix that appears in Equation (48) has column weight at most $\Delta + 1$. By Lemma 5.2, each term in the union is distributed according to a locally decaying distribution with failure rate $2^{\Delta+1}(p_{\text{phys}})^{1/2(\Delta+1)}$.

Finally, Lemma 5.1 allows us to bound the failure rate of the compositions of independent locally decaying distributions. This, in turn, is an upper bound on the rate of the locally decaying distribution $\mathcal{Z}$ over $\mathbf{d}_z, \mathbf{a}_z$. The union extends over the depth $\mathcal{T}_\mathsf{X}$ of the circuit required to measure X-type syndromes terms. Applying Lemma 5.1 repeatedly, we find $\mathcal{Z}$ is a locally decaying distribution with failure rate

$$2^{\Delta+1} \cdot \mathcal{T}_\mathsf{X} \cdot (p_{\text{phys}})^{1/2(\Delta+1)} . \tag{49}$$

By an identical argument, the X errors are distributed according to a $2^{\Delta+1} \cdot \mathcal{T}_\mathsf{X} \cdot p_{\text{phys}}^{1/2(\Delta+1)}$ locally decaying distribution. In turn, this means that the induced distributions $\mathcal{X}$ and $\mathcal{Z}$ are locally decaying distributions with failure rate $2^{\Delta+1} \cdot \mathcal{T}_\mathsf{X} \cdot (p_{\text{phys}})^{1/2(\Delta+1)}$.

Repeating the same analysis for the Z-type syndrome measurements, we find that the $\mathcal{X}$ and $\mathcal{Z}$ distributions describing induced errors are locally decaying distributions with failure rate

$$2^{\Delta+1} \cdot \mathcal{T}_\mathsf{Z} \cdot (p_{\text{phys}})^{1/2(\Delta+1)} . \tag{50}$$

We can use Lemma 5.1 again to bound the failure rate per for the entire circuit $C_n^\mathcal{Q}$. As $\mathcal{T}(C_n^\mathcal{Q}) = \mathcal{T}_\mathsf{X} + \mathcal{T}_\mathsf{Z}$, we arrive at the result that $\mathcal{X}$ and $\mathcal{Z}$ are locally decaying distributions with failure rate p_{round} where

$$p_{\text{round}} = 2^{\Delta+1} \cdot \mathcal{T}(C_n^\mathcal{Q}) \cdot (p_{\text{phys}})^{1/2(\Delta+1)} . \tag{51}$$

■

When qubits are arranged on an $L \times L$ lattice, the circuit depth $\mathcal{T}(C_n^\mathcal{Q})$ is $O(\sqrt{n}/R)$. If gates are constrained by geometric locality, i.e. $R = \omega(L)$, then the depth of the circuit $C_n^\mathcal{Q}$ grows with the code size n. However, for the existence of a threshold, we require p_{round} to be some fixed constant. We therefore only achieve a threshold if the physical failure probability vanishes as the size of the code increases:

$$p_{\text{phys}} = O\left[\left(\frac{1}{\mathcal{T}(C_n^\mathcal{Q})} \right)^{2(\Delta+1)} \right] . \tag{52}$$

However, if we were to use a concatenated construction, where the outer code is the constant-rate LDPC code $\mathcal{Q}_n$ and the inner code is a surface code $\mathcal{RS}_\ell$, then we can choose p_{phys} to decrease *exponentially* with the size of the inner code. We study this in the next section.

Finally, we comment that the factor $2^{\Delta+1}$ that appears in Theorem 5.4 can very likely be reduced. However, this particular version of the theorem is sufficient for our purposes, namely to prove the existence of a threshold for the hierarchical scheme. For readers interested in applying the hierarchical scheme to the real world, we shall estimate the logical failure rate of the hierarchical scheme numerically in Section 6.

5.2 Coarse graining concatenated circuits

In the next two sections, we will analyze the concatenated code by applying Gottesman's theorem described in Section 2.4 to both the inner code and the outer code. In this section, we apply it to the inner code; for $\mathcal{W} = \mathcal{W}(C_n^{\mathcal{Q}})$, the inner code $\mathcal{RS}_\ell^{\otimes \mathcal{W}}$ is itself an LDPC code. In Section 5.3, we will apply Gottesman's theorem to the outer code.

In Section 2.3, we described how we cannot ignore the details of the Level-0 syndrome-extraction circuit in a concatenated code. In this section, we show that if logical gates on surface codes are performed as described in Section 4, then they are fault tolerant. We show the existence of a threshold $q_{\text{phys}}^{(0)}$ such that if the failure rate per round is below $q_{\text{phys}}^{(0)}$, then we can directly study Level-1 operations and ignore Level-0 operations.

Consider an input state $\rho_{\text{in}} \in (\mathcal{RS}_\ell)^{\otimes \mathcal{W}}$ in the bilayer architecture. Let Level-0 faults on the syndrome-extraction circuit be distributed according to a locally decaying distribution with failure rate $p_{\text{phys}}^{(0)}$.

The failure rate per round on the data qubits and the syndrome qubits is the same because data and syndrome qubits both interact with 4 other qubits. Let $q_{\text{in}}^{(0)}$, $q_{\text{round}}^{(0)}$ be the thresholds for surface code error correction as defined in Section 2.5. Suppose we are below threshold. Then after error correction, tiles that have not failed are described by a locally decaying Level-0 error model with failure rate $p_{\text{round}}^{(0)}$. Theorem 5.4 guarantees that the failure rate per round grows with the depth of the syndrome-extraction circuit; it also relies on the degree of the qubits and stabilizer generators. If we measure X and Z syndromes separately, the depth of the syndrome-extraction circuit is at most 12. The degree of the qubits and stabilizers is 4. Using Theorem 5.4, we can bound the failure rate per round of surface code syndrome extraction:[15]

$$p_{\text{round}}^{(0)} < 384 \left(p_{\text{phys}}^{(0)}\right)^{1/10} . \tag{53}$$

This bound can be much better—for example X and Z syndromes can be measured in parallel which, in turn, can reduce the depth of the circuit; we can also likely reduce the constant 384 in front of p_{round}. However, we continue to use the bound in Equation (53) for simplicity.

We can use Theorem 5.4 to show that the logical operations for the bilayer architecture are fault tolerant. We argue that both the Level-0 and Level-1 failure rates after the operation are constant.

Theorem 5.5. *Let C be the circuit on a state $\rho_{\text{in}} \in \mathcal{RS}_\ell^{\otimes \mathcal{W}}$ such that each tile is involved in at most one logical gate in $\mathcal{K}_1$. Tiles that have not suffered a logical error are described by a locally decaying error with Level-0 input failure rate $p_{\text{round}}^{(0)}$.*

There exists a threshold $q_{\text{phys}}^{(0)}$ such that, if $p_{\text{phys}}^{(0)} \leq q_{\text{phys}}^{(0)}$, then

1. *the circuit C is described by a Level-1 locally decaying faults model with Level-1 failure rate $p_{\text{phys}}^{(1)} := \exp(-c_{\text{EC}} \cdot \ell)$.*
2. *the output is described by a Level-0 locally decaying errors model with failure rate less than $p_{\text{round}}^{(0)}$.*

[15] The constant $2^{\Delta+1} = 32$ and $32 \times 12 = 384$.

Proof. Let $\rho_{\text{in}} \in \mathcal{RS}_\ell^{\otimes \mathcal{W}}$ be a noisy code state with Level-0 errors described by a locally decaying distribution with failure rate $p^{(0)}_{\text{round}}$.

For sufficiently low logical failure rate, we can use Gottesman's result presented in Section 2.4 to bound the failure rate for error correction and to show that after error correction, the Level-0 errors are locally decaying distributions with failure rate $p^{(0)}_{\text{round}}$.

State preparation: Suppose we wished to prepare the state $|\overline{0}\rangle^{\otimes m}$ for the code $\mathcal{RS}_\ell^{\otimes m}$. Each Level-0 qubit is prepared in $|0\rangle$ and we then perform the syndrome-extraction circuit for the surface code on all m copies. The Level-0 errors are described by a locally decaying error model with failure rate $p^{(0)}_{\text{in}} = p^{(0)}_{\text{phys}}$. The faults in the syndrome-extraction circuit C are also described by a locally decaying faults model with failure rate $p^{(0)}_{\text{phys}}$. Error correction is successful if

$$p^{(0)}_{\text{phys}} < q^{(0)}_{\text{in}} \ , \qquad p^{(0)}_{\text{round}} < q^{(0)}_{\text{round}} \ . \tag{54}$$

Entangling gates: Entangling gates between data and ancilla blocks are performed in a transversal manner. Errors due to faults in the transversal gate are distributed according to a locally decaying distribution with failure rate $p^{(0)}_{\text{phys}}$. Lemma 5.1 shows that the input to error correction is a state with Level-0 errors described by a locally decaying distribution with failure rate $p^{(0)}_{\text{round}} + p^{(0)}_{\text{phys}}$. Error correction is successful if

$$p^{(0)}_{\text{round}} + p^{(0)}_{\text{phys}} < q^{(0)}_{\text{in}} \ , \qquad p^{(0)}_{\text{round}} < q^{(0)}_{\text{round}} \ . \tag{55}$$

SWAP gates: Assume that the Level-0 failure rate is $p^{(0)}_{\text{round}}$. The logical SWAP operation is decomposed entirely in terms of physical SWAP operations. As these are non-entangling operations, the error distribution is a locally decaying distribution with failure rate $\sqrt{p^{(0)}_{\text{phys}}}$. We can use Lemma 5.1 to find the effective failure rate per round. This is equal to the sum of the failure rate per round of syndrome extraction and the failure rate of the SWAP gate itself. Note that because the SWAP gate has larger depth, we perform more than d_ℓ rounds of syndrome extraction.

Therefore, the failure rate per round on both data and ancilla qubits is $p^{(0)}_{\text{round}} + \sqrt{p^{(0)}_{\text{phys}}}$. Error correction is successful if

$$p^{(0)}_{\text{round}} < q^{(0)}_{\text{in}} \ , \qquad p^{(0)}_{\text{round}} + \sqrt{p^{(0)}_{\text{phys}}} < q^{(0)}_{\text{round}} \ . \tag{56}$$

Logical measurement of Pauli operators:

We will wish to measure logical operators on tiles that represent Level-1 ancilla qubits. Consider a state with Level-0 errors distributed according to a locally decaying distribution with failure rate $p^{(0)}_{\text{round}}$. We first study the logical measurement of a single tile.

To destructively measure the logical X (Z) operator on a single tile, we can measure each of the physical qubits in the X (Z) basis. This is permitted by our available operations in $\mathcal{K}_0$. Faults on measurements are distributed according to a locally decaying distribution with rate $p^{(0)}_{\text{phys}}$. The resulting distribution on the output bits is a locally decaying distribution with rate $p^{(0)}_{\text{round}} + p^{(0)}_{\text{phys}}$. We can use each of the individual Level-0 qubit outputs to infer the values of each of the X-type (Z-type) stabilizer generators and correct Z (X) errors. This fails with probability $\exp(-c_{EC} \cdot \ell)$.

We can now study all tiles that undergo measurement. As measurements on each tile are performed separately, this induces a Level-1 measurement error with probability $\exp(-c_{\text{EC}} \cdot \ell)$.

In the mean time, tiles that represent data qubits remain idle for 1 time step. As we assume idle errors are distributed according to a locally decaying distribution with failure rate $p^{(0)}_{\text{phys}}$, the Level-0 error rates on these tiles are $p^{(0)}_{\text{round}} + p^{(0)}_{\text{phys}}$. Error correction is successful if

$$p^{(0)}_{\text{round}} + p^{(0)}_{\text{phys}} < q^{(0)}_{\text{in}} \ . \tag{57}$$

Combining requirements for all operations: We can use Equation (53) to state $p^{(0)}_{\text{round}} < 384(p^{(0)}_{\text{phys}})^{1/10}$ and note that both $p^{(0)}_{\text{phys}}$ and $\sqrt{p^{(0)}_{\text{phys}}}$ are less than $(p^{(0)}_{\text{phys}})^{1/10}$. Therefore, we can define the threshold $q^{(0)}_{\text{phys}}$ using the bounds in Equations (54), Equation (55), Equation (56) and Equation (57):

$$q^{(0)}_{\text{phys}} = \min\left[\left(\frac{q^{(0)}_{\text{in}}}{385}\right)^{10}, \left(\frac{q^{(0)}_{\text{round}}}{385}\right)^{10}\right] . \tag{58}$$

Below threshold, we can invoke Gottesman's result to guarantee error suppression; we obtain a logical failure rate $p^{(1)}_{\text{phys}} = \exp(-c_{EC} \cdot \ell)$.

■

5.3 The syndrome-extraction circuit $C^{\mathcal{H}}_N$ has a threshold

In this section, we prove that the hierarchical code $\mathcal{H}_N$ has a threshold if we measure syndromes using the circuit $C^{\mathcal{H}}_N$. We review the construction first and the corresponding assumptions on failure rates. Thus far, we have simply stated the relationship between ℓ and n, i.e. that $\ell = \Theta(\log(n))$, without justification. We show in Lemma 5.6 that letting the inner code have size $\ell = \Theta(\log(n))$ is indeed sufficient to achieve arbitrarily small, but constant, Level-1 failure rate per round $p^{(1)}_{\text{round}}$. We bring these elements together in Theorem 5.7 to show that the hierarchical construction has a threshold.

Recall that the hierarchical code $\mathcal{H}_N$ is constructed by concatenating an outer $[\![n, k, d, \Delta_q, \Delta_g]\!]$ constant-rate LDPC code $\{\mathcal{Q}_n\}$ and inner $[\![d^2_\ell, 1, d_\ell]\!]$ code $\mathcal{RS}_\ell$. The family $\mathcal{Q}_n$ has parameters $k = \rho \cdot n$ for $\rho > 0$ and distance $d = \Theta(n^\delta)$ for $\delta > 0$.

Suppose we are given an input state of the concatenated code $\mathcal{H}_N$ subject to the following error model:

1. Errors on the state are distributed according to locally decaying distributions:
   (a) on Level-0 with failure rate $p^{(0)}_{\text{in}}$, and
   (b) on Level-1 with failure rate $p^{(1)}_{\text{in}}$.
2. Level-0 faults in the circuit are distributed according to a locally decaying distribution with failure rate $p^{(0)}_{\text{phys}}$.

Let $p^{(0)}_{\text{round}}$ denote the failure rate per round of syndrome extraction for the rotated surface code. As the depth of the syndrome-extraction circuit for the rotated surface code is constant, for fixed values of $p^{(0)}_{\text{phys}}$, $p^{(0)}_{\text{round}}$ is also a constant (See Equation (53)). We assume $p^{(0)}_{\text{in}} \leq p^{(0)}_{\text{round}}$ because it will make the following statements easier.

Qubits are laid out on a bilayer architecture as described in Section 4. Physical qubits are aggregated to form $\mathcal{W}(C^{\mathcal{Q}}_n)$ rotated surface codes $\mathcal{RS}_\ell$; these form $2L^2$ tiles where L is the smallest integer satisfying $2L^2 \geq \mathcal{W}(C^{\mathcal{Q}}_n)$.

The product code $\mathcal{RS}^{\otimes \mathcal{W}}_\ell$ is itself an LDPC code. The tiles will be used to simulate long-range entangling gates required to perform the syndrome-extraction circuit $C^{\mathcal{Q}}_n$ for the outer code. Single-tile preparation and measurement, and two-tile entangling gates are described in Section 4.2; Level-1 SWAP gates and permutations of tiles were described in Section 4.3. Recall that $q^{(0)}_{\text{phys}} \in (0, 1]$ was defined in Section 5.2. Per Theorem 5.5, if the input state has Level-0 errors described by a locally decaying distribution with failure rate $p^{(0)}_{\text{round}}$ and $p^{(0)}_{\text{phys}} < q^{(0)}_{\text{phys}}$, Level-1 circuit faults are distributed according to a locally decaying distribution with failure rate $p^{(1)}_{\text{phys}}$ and Level-0 residual errors are described by a locally decaying distribution with failure rate $p^{(0)}_{\text{round}}$ on tiles that have

not failed. This result allows us to coarse grain the Level-0 circuit and study Level-1 errors and faults directly.

For the outer code to have a threshold, we require that the $[\![n, k, d, \Delta_q, \Delta_g]\!]$ LDPC code family $\{\mathcal{Q}_n\}$ has a syndrome-extraction circuit such that $p^{(1)}_{\text{round}}$ remains a sufficiently small constant as discussed in Section 2.4. In the following lemma, we show that $\ell = \Theta(\log(n))$ is sufficient to achieve this.

Lemma 5.6. *Suppose Level-1 faults on the syndrome-extraction circuit $C^{\mathcal{H}}_N$ are distributed according to a locally decaying distribution with failure rate $p^{(1)}_{\text{phys}}$. Then, for arbitarily small constant $\varepsilon > 0$, $p^{(1)}_{\text{round}} < \varepsilon$ can be achieved using $\ell = \Theta(\log(n))$.*

Proof. From Theorem 5.4, the failure rate per round scales as

$$p^{(1)}_{\text{round}} = 2^{\Delta+1} \cdot \mathcal{T}(C^{\mathcal{Q}}_n) \cdot \left(p^{(1)}_{\text{phys}}\right)^{1/2(\Delta+1)} , \tag{59}$$

where $\Delta = \max(\Delta_q, \Delta_g)$ is some constant for a fixed family $\mathcal{Q}_n$. We want $p^{(1)}_{\text{round}}$ to be an arbitrarily small constant $\varepsilon > 0$. Per Theorem 5.5, Level-1 faults are distributed according to a locally decaying distribution with failure rate which implies that $p^{(1)}_{\text{phys}} = \exp(-c_{\text{EC}} \cdot \ell)$. From Section 4.1, $\mathcal{T}(C^{\mathcal{Q}}_n) = O(L/R) = O(\sqrt{n}/R)$. Therefore, the upper bound on $p^{(1)}_{\text{phys}}$ can be satisfied by choosing $\ell = \Theta(\log(n))$. ∎

In particular, there exists a threshold $q^{(1)}_{\text{round}}$ for the outer code $\mathcal{Q}_n$ for the syndrome-extraction circuit $C^{\mathcal{Q}}_n$. This can be achieved for some ℓ such that $\ell = \Theta(\log(n))$.

Theorem 5.7. *There exists a choice of ℓ such that $\ell = \Theta(\log(n))$ and thresholds $q^{(0)}_{\text{in}}$, $q^{(0)}_{\text{phys}}$ and $q^{(1)}_{\text{in}}$ such that if*

$$\max\left(p^{(0)}_{\text{in}}, p^{(0)}_{\text{round}}\right) < q^{(0)}_{\text{in}} , \qquad p^{(0)}_{\text{phys}} < q^{(0)}_{\text{phys}} , \qquad p^{(1)}_{\text{in}} < q^{(1)}_{\text{in}} ,$$

then the following is true. With probability at least $1 - p_{\mathcal{H}}(N)$, the state after error correction is correctable by an ideal decoder where, for some positive number $c_{\mathcal{H}}$ that is independent of N,

$$p_{\mathcal{H}}(N) < \exp\left(-c_{\mathcal{H}} \cdot \frac{N^{\delta}}{\log^{2\delta}(N)}\right) .$$

Furthermore, the residual errors are distributed according to a locally decaying distribution with failure rates $p^{(1)}_{\text{round}}$ and $p^{(0)}_{\text{round}}$ on Level-1 and Level-0 respectively.

Proof. Suppose $p^{(0)}_{\text{in}} < q^{(0)}_{\text{in}}$ and $p^{(0)}_{\text{phys}} < q^{(0)}_{\text{phys}}$. By definition, this is sufficient to perform Level-1 logical gates and surface code error correction as per Theorem 5.5.

Next, the LDPC code has thresholds $q^{(1)}_{\text{in}}$ and $q^{(1)}_{\text{round}}$ (See Section 2.4). The input state has Level-1 errors described by a locally decaying distribution with failure rate $p^{(1)}_{\text{in}}$. For the syndrome-extraction circuit on the outer LDPC code to be successful, we require $p^{(1)}_{\text{in}} < q^{(1)}_{\text{in}}$.

Finally, we require that the Level-1 failure rate per round is below the corresponding threshold

$$p^{(1)}_{\text{round}} < q^{(1)}_{\text{round}} . \tag{60}$$

From Lemma 5.6, this can be achieved using $\ell = \Theta(\log(n))$.

By definition, syndrome-extraction is successful if the ideal decoder $\mathcal{R}_{\mathcal{H}}$ is able to recover the final state. The code $\mathcal{H}_N$ fails if the outer LDPC code $\mathcal{Q}_n$ fails, i.e. the probability of failure is $p_{\mathcal{H}}(N) = p_{\mathcal{Q}}(n) = \exp(-c_{\mathcal{Q}} \cdot d(n)) = \exp(-c_{\mathcal{Q}} \cdot \Theta(n^{\delta}))$.

Using Equation (11), we can express the probability of failure $p_{\mathcal{H}}(N)$ in terms of N:

$$p_{\mathcal{H}}(N) < \exp\left(-c_{\mathcal{H}} \cdot \frac{N^{\delta}}{\log^{2\delta}(N)}\right) . \tag{61}$$

for some positive number $c_{\mathcal{H}}$ that is independent of N.

Residual errors are distributed according to locally decaying distribution:

1. on Level-1 with failure rate $p^{(1)}_{\text{round}}$; this is guaranteed by Gottesman's result applied to the outer code.
2. on Level-0 errors failure rate $p^{(0)}_{\text{round}}$; This is guaranteed by Theorem 5.5.

The result follows. ■

We reiterate that $p_{\mathcal{H}}(N)$ is an upper bound on the failure rate for the Level-2 error probability distribution.

This analysis depends crucially on the failure rate of the SWAP gates; $p^{(1)}_{\text{round}}$, and therefore the size of the inner code, scales with the depth of the circuit because of noisy SWAP operations. In proving Theorem 5.7, we were agnostic to the failure modes in the circuit and assumed that all Level-1 two-qubit gates fail with probability $p^{(1)}_{\text{phys}}$. However, if the fidelity of physical SWAP gates can be improved over the fidelity of entangling gates, this can reduce the overhead for the hierarchical scheme significantly. We provide evidence for this in Section 6.3 where we estimate the logical failure rate for the hierarchical scheme. In certain architectures such as trapped neutral atoms, SWAP gates can be performed by physically moving the trap [Blu+22]. In this case, the failure rate for the SWAP gates may have no direct connection to the failure rate for CNOT operations.

6 Comparisons with the basic encoding

We have shown that the hierarchical code $\{\mathcal{H}_N\}$ has a syndrome-extraction circuit that can be constructed using gates restricted by geometric locality such that it has a threshold. Below threshold, the WER is suppressed superpolynomially, but subexponentially in N. It is natural to ask whether the resources spent in performing SWAP gates can be better spent simply building a more robust surface code. In this section, we consider the basic encoding $\mathcal{B}_M$ which encodes K logical qubits in surface codes $\mathcal{RS}_{\ell_M}$. We compare the hierarchical scheme and the basic enconding in different ways.

We show in Section 6.1 that for a target WER, the syndrome-extraction circuit for the hierarchical memory is more efficient than the syndrome-extraction circuit for the basic encoding. This is measured by the depth and width of the corresponding circuits. We will state and prove a formal version of Theorem 1.4. Depending on the value of the threshold for the outer LDPC codes however, it is not immediately obvious that this scaling manifests for practical block lengths. In the rest of this section, we present numerical estimates for the WER $p_{\mathcal{H}}(N)$ of the hierarchical memory and contrast it with the WER $p_{\mathcal{B}}(M)$ for the basic encoding. We do this by demanding a fixed total number of qubits for both schemes and compare $p_{\mathcal{H}}(N)$ and $p_{\mathcal{B}}(M)$. We demonstrate that there is a *crossover point*, i.e. a value of the physical error rate where, for fixed total number of qubits, the hierarchical memory outperforms the basic encoding, i.e. $p_{\mathcal{H}}(N) < p_{\mathcal{B}}(M)$. In our estimates, this happens at gate error rates roughly between 10^{-3} and 10^{-4}. While these are preliminary estimates, they are promising nonetheless as they are in the realm of possibility.

In Section 6.2, we briefly discuss the codes we use as outer and inner codes. To estimate the crossover point, we make some assumptions about the noise model, gates, and decoder. Owing to these assumptions, our estimates should only be interpreted as a proof-of-principle that the overhead of the hierarchical scheme pays off in a reasonable parameter regime. In Section 6.3, we present the results of our simulations. All together we believe these assumptions, especially those

related to the decoder, code, and noise model, are conservative. We return to these assumptions in Section 6.4 and for each assumption, we outline how one might expect it to change (1) in the future, and (2) in a more realistic setting. In general, we expect that with careful engineering (e.g. high-rate linear-distance codes, architecture-specific considerations, improved decoding algorithms) and more realistic noise modeling (e.g. including significant error correlations), the cross-over to when hierarchical memories outperform surface codes will occur at smaller numbers of logical qubits, higher physical error rates, and higher target WERs than in our estimates.

6.1 Asymptotic comparison with surface code

We have proved the existence of a threshold when we simulate an LDPC code using local gates. However, the existence of a threshold alone might not warrant switching over to a different scheme when there already exists an excellent local scheme — the surface code. We recall that we are only constructing a quantum memory, and not a scheme for universal, fault-tolerant quantum computation. In this section, we ask how the surface code would perform if we used the same total number of qubits used in the concatenated scheme above to plainly encode all logical qubits. We find that there is a space-time tradeoff to implementing a hierarchical scheme.

The hierarchical scheme $\{\mathcal{H}_N\}$ with corresponding fault tolerant syndrome-extraction circuits $\{C_N^{\mathcal{H}}\}$ achieves the following costs:

$$\mathcal{W}(C_N^{\mathcal{H}}) = \Theta(N) \qquad \mathcal{T}(C_N^{\mathcal{H}}) = O\left(\frac{\sqrt{N}}{R}\right) . \tag{62}$$

This family encodes $k(n) = \rho \cdot n$ qubits. Note that the depth is for a *single* round of syndrome extraction. We will return to this point shortly.

Consider the *basic encoding* defined by the family $\{\mathcal{B}_M\}$ where $M = \Theta(k \cdot \ell_M^2)$, and $\mathcal{B}_M = \bigotimes_{i=1}^{k} \mathcal{RS}_{\ell_M}$ is a k-fold product of rotated surface codes $\mathcal{RS}_{\ell_M}$. Each code $\mathcal{RS}_{\ell_M}$ has distance $d_M = \Theta(\ell_M)$. Let the corresponding circuits be denoted $\{C_M^{\mathcal{B}}\}$.

To compare with $\mathcal{H}_N$, we probe the parameters of $C_M^{\mathcal{B}}$ required to guarantee the same logical error suppression. Let $p_{\mathcal{B}}(M)$ denote the failure rate for the Level-1 logical probability of failure for $\mathcal{B}_M$—we declare failure if any of the k logical qubits fail.

We assume that the Level-0 physical failure rates are sufficiently below threshold to perform surface code error correction, i.e. $p_{\text{phys}}^{(0)} < q_{\text{phys}}^{(0)}$. Furthermore, we also assume that the code state $\mathcal{B}_M$ is prepared such that there are no input errors, i.e. $p_{\text{in}}^{(1)} = p_{\text{in}}^{(0)} = 0$. This allows us to isolate the rate of error suppression because of error correction.

Lemma 6.1. *Let $\{\mathcal{B}_M\}$ be the basic encoding such that $p_{\mathcal{B}}(M) < \exp(-c_{\mathcal{H}} \cdot N^{\delta}/\log(N)^{2\delta})$ where $c_{\mathcal{H}}$ is a positive constant. Then*

$$\mathcal{W}(C_M^{\mathcal{B}}) = \Omega\left[\left(\frac{N}{\log(N)}\right)^{1+2\delta}\right] , \qquad \mathcal{T}(C_M^{\mathcal{B}}) = \Omega\left[\left(\frac{N}{\log^2(N)}\right)^{\delta}\right] .$$

Proof. By assumption, $p_{\text{in}}^{(1)} = p_{\text{in}}^{(0)} = 0$ and $p_{\text{phys}}^{(0)} < q_{\text{phys}}^{(0)}$ and therefore we can perform error correction. The Level-1 logical failure probability for the code $\mathcal{RS}_{\ell_M}^{\otimes k}$ is described by a locally-decaying error model with failure rate $p_{\mathcal{RS}}(\ell_M)$ (See Section 2.5), where

$$p_{\mathcal{RS}}(\ell_M) = \exp(-c_{\text{EC}} \cdot \ell_M) . \tag{63}$$

We declare failure if any of the k tiles of $\mathcal{B}_M$ fails, which implies that

$$\begin{array}{ccccc} p_{\mathcal{RS}}(\ell) & \leq & p_{\mathcal{B}}(M) & \leq & 1-(1-p_{\mathcal{RS}}(\ell))^k \\ \exp(-c_{\text{EC}} \cdot \ell_M) & \leq & p_{\mathcal{B}}(M) & \leq & n \cdot \exp(-c_{\text{EC}} \cdot \ell_M) \end{array} . \tag{64}$$

To guarantee that the error rate $p_{\mathcal{B}}(M)$ is lower than $p_{\mathcal{H}}(N)$, we at least require that

$$\exp(-c_{\mathrm{EC}} \cdot \ell_M) \leq \exp\left(-c_{\mathcal{H}} \cdot \frac{N^\delta}{\log(N)^{2\delta}}\right) . \tag{65}$$

This implies that $\ell_M = \Omega(N^\delta / \log(N)^{2\delta})$.

We can now compute the space and depth requirements for $C_M^{\mathcal{B}}$. The space cost $\mathcal{W}(C_M^{\mathcal{B}})$ is $\Theta(k \cdot \ell_M^2)$. The hierarchical memory $\mathcal{H}_N$ uses a constant-rate $[\![n, k, d, \Delta_q, \Delta_g]\!]$ quantum LDPC code $\mathcal{Q}_n$ where $k(n) = \rho \cdot n$ and $d(n) = \Theta(n^\delta)$. This implies that

$$\mathcal{W}(C_M^{\mathcal{B}}) = \Omega\left[\left(\frac{N}{\log(N)}\right)^{1+2\delta}\right] . \tag{66}$$

Furthermore, each tile requires ℓ_M rounds of error correction; syndrome-extraction circuits on separate tiles can be run in parallel. Therefore

$$\mathcal{T}(C_M^{\mathcal{B}}) = \Theta(\ell_M) = \Omega(N^\delta / \log^{2\delta}(N)) . \tag{67}$$

This completes the proof. ■

Comparing with Equation (62), the basic encoding requires a larger space overhead for all $\delta > 0$:

$$\frac{\mathcal{W}(C_M^{\mathcal{B}})}{\mathcal{W}(C_N^{\mathcal{H}})} = \Omega\left[\frac{N^{2\delta}}{\log^{1+2\delta}(N)}\right] . \tag{68}$$

As stated, however, the time overhead is worse. Although the depth of the syndrome-extraction circuit $C_N^{\mathcal{H}}$ is $O(\sqrt{N}/R)$, we will need to perform $d(n)$ rounds of syndrome extraction to be fault tolerant. However, this is not a fundamental requirement; it is due to the nature of Gottesman's proposal in [Got13] which uses an inefficient minimum-weight decoder. There exist constant-rate LDPC codes that possess efficient, single-shot decoding algorithms, i.e. syndrome extraction only needs to be performed a constant number of times for the decoding algorithm to work [LTZ15; FGL18a; FGL18b]; furthermore the algorithm requires $O(N)$ time. For such codes, we can compare the depth of the syndrome-extraction circuit $\mathcal{T}(C_N^{\mathcal{H}})$ and that of the basic encoding $C_M^{\mathcal{B}}$. In addition to the width blowup, the basic encoding also requires a larger time overhead when $\delta > 1/2$.

$$\frac{\mathcal{T}(C_M^{\mathcal{B}})}{\mathcal{T}(C_N^{\mathcal{H}})} = \Omega\left[\frac{N^{\delta-1/2}}{R \cdot \log^{2\delta}(N)}\right] . \tag{69}$$

Using LDPC codes with single-shot decoding algorithms, the hierarchical memory is a more efficient way to achieve a low logical error rate in terms of both circuit depth and width.

Having said this, it is not clear if this advantage manifests for practical block lengths.

For small codes and high error rates, it may well be that it is still optimal to use the basic encoding. We expect to see a *crossover point*—a value of physical error rate where the hierarchical scheme $\{\mathcal{H}_N\}$ has a lower logical failure rate than the basic encoding $\{\mathcal{B}_M\}$ with the same overhead. Where exactly this crossover happens will depend on a number of parameters that are specific to the implementation, including the choice of the outer code, its threshold and our choice of decoders. In the rest of this section, we attempt to estimate where this happens.

6.2 Setup for numerical estimates

Outer code: We choose a quantum expander code as our outer code [TZ14; LTZ15]. We do not utilize any of the structure of the code, so any LDPC code with constant rate and polynomially scaling distance will suffice. For these reasons, we only briefly discuss the code construction. We

pick a classical code by sampling the check matrix from the ensemble of $m \times n$ matrices with 5 ones in each column and 8 ones in each row.

In particular, we pick a classical code with length 896 and 336 encoded bits. Work by Litsyn and Shevelev [LS02] computes the asymptotic weight distribution of codewords — with high probability, this code has distance 119. The resulting quantum code has parameters

$$N = 1\,116\,416, \quad K = 112\,896, \quad D = 119, \quad \Delta_q = 16, \quad \Delta_g = 13\ . \tag{70}$$

We choose this code as it has a high rate which is necessary to reduce the amount of overhead in the scheme. The large block length we consider here is a consequence of using code families with sub-linear distance scaling. However, the full trade-off for rate, distance, check weight, etc for linear-distance codes has not yet been explored. We note that even at a relative distance d/n of 10^{-3}, a linear-distance code of such large block length would achieve a distance of roughly 10^3.

While the hierarchical memory construction has good asymptotic performance guarantees, if the overhead is too high then the hierarchical memory wins only at an extremely low WER.[16]

Inner code: As in the earlier sections, we consider the square rotated surface codes $\mathcal{RS}_\ell$ for the inner code. In our estimates, we allow for $d_\ell = 3, 5, 9, 15, 21, 27$.

We make some assumptions about errors on both the logical and physical levels. We present these assumptions together below and discuss justifications for some assumptions in what follows.

Level-0 noise model: We assume circuit-level Pauli noise on each physical qubit. We treat SWAP gates and other Clifford operations separately.

1. Each t-qubit gate (except SWAP gates) at the physical level fails with a probability p and leaves behind one of the $4^t - 1$ non-trivial t-qubit Pauli operators picked uniformly at random. We assume that qubit reset completely removes all traces of the original state. However, it may reset to the wrong computational basis state with probability p.
2. The failure probability of the physical SWAP-gate is $r_{\mathsf{SWAP}} \cdot p$, where $r_{\mathsf{SWAP}} = 1, 10^{-1}, 10^{-2}$. In this setting, the surface code syndrome-extraction circuit is performed every $1/r_{\mathsf{SWAP}}$ SWAP-gates, so that at the physical level the circuit-level noise model remains relatively unchanged for different values of r_{SWAP}. This assumption will be discussed in detail in Section 6.2.2.

Level-1 noise model: We assume that the surface code fails at a probability $p^{(1)}_{\mathcal{RS}}(d_\ell)$, and that the effective noise witnessed by the outer code because of all the SWAP gates is $p^{(1)}$.

1. The logical error rate $p^{(1)}_{\text{phys}}(d_\ell)$ of each $\ell \times \ell$ rotated surface code tile is [WFH11; FMMC12]

$$p^{(1)}_{\text{phys}}(d_\ell) \approx 0.1 \left(\frac{p}{10^{-2}}\right)^{\lceil d_\ell/2 \rceil} \tag{71}$$

   per d_ℓ physical level timesteps where one physical level timestep is one round of syndrome extraction plus one (optional) transversal gate which totals roughly 6 gates. This assumption is discussed in Section 6.2.1.
2. The effective error rate per long-range CNOT gate is $p^{(1)} = 1 - (1 - p^{(1)}_{\text{phys}}(d_\ell))^{t_{\text{route}}+1}$. It is analogous to the two-qubit gate error rate in the model with long-range gates. t_{route} is the time required for permutation routing presented in Equation (16).

Level-2 noise model: Finally, we assume that the logical failure rate for the LDPC code, $p_{\mathcal{Q}}(n)$, is consistent with a minimum-weight decoder.

[16] 1 year is $\approx 10^{16}$ nanoseconds, 1 Hubble time is $\approx 10^{10}$ years or $\approx 10^{26}$ nanoseconds. For any practical purpose a WER of 10^{-25} per gate time should suffice.

1. For our LDPC code, we assume that the WER under circuit-level Pauli noise using long-range gates is

$$p_{\mathcal{Q}} = \left(\frac{p^{(1)}}{10^{-3}}\right)^{10} \tag{72}$$

per cycle of syndrome extraction. The threshold of the code is assumed to be about 10^{-3} under circuit noise. The exponent is 10 rather than half the distance which is ~ 55 because of hook errors. This assumption is discussed in Section 6.2.4.

If desired, readers can skip ahead to the numerical estimates in Section 6.3 and return to the justification of the noise model later.

6.2.1 Decoder performance for the inner code

Consider Equation (71) for the scaling of the logical failure rate for a surface code of distance d_ℓ. We assumed a surface code threshold of 10^{-2}.

This equation neglects:

1. finite-size effects present at very small code distances.
2. the slight reduction in threshold from inserting a layer of gates failing with rate p between syndrome extraction cycles[17]. Recall that this is necessary to implement a logical SWAP operation in the bilayer architecture as discussed in Section 4.3.
3. the distinction between rotated and standard surface codes. Owing to this, the expression for the logical error rate is an order of magnitude estimate. We expect our conclusions should be somewhat insensitive to the precise form of the logical error rate and also apply to more general locally decaying error models. For calculational convenience, we assume that the failure rate q after T syndrome extraction rounds is given by $1 - q = \left(1 - p^{(1)}_{\text{phys}}(d_\ell)\right)^{T/d}$.

6.2.2 Physical SWAP fidelity

Recall that simulating a long-range CNOT via SWAP gates results in a CNOT failure rate of $p^{(1)}$. We assume that the effective failure rate witnessed by the outer code is

$$p^{(1)} = 1 - (1 - p^{(1)}_{\text{phys}}(d_\ell))^{\frac{r_{\mathsf{SWAP}} t_{\text{route}}}{d_\ell}+1} . \tag{73}$$

The parameter t_{route} is the time required to perform a permutation routing in the bilayer architecture as specified in Equation (16). For convenience, we restate it here

$$t_{\text{route}} = (2d_\ell + 1)(3L - 3) + 8 . \tag{16}$$

The parameter r_{SWAP} bounds the (in)fidelity of the SWAP operation in terms of the CNOT gate (in)fidelity as we now explain.

In the previous sections, we assumed that all gates failed at the same rate. As noted in Section 5.3, the main source of noise in the hierarchical model stems from the SWAP gates. This worst-case model was convenient for a proof of the existence of a threshold. Furthermore, in many devices the SWAP gate is implemented using the same mechanism as the two-qubit entangling gates and so the noise rates are comparable. However, this is not the only way to implement SWAP gates.

In platforms where the qubits can be physically moved, we can effectively "rewire" the connectivity of the device at runtime. Physically swapping qubits does not require the qubit degree of freedom to be coupled to, and so one might expect that it is an easier task to perform with higher fidelity

[17] In general, the precise value of the circuit-level threshold already requires some assumptions about what gates are native in the device: The optimal syndrome extraction circuit with our physical layer layout requires 5 to 8 gates depending on these assumptions, so the insertion of an additional gate is relatively unimportant.

or speed. Such techniques have been demonstrated in some experimental platforms: Rearrangable tweezers in Rydberg platforms [End+16; Bar+16; Blu+22] and ion shuttling in trapped ion platforms [Hen+06; Kau+17]. In this setting, it is possible that the SWAP gate has much higher fidelity than CNOT gates.

Accordingly, in our model, we assign a constant r_{SWAP} which specifies the ratio of SWAP-gate and idle noise to CNOT-gate noise. With a less noisy r_{SWAP}, we perform $1/r_{\mathsf{SWAP}}$ level-0 SWAP operations per round of surface code syndrome extraction such that the physical error rate in the surface code syndrome extraction circuit remains constant with respect to r_{SWAP}. Utilizing this optimization, an entire permutation takes $r_{\mathsf{SWAP}} t_{\text{route}}$ rounds of syndrome extraction. Equation (73) is in terms of the surface code cycle (d_ℓ rounds of syndrome extraction), and the total number of surface code cycles is $\frac{r_{\mathsf{SWAP}} t_{\text{route}}}{d_\ell}$ for a permutation and 1 for an entangling gate. We have omitted floor and ceiling functions in this discussion for simplicity.

For example, in a neutral atom system [Blu+22], an array of qubits with a coherence time of seconds was rearranged with an average rearrangement speed of several microseconds per lattice site moved. If the dominant source of errors in rearrangement is due to idle errors, then we should assign an infidelity to the SWAP gate of roughly 10^{-5} whereas the two-qubit gate possessed an infidelity of about 10^{-2} i.e. $r_{\mathsf{SWAP}} \approx 10^{-3} - 10^{-2}$. Routing does not require generating entanglement, so the qubit can remain encoded in well-isolated degrees of freedom. Owing to this, we consider three scenarios: where r_{SWAP} is 10^0, 10^{-1}, or 10^{-2}.

We note that it is a simplification to consider the rearrangement primitive in each platform (tweezers, ion shuttling, etc.) as simply SWAP-gates: Frequently there are effects like accumulated motional heating, recooling, acceleration speed limits, etc, but we expect the basic routing ideas and qualitative conclusions remain the same even in the more complicated setting.

6.2.3 Hook errors

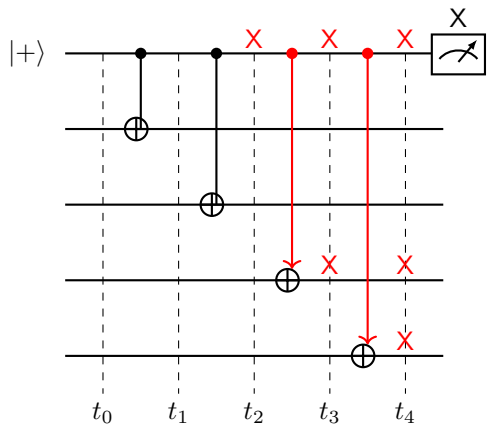

Figure 17: Hook errors: errors flowing onto data qubits. In this case, an X error appears in the midst of a syndrome-extraction circuit. This then propagates to the data qubits.

Current decoder technology for LDPC codes is relatively immature, so we assume a WER scaling consistent with a minimum-weight decoder. At physical failure rate p, we assume that the logical failure rate of an $[\![n, k, d, \Delta_q, \Delta_g]\!]$ LDPC code $\mathcal{Q}_n$ below threshold is dominated by a term proportional to p^t where t is the smallest number of fault locations that is uncorrectable. If our intuition is informed by an i.i.d. error model on qubits, we may expect $t \approx d/2$. However, this is not true in the context of syndrome-extraction circuits as corrupted syndrome qubits can spread errors to many data qubits.

These errors, called hook errors [DKLP02], are harmful errors that can dominate the lower error rate performance of the quantum code. By a rough estimate, they can reduce the distance of a $[\![n, k, d, \Delta_q, \Delta_g]\!]$ LDPC code by a factor of $\Delta_g/2$. To explain how they work, consider the example

measurement circuit for an X-type stabilizer generator as shown in Figure 4. An X error on the ancilla can propagate to a much larger data error.

In theory, the hook error is $O(1)$-sized and the syndrome extraction circuit is fault-tolerant. However, addressing these errors can significantly reduce the size of the LDPC code required to achieve a target logical failure rate.

Suppose ancilla qubits fail with probability p. A measurement circuit for a weight w operator can create hook errors with weight ranging from 1 up to w. If the circuit is measuring the checks of a code, the weight of the hook error can be reduced by using the measured stabilizer generator giving a maximum reduced weight of $\lfloor w/2 \rfloor$. An error is uncorrectable if it has weight at least $\lceil d/2 \rceil$. If we assume that each ancilla failure results in an error of weight $\lceil \Delta_g/2 \rceil$, then we only need t failures to cause an uncorrectable error, where t satisfies

$$t \cdot \lfloor \Delta_g/2 \rfloor \geq \lceil d/2 \rceil \ . \tag{74}$$

Then the probability of logical failure is p^t where $t \approx d/\Delta_g$.

This assumption is conservative — hook errors depend on the choice of syndrome-extraction circuit and may be minimized by particular choices of gate scheduling. For example, in the rotated surface code, there is a two-qubit gate schedule for the syndrome-extraction circuit such that the hook error has intersection-1 with a logical operator [TS14; CB18]. Using such a schedule, the below-threshold scaling is $\propto p^{\lceil d/2 \rceil}$ as one would expect from a depolarizing noise model. For general LDPC codes, the existence of measurement schedules that reduce the effects of hook errors is not yet clear.

While there exist many methods for suppressing hook errors such as Shor, Steane or Knill error correction [NC02], nearly all require more ancilla qubits. This presents a trade-off where, for a given number of qubits, either a larger block length code with correspondingly better parameters or a more resource-intensive syndrome-extraction circuit could be used. In the setting of a constant-rate LDPC code, larger distances come with more logical qubits, so the lowest overhead solution is to use the naive syndrome-extraction circuit with as large a code as possible[18]. Later, in Section 6.4.1, we will propose a method to mitigate the effects of hook errors outside of the asymptotic regime.

6.2.4 Decoder performance for the outer code

Hook errors discussed in the previous section need to be considered in the context of the concatenated scheme — the probability that an ancilla qubit fails is $p^{(1)}$.

Following the discussion in Section 6.2.3 on hook errors, $\lceil \lceil d/2 \rceil / \lfloor \Delta_g/2 \rfloor \rceil = 10$ for the code selected in Section 6.2. Assuming a threshold of about 10^{-3} under circuit noise, the WER under circuit-level Pauli noise using long-range gates goes as

$$\left(\frac{p^{(1)}}{10^{-3}} \right)^{10} \ . \tag{75}$$

per cycle of syndrome extraction.

For context, a slightly better threshold of about 3×10^{-3} has been observed for $(3, 4)$ hypergraph product codes using efficient decoders [TDB22] under circuit-level noise for syndrome-extraction circuits with long-range gates.

In practice, more information is available to the decoder owing to the concatenated structure: A decoder using this extra information about individual qubit reliability is likely to have a better threshold. We return to this subject in Section 6.4.2.

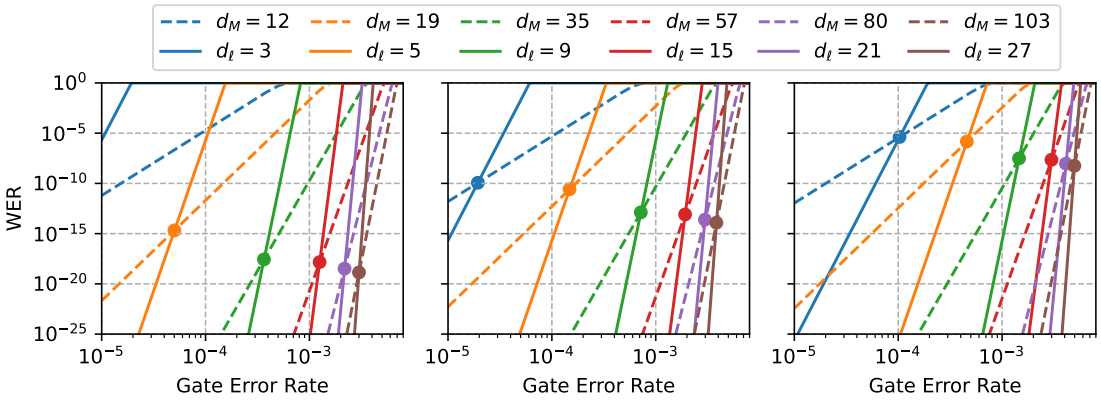

Figure 18: Comparison of a hierarchical memory (solid line) using a $(5, 8)$ quantum expander code with parameters $[\![1\,116\,416, 112\,896, 119]\!]$ and inner code distance d_ℓ, and a surface code (dashed line) with distance d_M. Lines of the same color use roughly the same number of physical qubits including all necessary ancilla qubits. All memories store $112\,896$ logical qubits. The 3 plots correspond to different values of r_{SWAP} equal to 10^0, 10^{-1}, and 10^{-2} (left to right) under the decoder performance assumptions made in Section 6.2. The surface code distance is rounded up, so it always uses slightly more qubits. The WER is with respect to the hierarchical memory syndrome extraction cycle.

6.3 Results

Using the duration of the hierarchical code syndrome extraction cycle as the unit of time that defines the WER, the results of the estimates are shown in Figure 18 for $r_{\mathsf{SWAP}} = 10^0$, 10^{-1}, or 10^{-2} and several sizes of inner rotated surface code. We can see the better scaling with gate error rate that the hierarchical memory achieves. While the LDPC code distance is fairly large, the "effective" distance has been reduced immensely by the weight-6 hook errors (potentially arising from the measurement of weight-13 check operators); because the outer code has distance $d = 119$, under our pessimistic assumptions just 10 fault locations are sufficient to cause an uncorrectable error. We expect future LDPC codes with better distance and better understanding of hook errors in syndrome extraction circuit gate scheduling will improve the WER scaling.

Under a standard circuit-level noise model, below a gate error rate of around 10^{-3}, and for a target WER of 10^{-20} to 10^{-10}, the hierarchical scheme may realize significant resource savings. Especially so if SWAP gates have much lower gate error rates than CNOT gates. We plot such a comparison in Figure 19 with a gate error rate of 3×10^{-3} (99.7% gate fidelity) and $r_{\mathsf{SWAP}} = 0.1$ With further engineering and more careful modeling, we believe the overhead of the hierarchical scheme can be reduced much more, so that the crossover point occurs at a practically relevant gate error rate and target WER. In the next section we will outline two ideas that will improve the performance of the hierarchical scheme: The decoder for the outer code is given far more information about the level-1 qubit reliabilities than in a circuit level noise model, and in the presence of noise bias, the syndrome extraction circuit can be tailored to reduce the effects of hook errors.

Another reason we expect this estimate is conservative is that we have assumed that the noise model is independent circuit noise which creates only 2-body correlated errors in the underling surface codes. In the setting of large, long-lived quantum memories, we expect it will become necessary to address noise sources that affect large patches of the system. Sources of such noise could include cosmic rays (superconducting qubits), large deviations in global control devices such as lasers (AMO systems), lightning strikes, power supply ripples, etc. For large memories, different parts of the memory may rely on systems operating independently (ex. lasers, fridges, power supplies, etc) which would make such "global" noise large on the scale of any reasonable surface code patch, but small on the scale of the full hierarchical memory. Concatenation of surface codes

[18] For very resource-constrained settings, it may still be worthwhile to use more sophisticated syndrome extraction circuits for a better effective relative distance.

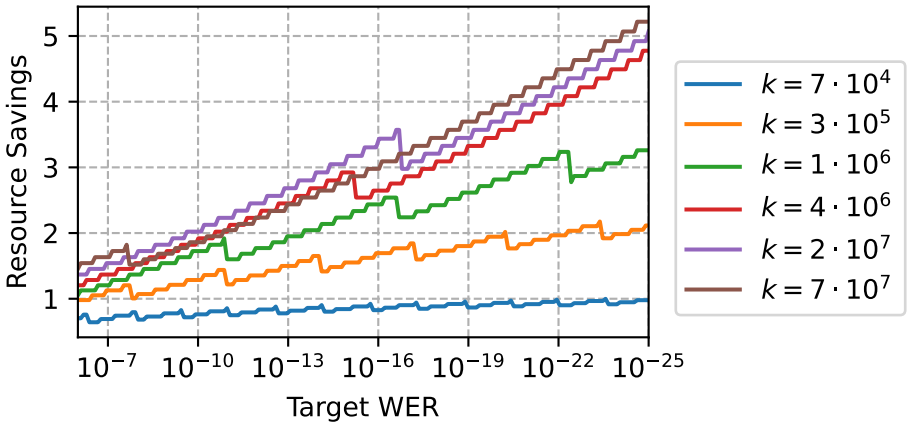

Figure 19: Estimated resource savings over surface codes for a hierarchical memory for $r_{\text{SWAP}} = 10^{-1}$ and a gate error rate of $3 \cdot 10^{-3}$ under the performance assumption of Section 6.2. The resource here refers to the total space footprint of the circuits; the y-axis represents the ratio $\mathcal{W}(C_M^{\mathcal{B}})/\mathcal{W}(C_N^{\mathcal{H}})$. We plot the resource savings for the $(4, 8)$ family of quantum expander codes with input code block lengths $512 \cdot 2^m$ for $m \in \{0, 1, 2, 3, 4, 5\}$. The number of logical qubits is indicated in the legend. Discontinuities in the plot are due to discretization of the surface code distance. Rare noise sources that create high weight errors may provide further resources savings over surface codes.

with constant length outer codes [Xu+22] has previously been considered in order to address such issues. It may be practical[19] to protect against such noise sources with a hierarchical scheme without additional overhead.

6.4 Future Performance Improvements

Having concluded a rough estimate of what the performance of the hierarchical memory might look like, we outline some ideas that could further improve the performance of the hierarchical memory relative to surface codes. In this section, we re-examine the WER for LDPC codes using biased-noise qubits and message-passing decoders.

6.4.1 Noise-Bias Tailored Syndrome Extraction

As discussed in Section 6.2.3, hook errors can be very damaging for general LDPC codes. In this section, we estimate the failure rate for hierarchical codes by making further assumptions on the dependence of logical failure given η-biased qubits. In particular, Equation (76) presented below is an ansatz for the logical failure probability $p_{\mathcal{Q}}$ of the outer code. However, we expect that this estimate can be considerably improved in the future by investigating in more detail how $p_{\mathcal{Q}}$ and depends on the bias η.

X errors on the ancilla qubit will propagate to an X or Z error on the data while Z errors on the ancilla qubit will simply flip the measurement outcome without propagating to a higher weight data error. If X errors can be suppressed on the ancilla qubits, then hook errors become much less likely. In many platforms, such noise is common or can be engineered into the experiment [Gri+20; Les+20; Con+22]. Noise bias has been exploited in the past by tailoring the quantum error correction scheme [AP08; WBP15; Pur+20; Bon+21; Rof+23] to the noise.

In Section 4.4, we introduced a technique to modify the bilayer architecture such that Level-1 qubits are noise biased. We can use this noise bias to suppress errors on the ancilla (X) that propagate

[19] Physics is local, so a very large surface code is likely sufficient, but it may be impractically large.

to higher weight data errors. We modify the assumptions of Subsection 6.2.4 and Equation (75) in a way that attempts to capture this behavior. Further study will be needed to make more precise estimates of logical error rates in this modified architecture.

The modified bilayer architecture uses elongated Level-1 qubits. If we choose the X distance to be larger than the Z distance according to $d_X = d_Z + \lceil 2\log(\eta)/\log(1/p)\rceil$, then the logical X error rate of the inner code is suppressed relative to the logical Z error rate by the bias factor $1/\eta$. If the accuracy threshold of the outer code is still 10^{-3} as we assumed for the case without noise bias, then for the modified architecture our estimate for the Level-2 WER becomes

$$p_{\mathcal{Q}} = \left(\frac{p^{(1)}}{10^{-3}}\right)^{\lceil d/2\rceil} + \left(\frac{p^{(1)}/\eta}{10^{-3}}\right)^{\lceil\lceil d/2\rceil/\lfloor \Delta_g/2\rfloor\rceil} . \tag{76}$$

The first term is the contribution from Level-1 logical Z errors; these do not propagate from ancilla to data, so that $\lceil d/2\rceil$ Level-1 errors are needed to cause a logical error at level 2. The second term arises from Level-1 logical X errors. These can propagate from ancilla to data, but they occur at a rate suppressed by the bias factor $1/\eta$.

Since surface codes are CSS codes, the X and Z noise can be corrected independently, so the X and Z logical failure rates can be examined independently up to small correlations introduced by Y-errors. Ignoring theses correlations and assuming that Equation (71) still holds with d replaced by d_X or d_Z,

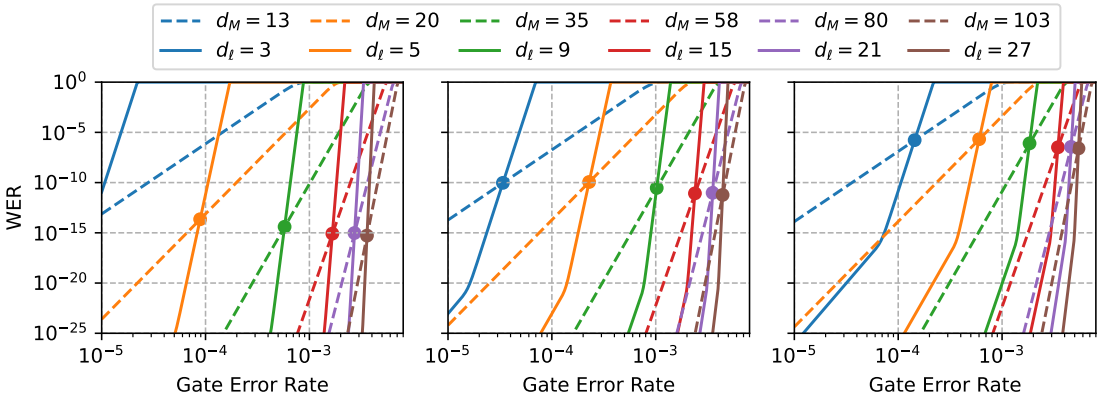

Figure 20: Comparison of a hierarchical memory (solid line) using a $(4, 8)$ quantum expander code with parameters $[\![327\,680, 65\,536, 32]\!]$ and inner code distance $d_Z = d_\ell$, and a surface code (dashed line) with distance ℓ_M. Lines of the same color use roughly the same number of physical qubits including all necessary ancilla qubits. The noise bias permits a smaller block length, so all memories store 65 536 logical qubits. The 3 plots correspond to different values of r_{SWAP} equal to 10^0, 10^{-1}, and 10^{-2} (left to right) under the modified decoder performance assumptions made in subsection 6.4.1. The surface codes underlying the hierarchical memory are rectangular with $d_X = 2d_\ell + 1$ The WER is for one round of the hierarchical memory's syndrome-extraction cycle.

We plot a similar comparison to Figure 18 with $d_X = 2d_Z + 1$, so that $\lceil d_X/2\rceil = 2\lceil d_Z/2\rceil$[20] (Figure 20). Using the greater resilience to hook errors, we also pick a smaller code with higher rate with parameters $[\![327\,680, 65\,536, 32]\!]$. Notice the effect of the bias is to increase the rate at which the WER falls with the level-0 gate error rate. The increased slope only persists until the two terms in Equation (76) become equal. One can see that using the bias, the hook errors are greatly suppressed leading to a better logical failure rate scaling in practically relevant regimes and a crossover point at a larger physical gate error rate.

[20]This choice is somewhat arbitrary. For a given WER target and gate error rate, the optimal aspect ratio is likely to be such that the target is at the "kink" of the WER in Figure 20.

6.4.2 Decoders that use the concatenated structure

Our asymptotic analysis used the underlying surface codes in a black-box manner—when decoding the outer LDPC code $\{\mathcal{Q}_n\}$, the tiles had either failed or succeeded. In contrast to this "hard information", much more information is available to the decoder for the outer code in the hierarchical setting. We may have access to "soft information", i.e. information about how reliable individual surface code patches are, which can then be passed to the outer code decoder. It is known that maximum-likelihood decoding on each level of a concatenated code, together with message passing between levels is an optimal decoding algorithm [Pou06].

Choice of outer code decoder: Soft information can be used in the quantum setting using Belief Propagation (BP), a class of iterative algorithms. Broadly, in each iteration, BP makes a series of graph-local decisions—qubits that are in the support of a stabilizer generator exchange information and update their beliefs about whether they have been corrupted. As there are only a constant number of qubits in the support of each stabilizer generator, the decision requires a constant-sized computation. Although it is very successful in the classical setting, BP faces difficulties when applied to quantum codes. In the classical setting, BP converges to an distribution over bits that corresponds to the most likely error. In the quantum setting, it was pointed out early on [PC08] that degeneracy is a major issue for BP—there are many errors that are equivalent as they differ only by a stabilizer generator. However, BP is unable to tell the difference and gets stuck in a local minimum. One simple way to get around this issue would be if more information were available about the qubits. If each qubit were known to fail with a different probability – even if that difference is small — it can help BP avoid local minima.

Since then, many ideas have been developed to use soft information in the quantum setting that overcome the shortcomings of BP [PK21; RWBC20; QVRC21; GGKL21; KL22; LP19; DMS22]. We now discuss ways to obtain soft information from the surface code.

Choice of inner code decoder: The tensor network decoders [BSV14; Chu21; TBF18; Bon+21; Tuc+19] are one class of surface code decoders that yield such soft information. The decoder outputs the probability of different (coset) logical failures for each tile. Unfortunately, it is unclear how to implement these algorithms in the fault-tolerant setting where syndrome information is unreliable. This setting requires growing bond dimension which makes implementing the decoder quite challenging.

More recently, BP decoders have been implemented for surface codes [RWBC20; OR23; Ach+23]. It is conceivable that such an algorithm could serve as a soft decoder for the surface codes as well.

A natural question is whether standard decoders such as Min-Weight Perfect Matching (MWPM) [DKLP02] or the Union-Find Decoder (UFD) [DN21] could be modified to yield soft information. For simplicity, consider bit flip noise at a rate of p. We define the *decoding graph* given by associating a vertex with each measured stabilizer generator. We add a special *boundary vertex* to which we associate the total parity of all measured stabilizer generators. Including the boundary vertex, each single qubit error is detected in exactly two places. For each error, an edge is added between the vertices where it is detected. To each edge, assign the weight $-\log\left(\frac{p}{1-p}\right)$ which is the log-likelihood of an error. The most likely error given the syndrome is then a subset of edges with minimal weight that produces the syndrome and can be computed efficiently by mapping onto the minimum-weight perfect matching problem.

On average, the expected weight of an error (and correction) will be linear in the block length. This is asymptotically larger than the distance of a surface code, so the most important feature of the correction is its *shape*. The Union-Find Decoder operates in two steps: First, it identifies clusters such that a valid correction is contained within the support of the clusters. Then, it treats the identified clusters as an erasure and runs an erasure correction decoder which produces a valid correction contained within the erasure.

One such way to obtain soft information from this process is to compute the log-likelihood of the minimum weight error that would lead to a logical fault when combined with the erasure. This can be computed efficiently by setting the edge weights within the erasure to 0 and computing the

minimal weight path between inequivalent boundaries. Call this quantity ϕ. We note that when no errors are detected, $\phi = -d \log\left(\frac{p}{1-p}\right)$, and when the cluster spans the system, $\phi = 0$. In the first case, it is extremely unlikely ($\propto p^d$) for a logical fault to have occurred while in the latter case, there is a 50% probability for a logical fault to have occurred. When passed to an outer-level decoder, ϕ or a monotonic function of ϕ may yield sufficient information to improve the logical failure rate dramatically.

7 Conclusions

We have constructed a quantum memory with a threshold using geometrically local gates to simulate long-range connectivity. We did so by constructing a code family $\{\mathcal{H}_N\}$ that we refer to as the hierarchical code. N indexes the size of the code; the N^{th} element $\mathcal{H}_N$ of this code is obtained by concatenating a constant-rate quantum LDPC code $\mathcal{Q}_n$ (the n-qubit outer code) and the surface code $\mathcal{RS}_\ell$ (the inner code). The outer code has a number of encoded qubits $k(n) = \rho \cdot n$ and distance $d(n) = \Theta(n^\delta)$ for positive constants ρ, δ. Our construction builds on Gottesman's proof of the existence of a threshold using quantum LDPC codes (Theorem 4 from [Got13]). The central idea in Gottesman's construction is that if the failure rate *per round* of syndrome extraction, denoted p_{round}, is a sufficiently small constant, then logical errors can be suppressed exponentially in the distance of the code. We showed that the requisite constant error rate per round can be achieved using geometrically-local gates if the inner code has suitable properties.

Although $\mathcal{H}_N$ is no longer an LDPC code, local operations suffice for extracting the error syndrome. In Section 4, we presented an explicit family of syndrome-extraction circuits $\{C_N^{\mathcal{H}}\}$ for $\mathcal{H}_N$. This circuit has width $\mathcal{W}(C_N^{\mathcal{H}}) = \Theta(N)$ and depth $\mathcal{T}(C_N^{\mathcal{H}}) = O(\sqrt{N}/R)$, where R denotes the range of physical SWAP gates. To describe this circuit for the hierarchical code, we first presented a construction of the syndrome-extraction circuit C_n for the outer LDPC code $\mathcal{Q}_n$ in Section 4.1. This circuit is based on a bilayer architecture — physical qubits are laid out in two layers in 2 dimensions. In our concatenated construction, the outer qubits of $\mathcal{Q}_n$ are replaced by rotated surface codes referred to as tiles. In Section 4.3, we demonstrated how to perform Level-1 logical Clifford operations on tiles using physical nearest-neighbor gates, including a novel technique for performing nearest-neighbor logical SWAP gates. We also discussed how to perform logical SWAP operations on tiles with range R_1 using physical SWAP operations with range R_0.

In Section 5, we showed that for fixed values of the physical failure rate p_{phys}, the error rate per round of syndrome-extraction, p_{round}, is a polynomial function of the depth $\mathcal{T}(C_N^{\mathcal{H}})$. Using an inner surface code with linear size ℓ, which can suppress errors exponentially in ℓ, we can guarantee that the Level-1 error rate per round is a constant by choosing $\ell = \Theta(\log(n))$. The resulting concatenated code $\mathcal{H}_N$ encodes a number of encoded qubits $K = \Omega(N/\log(N)^2)$. Furthermore, if the distance of the LDPC code $\mathcal{Q}_n$ is $d(n) = \Theta(n^\delta)$, then $\mathcal{H}_N$ can suppress errors superpolynomially; the Word Error Rate (WER) satisfies $p_{\mathcal{H}}(N) < \exp(-\Theta[N^\delta/\log^{2\delta}(N)])$. Given access to physical SWAP operations of range R, the syndrome-extraction circuit $C_N^{\mathcal{H}}$ has depth $O(\sqrt{N}/R)$.

Using this architecture we made numerical estimates of the WER $p_{\mathcal{H}}(N)$ in Section 6. We contrasted this with the WER $p_{\mathcal{B}}(M)$ of the *basic encoding* $\mathcal{B}_M$, where all logical qubits are encoded using only the surface code. We first made comparisons in the asymptotic regime, showing in Section 6.1 that if the outer constant-rate LDPC code has an efficient single-shot decoder, then a target logical error rate can be achieved more efficiently using the hierarchical encoding rather than the basic encoding.

We then proceeded with numerical estimates probing whether this advantage holds for practical code sizes and noise parameters. For this purpose, we compared the WERs of the basic encoding and hierarchical encoding when both schemes use the same total number of physical qubits. We found that the physical error rate has a *crossover point*; when the physical error rate is below this value, the hierarchical code outperforms the basic encoding. To perform these estimates, we made assumptions about the noise model and about the WER for surface codes and LDPC codes, and we assessed the impact of these assumptions on our conclusions. We also discussed some ways to

reduce the WER of hierarchical codes by modifying the syndrome-extraction circuit, improving the fidelity of SWAP operations, and using more sophisticated decoding algorithms.

1. We made the conservative assumption that propagation of error from Level-1 ancilla qubits to Level-1 data qubits reduces the effective distance of the outer code by a factor of Δ_g, the degree of the outer-code stabilizer generators. This error propagation can be mitigated if the noise in Level-1 qubits is highly biased, with X errors occurring much less frequently than Z errors. Even if the noise afflicting the physical qubits is unbiased, this Level-1 noise bias can be enforced by using an asymmetric surface code as the inner code of the hierarchical scheme.
2. The failure rate of the outer code grows in proportion to the depth of the permutation routing, and hence is sensitive to the error rate of Level-1 SWAP operations. By improving the error rate of physical SWAP gates we can improve the performance of the hierarchical code significantly.
3. We assumed that the decoding algorithm for the outer code makes no use of the syndrome information from the inner code blocks. We expect that a much better decoding scheme for the hierarchical code can be achieved by exploiting such information from the inner code when decoding the outer code.

Finally, we also highlighted that a hierarchical architecture might deal effectively with "burst" errors that damage a large cluster of physical qubits simultaneously. A severe burst error could corrupt several of the inner-code tiles, but the resulting Level-1 erasure errors can be adequately addressed by the decoder for the outer code.

8 Acknowledgements

AK is supported by the Bloch Postdoctoral Fellowship from Stanford University. AK acknowledges funding from NSF award CCF-1844628. CAP acknowledges funding from the Air Force Office of Scientific Research (AFOSR), FA9550-19-1-0360. JP acknowledges funding from the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research, (DE-NA0003525, DE-SC0020290), the U.S. Department of Energy QuantISED program (DE-SC0018407), the U.S. Department of Energy Quantum Systems Accelerator, the Air Force Office of Scientific Research (FA9550-19-1-0360), and the National Science Foundation (PHY-1733907). The Institute for Quantum Information and Matter is an NSF Physics Frontiers Center. We thank Nicolas Delfosse, Mary Wootters, Anthony Leverrier, Nouédyn Baspin, Bailey Gu, Alex Kubica, David Schuster, Manuel Endres, Michael Vasmer, and Pavel Panteleev for helpful conversations.

References

[AB90] F. Annexstein and M. Baumslag. "A unified approach to off-line permutation routing on parallel networks". In: *Proceedings of the second annual ACM symposium on Parallel algorithms and architectures - SPAA '90*. SPAA '90. ACM Press, 1990, pp. 398–406. DOI: 10.1145/97444.97707. URL: http://dx.doi.org/10.1145/97444.97707 (cit. on pp. 3, 20, 21).

[ACG94] Noga Alon, F. R. K. Chung, and R. L. Graham. "Routing Permutations on Graphs via Matchings". In: *SIAM Journal on Discrete Mathematics* 7.3 (May 1994), pp. 513–530. ISSN: 1095-7146. DOI: 10.1137/s0895480192236628. URL: http://dx.doi.org/10.1137/S0895480192236628 (cit. on p. 23).

[Ach+23] Rajeev Acharya et al. "Suppressing quantum errors by scaling a surface code logical qubit". In: *Nature* 614.7949 (Feb. 2023), pp. 676–681. ISSN: 1476-4687. DOI: 10.1038/s41586-022-05434-1. URL: http://dx.doi.org/10.1038/s41586-022-05434-1 (cit. on pp. 8, 57).

[AGP06] P. Aliferis, D. Gottesman, and J. Preskill. “Quantum accuracy threshold for concatenated distance-3 code”. In: *Quantum Information and Computation* 6.2 (Mar. 2006), pp. 97–165. ISSN: 1533-7146. DOI: 10.26421/qic6.2-1. arXiv: quant-ph/0504218. URL: http://arxiv.org/abs/quant-ph/0504218v3 (cit. on p. 13).

[AP08] Panos Aliferis and John Preskill. “Fault-tolerant quantum computation against biased noise”. In: *Physical Review A* 78.5 (Nov. 2008). ISSN: 1094-1622. DOI: 10.1103/physreva.78.052331. URL: http://dx.doi.org/10.1103/PhysRevA.78.052331 (cit. on p. 55).

[Aru+19] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. “Quantum supremacy using a programmable superconducting processor”. In: *Nature* 574.7779 (2019), pp. 505–510 (cit. on p. 8).

[Bar+16] Daniel Barredo, Sylvain de Léséleuc, Vincent Lienhard, Thierry Lahaye, and Antoine Browaeys. “An atom-by-atom assembler of defect-free arbitrary two-dimensional atomic arrays”. In: *Science* 354.6315 (Nov. 2016), pp. 1021–1023. ISSN: 1095-9203. DOI: 10.1126/science.aah3778. URL: http://dx.doi.org/10.1126/science.aah3778 (cit. on pp. 6, 52).

[BE21] Nikolas P. Breuckmann and Jens N. Eberhardt. “Balanced Product Quantum Codes”. In: *IEEE Transactions on Information Theory* 67.10 (Oct. 2021), pp. 6653–6674. ISSN: 1557-9654. DOI: 10.1109/tit.2021.3097347. URL: http://dx.doi.org/10.1109/TIT.2021.3097347 (cit. on pp. 2, 17).

[Bea+13] Robert Beals, Stephen Brierley, Oliver Gray, Aram W Harrow, Samuel Kutin, Noah Linden, Dan Shepherd, and Mark Stather. “Efficient distributed quantum computing”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 469.2153 (2013), p. 20120686 (cit. on p. 19).

[BFS23] Nouédyn Baspin, Omar Fawzi, and Ala Shayeghi. “A lower bound on the overhead of quantum error correction in low dimensions”. In: (2023). arXiv: 2302.04317. URL: http://arxiv.org/abs/2302.04317v1 (cit. on p. 8).

[BK22a] Nouédyn Baspin and Anirudh Krishna. “Connectivity constrains quantum codes”. In: *Quantum* 6 (May 2022), p. 711. ISSN: 2521-327X. DOI: 10.22331/q-2022-05-13-711. URL: http://dx.doi.org/10.22331/q-2022-05-13-711 (cit. on pp. 2, 16).

[BK22b] Nouédyn Baspin and Anirudh Krishna. “Quantifying Nonlocality: How Outperforming Local Quantum Codes Is Expensive”. In: *Physical Review Letters* 129.5 (July 2022). ISSN: 1079-7114. DOI: 10.1103/physrevlett.129.050505. URL: http://dx.doi.org/10.1103/PhysRevLett.129.050505 (cit. on pp. 2, 16).

[BK98] S. B. Bravyi and A. Yu. Kitaev. “Quantum codes on a lattice with boundary”. In: (1998). arXiv: quant-ph/9811052. URL: http://arxiv.org/abs/quant-ph/9811052v1 (cit. on pp. 2, 17).

[Blu+22] Dolev Bluvstein, Harry Levine, Giulia Semeghini, Tout T. Wang, Sepehr Ebadi, Marcin Kalinowski, Alexander Keesling, Nishad Maskara, Hannes Pichler, Markus Greiner, Vladan Vuletić, and Mikhail D. Lukin. “A quantum processor based on coherent transport of entangled atom arrays”. In: *Nature* 604.7906 (Apr. 2022), pp. 451–456. ISSN: 1476-4687. DOI: 10.1038/s41586-022-04592-6. URL: http://dx.doi.org/10.1038/s41586-022-04592-6 (cit. on pp. 6, 47, 52).

[BM06] H. Bombin and M. A. Martin-Delgado. “Topological Quantum Distillation”. In: *Physical Review Letters* 97.18 (Oct. 2006). ISSN: 1079-7114. DOI: 10.1103/physrevlett.97.180501. URL: http://dx.doi.org/10.1103/PhysRevLett.97.180501 (cit. on p. 2).

[Bom+21] Hector Bombin, Chris Dawson, Ryan V Mishmash, Naomi Nickerson, Fernando Pastawski, and Sam Roberts. “Logical blocks for fault-tolerant topological quantum computation”. In: *arXiv preprint arXiv:2112.12160* (2021) (cit. on p. 5).

[Bon+21] J. Pablo Bonilla Ataides, David K. Tuckett, Stephen D. Bartlett, Steven T. Flammia, and Benjamin J. Brown. "The XZZX surface code". In: *Nature Communications* 12.1 (Apr. 2021). ISSN: 2041-1723. DOI: 10.1038/s41467-021-22274-1. URL: http://dx.doi.org/10.1038/s41467-021-22274-1 (cit. on pp. 55, 57).

[BPT10] Sergey Bravyi, David Poulin, and Barbara Terhal. "Tradeoffs for Reliable Quantum Information Storage in 2D Systems". In: *Physical Review Letters* 104.5 (Feb. 2010). ISSN: 1079-7114. DOI: 10.1103/physrevlett.104.050503. URL: http://dx.doi.org/10.1103/PhysRevLett.104.050503 (cit. on pp. 2, 16).

[BSV14] Sergey Bravyi, Martin Suchara, and Alexander Vargo. "Efficient algorithms for maximum likelihood decoding in the surface code". In: *Physical Review A* 90.3 (Sept. 2014). ISSN: 1094-1622. DOI: 10.1103/physreva.90.032326. URL: http://dx.doi.org/10.1103/PhysRevA.90.032326 (cit. on p. 57).

[BT09] Sergey Bravyi and Barbara Terhal. "A no-go theorem for a two-dimensional self-correcting quantum memory based on stabilizer codes". In: *New Journal of Physics* 11.4 (Apr. 2009), p. 043029. ISSN: 1367-2630. DOI: 10.1088/1367-2630/11/4/043029. URL: http://dx.doi.org/10.1088/1367-2630/11/4/043029 (cit. on pp. 2, 16).

[Car+23] L. Cardani, I. Colantoni, A. Cruciani, F. De Dominicis, G. D'Imperio, M. Laubenstein, A. Mariani, L. Pagnanini, S. Pirro, C. Tomei, N. Casali, F. Ferroni, D. Frolov, L. Gironi, A. Grassellino, M. Junker, C. Kopas, E. Lachman, C. R. H. McRae, J. Mutus, M. Nastasi, D. P. Pappas, R. Pilipenko, M. Sisti, V. Pettinacci, A. Romanenko, D. Van Zanten, M. Vignati, J. D. Withrow, and N. Z. Zhelev. "Disentangling the sources of ionizing radiation in superconducting qubits". In: *The European Physical Journal C* 83.1 (Jan. 2023). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-023-11199-2. URL: http://dx.doi.org/10.1140/epjc/s10052-023-11199-2 (cit. on p. 8).

[CB18] Christopher Chamberland and Michael E. Beverland. "Flag fault-tolerant error correction with arbitrary distance codes". In: *Quantum* 2 (Feb. 2018), p. 53. ISSN: 2521-327X. DOI: 10.22331/q-2018-02-08-53. URL: http://dx.doi.org/10.22331/q-2018-02-08-53 (cit. on p. 53).

[Chu21] Christopher T. Chubb. "General tensor network decoding of 2D Pauli codes". In: (2021). arXiv: 2101.04125. URL: http://arxiv.org/abs/2101.04125v3 (cit. on p. 57).

[Con+22] Iris Cong, Harry Levine, Alexander Keesling, Dolev Bluvstein, Sheng-Tao Wang, and Mikhail D. Lukin. "Hardware-Efficient, Fault-Tolerant Quantum Computation with Rydberg Atoms". In: *Physical Review X* 12.2 (June 2022). ISSN: 2160-3308. DOI: 10.1103/physrevx.12.021049. URL: http://dx.doi.org/10.1103/PhysRevX.12.021049 (cit. on p. 55).

[CS96] A. R. Calderbank and Peter W. Shor. "Good quantum error-correcting codes exist". In: *Physical Review A* 54.2 (Aug. 1996), pp. 1098–1105. ISSN: 1094-1622. DOI: 10.1103/physreva.54.1098. URL: http://dx.doi.org/10.1103/physreva.54.1098 (cit. on p. 9).

[DBT21] Nicolas Delfosse, Michael E. Beverland, and Maxime A. Tremblay. "Bounds on stabilizer measurement circuits and obstructions to local implementations of quantum LDPC codes". In: (2021). arXiv: 2109.14599. URL: http://arxiv.org/abs/2109.14599v1 (cit. on pp. 2, 3, 17, 19, 26).

[DKLP02] Eric Dennis, Alexei Kitaev, Andrew Landahl, and John Preskill. "Topological quantum memory". In: *Journal of Mathematical Physics* 43.9 (Sept. 2002), pp. 4452–4505. ISSN: 1089-7658. DOI: 10.1063/1.1499754. URL: http://dx.doi.org/10.1063/1.1499754 (cit. on pp. 52, 57).

[DMS22] Julien Du Crest, Mehdi Mhalla, and Valentin Savin. "Stabilizer inactivation for message-passing decoding of quantum LDPC codes". In: *2022 IEEE Information Theory Workshop (ITW)*. IEEE. 2022, pp. 488–493 (cit. on p. 57).

[DN21] Nicolas Delfosse and Naomi H. Nickerson. “Almost-linear time decoding algorithm for topological codes”. In: *Quantum* 5 (Dec. 2021), p. 595. ISSN: 2521-327X. DOI: 10.22331/q-2021-12-02-595. URL: http://dx.doi.org/10.22331/q-2021-12-02-595 (cit. on p. 57).

[Eba+21] Sepehr Ebadi, Tout T. Wang, Harry Levine, Alexander Keesling, Giulia Semeghini, Ahmed Omran, Dolev Bluvstein, Rhine Samajdar, Hannes Pichler, Wen Wei Ho, Soonwon Choi, Subir Sachdev, Markus Greiner, Vladan Vuletić, and Mikhail D. Lukin. “Quantum phases of matter on a 256-atom programmable quantum simulator”. In: *Nature* 595.7866 (July 2021), pp. 227–232. ISSN: 1476-4687. DOI: 10.1038/s41586-021-03582-4. URL: http://dx.doi.org/10.1038/s41586-021-03582-4 (cit. on p. 8).

[EKZ20] Shai Evra, Tali Kaufman, and Gilles Zemor. “Decodable quantum LDPC codes beyond the square root distance barrier using high dimensional expanders”. In: *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, Nov. 2020, pp. 218–227. DOI: 10.1109/focs46700.2020.00029. URL: http://dx.doi.org/10.1109/FOCS46700.2020.00029 (cit. on p. 17).

[End+16] Manuel Endres, Hannes Bernien, Alexander Keesling, Harry Levine, Eric R Anschuetz, Alexandre Krajenbrink, Crystal Senko, Vladan Vuletic, Markus Greiner, and Mikhail D Lukin. “Atom-by-atom assembly of defect-free one-dimensional cold atom arrays”. In: *Science* 354.6315 (2016), pp. 1024–1027 (cit. on pp. 6, 52).

[Fes+22] Lorenzo Festa, Nikolaus Lorenz, Lea-Marina Steinert, Zaijun Chen, Philip Osterholz, Robin Eberhard, and Christian Gross. “Blackbody-radiation-induced facilitated excitation of Rydberg atoms in optical tweezers”. In: *Physical Review A* 105.1 (Jan. 2022). ISSN: 2469-9934. DOI: 10.1103/physreva.105.013109. URL: http://dx.doi.org/10.1103/PhysRevA.105.013109 (cit. on p. 8).

[FGL18a] Omar Fawzi, Antoine Grospellier, and Anthony Leverrier. “Constant Overhead Quantum Fault-Tolerance with Quantum Expander Codes”. In: *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, Oct. 2018, pp. 743–754. DOI: 10.1109/focs.2018.00076. URL: http://dx.doi.org/10.1109/FOCS.2018.00076 (cit. on pp. 6, 17, 49).

[FGL18b] Omar Fawzi, Antoine Grospellier, and Anthony Leverrier. “Efficient decoding of random errors for quantum expander codes”. In: *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*. STOC ’18. ACM, June 2018, pp. 521–534. DOI: 10.1145/3188745.3188886. URL: http://dx.doi.org/10.1145/3188745.3188886 (cit. on pp. 6, 17, 49).

[FMMC12] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. “Surface codes: Towards practical large-scale quantum computation”. In: *Physical Review A* 86.3 (Sept. 2012). ISSN: 1094-1622. DOI: 10.1103/physreva.86.032324. URL: http://dx.doi.org/10.1103/physreva.86.032324 (cit. on p. 50).

[Fri03] Joel Friedman. “A proof of alon’s second eigenvalue conjecture”. In: *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*. STOC03. ACM, June 2003, pp. 720–724. DOI: 10.1145/780542.780646. URL: http://dx.doi.org/10.1145/780542.780646 (cit. on p. 23).

[GGKL21] Antoine Grospellier, Lucien Grouès, Anirudh Krishna, and Anthony Leverrier. “Combining hard and soft decoders for hypergraph product codes”. In: *Quantum* 5 (Apr. 2021), p. 432. ISSN: 2521-327X. DOI: 10.22331/q-2021-04-15-432. URL: http://dx.doi.org/10.22331/q-2021-04-15-432 (cit. on p. 57).

[Got00] Daniel Gottesman. “Fault-tolerant quantum computation with local gates”. In: *Journal of Modern Optics* 47.2–3 (Feb. 2000), pp. 333–345. ISSN: 1362-3044. DOI: 10.1080/09500340008244046. URL: http://dx.doi.org/10.1080/09500340008244046 (cit. on p. 8).

[Got13] Daniel Gottesman. “Fault-Tolerant Quantum Computation with Constant Overhead”. In: (2013). arXiv: 1310.2984. URL: http://arxiv.org/abs/1310.2984v3 (cit. on pp. 2, 4, 9, 16, 18, 49, 58).

[Got97] Daniel Gottesman. “Stabilizer Codes and Quantum Error Correction”. In: (1997). arXiv: quant-ph/9705052. URL: http://arxiv.org/abs/quant-ph/9705052v1 (cit. on p. 38).

[Gri+20] A. Grimm, N. E. Frattini, S. Puri, S. O. Mundhada, S. Touzard, M. Mirrahimi, S. M. Girvin, S. Shankar, and M. H. Devoret. “Stabilization and operation of a Kerr-cat qubit”. In: *Nature* 584.7820 (Aug. 2020), pp. 205–209. ISSN: 1476-4687. DOI: 10.1038/s41586-020-2587-z. URL: http://dx.doi.org/10.1038/s41586-020-2587-z (cit. on p. 55).

[Hen+06] W. K. Hensinger, S. Olmschenk, D. Stick, D. Hucul, M. Yeo, M. Acton, L. Deslauriers, C. Monroe, and J. Rabchuk. “T-junction ion trap array for two-dimensional ion shuttling, storage, and manipulation”. In: *Applied Physics Letters* 88.3 (Jan. 2006). ISSN: 1077-3118. DOI: 10.1063/1.2164910. URL: http://dx.doi.org/10.1063/1.2164910 (cit. on pp. 6, 52).

[HFDM12] Dominic Horsman, Austin G Fowler, Simon Devitt, and Rodney Van Meter. “Surface code quantum computing by lattice surgery”. In: *New Journal of Physics* 14.12 (Dec. 2012), p. 123011. ISSN: 1367-2630. DOI: 10.1088/1367-2630/14/12/123011. URL: http://dx.doi.org/10.1088/1367-2630/14/12/123011 (cit. on pp. 5, 17).

[HHO21] Matthew B. Hastings, Jeongwan Haah, and Ryan O’Donnell. “Fiber bundle codes: breaking the n 1/2 polylog(n) barrier for Quantum LDPC codes”. In: *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. STOC ’21. ACM, June 2021, pp. 1276–1288. DOI: 10.1145/3406325.3451005. URL: http://dx.doi.org/10.1145/3406325.3451005 (cit. on p. 17).

[Kau+17] H. Kaufmann, T. Ruster, C. T. Schmiegelow, M. A. Luda, V. Kaushal, J. Schulz, D. von Lindenfels, F. Schmidt-Kaler, and U. G. Poschinger. “Fast ion swapping for quantum-information processing”. In: *Physical Review A* 95.5 (May 2017). ISSN: 2469-9934. DOI: 10.1103/physreva.95.052319. URL: http://dx.doi.org/10.1103/PhysRevA.95.052319 (cit. on pp. 6, 52).

[KB15] Aleksander Kubica and Michael E. Beverland. “Universal transversal gates with color codes: A simplified approach”. In: *Physical Review A* 91.3 (Mar. 2015). ISSN: 1094-1622. DOI: 10.1103/physreva.91.032330. URL: http://dx.doi.org/10.1103/PhysRevA.91.032330 (cit. on p. 2).

[Kit03] A.Yu. Kitaev. “Fault-tolerant quantum computation by anyons”. In: *Annals of Physics* 303.1 (Jan. 2003), pp. 2–30. ISSN: 0003-4916. DOI: 10.1016/s0003-4916(02)00018-0. URL: http://dx.doi.org/10.1016/S0003-4916(02)00018-0 (cit. on p. 2).

[KL22] Kao-Yueh Kuo and Ching-Yi Lai. “Exploiting degeneracy in belief propagation decoding of quantum codes”. In: *npj Quantum Information* 8.1 (2022), pp. 1–9 (cit. on p. 57).

[Knu97] Donald Ervin Knuth. *The art of computer programming*. Vol. 3. Pearson Education, 1997 (cit. on p. 20).

[KP13] Alexey A. Kovalev and Leonid P. Pryadko. “Fault tolerance of quantum low-density parity check codes with sublinear distance scaling”. In: *Physical Review A* 87.2 (Feb. 2013). ISSN: 1094-1622. DOI: 10.1103/physreva.87.020304. URL: http://dx.doi.org/10.1103/PhysRevA.87.020304 (cit. on pp. 2, 16).

[Kra+19] Christian Kraglund Andersen, Ants Remm, Stefania Balasiu, Sebastian Krinner, Johannes Heinsoo, Jean-Claude Besse, Mihai Gabureac, Andreas Wallraff, and Christopher Eichler. “Entanglement stabilization using parity detection and real-time feedback in superconducting circuits”. In: *arXiv e-prints* (2019), arXiv–1902 (cit. on p. 8).

[KT21] Tali Kaufman and Ran J. Tessler. “New cosystolic expanders from tensors imply explicit Quantum LDPC codes with $\Omega(\sqrt{n}\log^k n)$ distance”. In: *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. STOC ’21. ACM, June 2021, pp. 1317–1329. DOI: 10.1145/3406325.3451029. URL: http://dx.doi.org/10.1145/3406325.3451029 (cit. on p. 17).

[Les+20] Raphaël Lescanne, Marius Villiers, Théau Peronnin, Alain Sarlette, Matthieu Delbecq, Benjamin Huard, Takis Kontos, Mazyar Mirrahimi, and Zaki Leghtas. "Exponential suppression of bit-flips in a qubit encoded in an oscillator". In: *Nature Physics* 16.5 (Mar. 2020), pp. 509–513. ISSN: 1745-2481. DOI: 10.1038/s41567-020-0824-x. URL: http://dx.doi.org/10.1038/s41567-020-0824-x (cit. on p. 55).

[Leu+19] N. Leung, Y. Lu, S. Chakram, R. K. Naik, N. Earnest, R. Ma, K. Jacobs, A. N. Cleland, and D. I. Schuster. "Deterministic bidirectional communication and remote entanglement generation between superconducting qubits". In: *npj Quantum Information* 5.1 (Feb. 2019). ISSN: 2056-6387. DOI: 10.1038/s41534-019-0128-0. URL: http://dx.doi.org/10.1038/s41534-019-0128-0 (cit. on p. 3).

[LH22] Ting-Chun Lin and Min-Hsiu Hsieh. "Good quantum LDPC codes with linear time decoder from lossless expanders". In: (2022). arXiv: 2203.03581. URL: http://arxiv.org/abs/2203.03581v1 (cit. on p. 2).

[Lit19] Daniel Litinski. "A Game of Surface Codes: Large-Scale Quantum Computing with Lattice Surgery". In: *Quantum* 3 (Mar. 2019), p. 128. ISSN: 2521-327X. DOI: 10.22331/q-2019-03-05-128. URL: http://dx.doi.org/10.22331/q-2019-03-05-128 (cit. on p. 5).

[LP19] Ye-Hua Liu and David Poulin. "Neural Belief-Propagation Decoders for Quantum Error-Correcting Codes". In: *Physical Review Letters* 122.20 (May 2019). ISSN: 1079-7114. DOI: 10.1103/physrevlett.122.200501. URL: http://dx.doi.org/10.1103/PhysRevLett.122.200501 (cit. on p. 57).

[LS02] S. Litsyn and V. Shevelev. "On ensembles of low-density parity-check codes: asymptotic distance distributions". In: *IEEE Transactions on Information Theory* 48.4 (Apr. 2002), pp. 887–908. ISSN: 0018-9448. DOI: 10.1109/18.992777. URL: http://dx.doi.org/10.1109/18.992777 (cit. on p. 50).

[LTZ15] Anthony Leverrier, Jean-Pierre Tillich, and Gilles Zémor. "Quantum expander codes". In: *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*. IEEE. 2015, pp. 810–824 (cit. on pp. 6, 17, 49).

[LZ22] Anthony Leverrier and Gilles Zemor. "Quantum Tanner codes". In: *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, Oct. 2022, pp. 872–883. DOI: 10.1109/focs54457.2022.00117. URL: http://dx.doi.org/10.1109/FOCS54457.2022.00117 (cit. on pp. 2, 17).

[Ma+22] Shuo Ma, Alex P. Burgers, Genyue Liu, Jack Wilson, Bichen Zhang, and Jeff D. Thompson. "Universal Gate Operations on Nuclear Spin Qubits in an Optical Tweezer Array of Yb 171 Atoms". In: *Physical Review X* 12.2 (May 2022). ISSN: 2160-3308. DOI: 10.1103/physrevx.12.021028. URL: http://dx.doi.org/10.1103/physrevx.12.021028 (cit. on p. 8).

[Mad+20] Ivaylo S. Madjarov, Jacob P. Covey, Adam L. Shaw, Joonhee Choi, Anant Kale, Alexandre Cooper, Hannes Pichler, Vladimir Schkolnik, Jason R. Williams, and Manuel Endres. "High-fidelity entanglement and detection of alkaline-earth Rydberg atoms". In: *Nature Physics* 16.8 (May 2020), pp. 857–861. ISSN: 1745-2481. DOI: 10.1038/s41567-020-0903-z. URL: http://dx.doi.org/10.1038/s41567-020-0903-z (cit. on p. 8).

[McE+21] Matt McEwen, Lara Faoro, Kunal Arya, Andrew Dunsworth, Trent Huang, Seon Kim, Brian Burkett, Austin Fowler, Frank Arute, Joseph C. Bardin, Andreas Bengtsson, Alexander Bilmes, Bob B. Buckley, Nicholas Bushnell, Zijun Chen, Roberto Collins, Sean Demura, Alan R. Derk, Catherine Erickson, Marissa Giustina, Sean D. Harrington, Sabrina Hong, Evan Jeffrey, Julian Kelly, Paul V. Klimov, Fedor Kostritsa, Pavel Laptev, Aditya Locharla, Xiao Mi, Kevin C. Miao, Shirin Montazeri, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Alex Opremcak, Chris Quintana, Nicholas Redd, Pedram Roushan, Daniel Sank, Kevin J. Satzinger, Vladimir Shvarts, Theodore White, Z. Jamie Yao, Ping Yeh, Juhwan Yoo, Yu Chen, Vadim Smelyanskiy, John M. Martinis, Hartmut Neven, Anthony Megrant, Lev Ioffe, and Rami Barends.

"Resolving catastrophic error bursts from cosmic rays in large arrays of superconducting qubits". In: *Nature Physics* 18.1 (Dec. 2021), pp. 107–111. ISSN: 1745-2481. DOI: 10.1038/s41567-021-01432-8. URL: http://dx.doi.org/10.1038/s41567-021-01432-8 (cit. on p. 8).

[NC02] Michael A Nielsen and Isaac Chuang. *Quantum computation and quantum information*. 2002. DOI: 10.1038/46503 (cit. on p. 53).

[OR23] Josias Old and Manuel Rispler. "Generalized Belief Propagation Algorithms for Decoding of Surface Codes". In: *Quantum* 7 (June 2023), p. 1037. ISSN: 2521-327X. DOI: 10.22331/q-2023-06-07-1037. URL: http://dx.doi.org/10.22331/q-2023-06-07-1037 (cit. on p. 57).

[PC08] David Poulin and Yeojin Chung. "On the iterative decoding of sparse quantum codes". In: *Quantum Information & Computation* 8.10 (2008), pp. 987–1000 (cit. on pp. 8, 57).

[Per+21] Avikar Periwal, Eric S. Cooper, Philipp Kunkel, Julian F. Wienand, Emily J. Davis, and Monika Schleier-Smith. "Programmable interactions and emergent geometry in an array of atom clouds". In: *Nature* 600.7890 (Dec. 2021), pp. 630–635. ISSN: 1476-4687. DOI: 10.1038/s41586-021-04156-0. URL: http://dx.doi.org/10.1038/s41586-021-04156-0 (cit. on p. 3).

[PK20] Pavel Panteleev and Gleb Kalachev. "Quantum LDPC Codes with Almost Linear Minimum Distance". In: *arXiv preprint arXiv:2012.04068* (2020) (cit. on p. 17).

[PK21] Pavel Panteleev and Gleb Kalachev. "Degenerate Quantum LDPC Codes With Good Finite Length Performance". In: *Quantum* 5 (Nov. 2021), p. 585. ISSN: 2521-327X. DOI: 10.22331/q-2021-11-22-585. URL: http://dx.doi.org/10.22331/q-2021-11-22-585 (cit. on p. 57).

[PK22] Pavel Panteleev and Gleb Kalachev. "Asymptotically good Quantum and locally testable classical LDPC codes". In: *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*. STOC '22. ACM, June 2022. DOI: 10.1145/3519935.3520017. URL: http://dx.doi.org/10.1145/3519935.3520017 (cit. on pp. 2, 17).

[Pou06] David Poulin. "Optimal and efficient decoding of concatenated quantum block codes". In: *Physical Review A* 74.5 (Nov. 2006). ISSN: 1094-1622. DOI: 10.1103/physreva.74.052333. URL: http://dx.doi.org/10.1103/PhysRevA.74.052333 (cit. on p. 57).

[Pur+20] Shruti Puri, Lucas St-Jean, Jonathan A. Gross, Alexander Grimm, Nicholas E. Frattini, Pavithran S. Iyer, Anirudh Krishna, Steven Touzard, Liang Jiang, Alexandre Blais, Steven T. Flammia, and S. M. Girvin. "Bias-preserving gates with stabilized cat qubits". In: *Science Advances* 6.34 (Aug. 2020). ISSN: 2375-2548. DOI: 10.1126/sciadv.aay5901. URL: http://dx.doi.org/10.1126/sciadv.aay5901 (cit. on p. 55).

[QVRC21] Armanda O. Quintavalle, Michael Vasmer, Joschka Roffe, and Earl T. Campbell. "Single-Shot Error Correction of Three-Dimensional Homological Product Codes". In: *PRX Quantum* 2.2 (June 2021). ISSN: 2691-3399. DOI: 10.1103/prxquantum.2.020340. URL: http://dx.doi.org/10.1103/PRXQuantum.2.020340 (cit. on pp. 17, 57).

[Rav+22] Nithin Raveendran, Narayanan Rengaswamy, Filip Rozpędek, Ankur Raina, Liang Jiang, and Bane Vasić. "Finite Rate QLDPC-GKP Coding Scheme that Surpasses the CSS Hamming Bound". In: *Quantum* 6 (July 2022), p. 767. ISSN: 2521-327X. DOI: 10.22331/q-2022-07-20-767. URL: http://dx.doi.org/10.22331/q-2022-07-20-767 (cit. on p. 8).

[Rof+23] Joschka Roffe, Lawrence Z. Cohen, Armanda O. Quintavalle, Daryus Chandra, and Earl T. Campbell. "Bias-tailored quantum LDPC codes". In: *Quantum* 7 (May 2023), p. 1005. ISSN: 2521-327X. DOI: 10.22331/q-2023-05-15-1005. URL: http://dx.doi.org/10.22331/q-2023-05-15-1005 (cit. on p. 55).

[RWBC20] Joschka Roffe, David R. White, Simon Burton, and Earl Campbell. "Decoding across the quantum low-density parity-check code landscape". In: *Physical Review Research* 2.4 (Dec. 2020). ISSN: 2643-1564. DOI: 10.1103/physrevresearch.2.043423. URL: http://dx.doi.org/10.1103/PhysRevResearch.2.043423 (cit. on p. 57).

[Sch+03] Alexander Schrijver et al. *Combinatorial optimization: polyhedra and efficiency*. Vol. 24. Springer, 2003 (cit. on pp. 21, 22, 68).

[SDT06] Krysta M. Svore, David P. DiVincenzo, and Barbara M. Terhal. "Noise Threshold for a Fault-Tolerant Two-Dimensional Lattice Architecture". In: (2006). arXiv: quant-ph/0604090. URL: http://arxiv.org/abs/quant-ph/0604090v3 (cit. on p. 8).

[STD05] Krysta M. Svore, Barbara M. Terhal, and David P. DiVincenzo. "Local fault-tolerant quantum computation". In: *Physical Review A* 72.2 (Aug. 2005). ISSN: 1094-1622. DOI: 10.1103/physreva.72.022317. URL: http://dx.doi.org/10.1103/PhysRevA.72.022317 (cit. on p. 8).

[Ste96] Andrew Steane. "Multiple-particle interference and quantum error correction". In: *Proceedings of the Royal Society A* 452.1954 (1996), pp. 2551–2577 (cit. on p. 9).

[TBF18] David K. Tuckett, Stephen D. Bartlett, and Steven T. Flammia. "Ultrahigh Error Threshold for Surface Codes with Biased Noise". In: *Physical Review Letters* 120.5 (Jan. 2018). ISSN: 1079-7114. DOI: 10.1103/physrevlett.120.050505. URL: http://dx.doi.org/10.1103/PhysRevLett.120.050505 (cit. on p. 57).

[TDB22] Maxime A. Tremblay, Nicolas Delfosse, and Michael E. Beverland. "Constant-Overhead Quantum Error Correction with Thin Planar Connectivity". In: *Physical Review Letters* 129.5 (July 2022). ISSN: 1079-7114. DOI: 10.1103/physrevlett.129.050504. URL: http://dx.doi.org/10.1103/PhysRevLett.129.050504 (cit. on p. 53).

[Tho+22] Ted Thorbeck, Andrew Eddins, Isaac Lauer, Douglas T McClure, and Malcolm Carroll. "TLS Dynamics in a Superconducting Qubit Due to Background Ionizing Radiation". In: *arXiv preprint arXiv:2210.04780* (2022) (cit. on p. 8).

[TK77] Clark D Thompson and Hsiang Tsung Kung. "Sorting on a mesh-connected parallel computer". In: *Communications of the ACM* 20.4 (1977), pp. 263–271 (cit. on p. 22).

[TS14] Yu Tomita and Krysta M. Svore. "Low-distance surface codes under realistic quantum noise". In: *Physical Review A* 90.6 (Dec. 2014). ISSN: 1094-1622. DOI: 10.1103/physreva.90.062320. URL: http://dx.doi.org/10.1103/PhysRevA.90.062320 (cit. on p. 53).

[Tuc+19] David K. Tuckett, Andrew S. Darmawan, Christopher T. Chubb, Sergey Bravyi, Stephen D. Bartlett, and Steven T. Flammia. "Tailoring Surface Codes for Highly Biased Noise". In: *Physical Review X* 9.4 (Nov. 2019). ISSN: 2160-3308. DOI: 10.1103/physrevx.9.041031. URL: http://dx.doi.org/10.1103/PhysRevX.9.041031 (cit. on p. 57).

[TZ14] Jean-Pierre Tillich and Gilles Zemor. "Quantum LDPC Codes With Positive Rate and Minimum Distance Proportional to the Square Root of the Blocklength". In: *IEEE Transactions on Information Theory* 60.2 (Feb. 2014), pp. 1193–1202. ISSN: 1557-9654. DOI: 10.1109/tit.2013.2292061. URL: http://dx.doi.org/10.1109/TIT.2013.2292061 (cit. on pp. 2, 17, 49).

[Vep+20] Antti P Vepsäläinen, Amir H Karamlou, John L Orrell, Akshunna S Dogra, Ben Loer, Francisca Vasconcelos, David K Kim, Alexander J Melville, Bethany M Niedzielski, Jonilyn L Yoder, et al. "Impact of ionizing radiation on superconducting qubit coherence". In: *Nature* 584.7822 (2020), pp. 551–556 (cit. on p. 8).

[WBP15] Paul Webster, Stephen D Bartlett, and David Poulin. "Reducing the overhead for quantum computation when noise is biased". In: *Physical Review A* 92.6 (2015), p. 062309 (cit. on p. 55).

[WFH11] David S. Wang, Austin G. Fowler, and Lloyd C. L. Hollenberg. "Surface code quantum computing with error rates over 1%". In: *Physical Review A* 83.2 (Feb. 2011). ISSN: 1094-1622. DOI: 10.1103/physreva.83.020302. URL: http://dx.doi.org/10.1103/PhysRevA.83.020302 (cit. on p. 50).

[Wik22] Wikipedia contributors. *Permutation — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Permutation&oldid=1118545340. [Online; accessed 23-November-2022]. 2022 (cit. on p. 20).

[Xu+22] Qian Xu, Alireza Seif, Haoxiong Yan, Nam Mannucci, Bernard Ousmane Sane, Rodney Van Meter, Andrew N. Cleland, and Liang Jiang. "Distributed Quantum Error Correction for Chip-Level Catastrophic Errors". In: *Physical Review Letters* 129.24 (Dec. 2022). ISSN: 1079-7114. DOI: 10.1103/physrevlett.129.240502. URL: http://dx.doi.org/10.1103/PhysRevLett.129.240502 (cit. on pp. 8, 55).

[YK24] Hayata Yamasaki and Masato Koashi. "Time-Efficient Constant-Space-Overhead Fault-Tolerant Quantum Computation". In: *Nature Physics* 20.2 (Jan. 2024), pp. 247–253. ISSN: 1745-2481. DOI: 10.1038/s41567-023-02325-8. URL: http://dx.doi.org/10.1038/s41567-023-02325-8 (cit. on p. 8).

[Zei+16] Johannes Zeiher, Rick van Bijnen, Peter Schauß, Sebastian Hild, Jae-yoon Choi, Thomas Pohl, Immanuel Bloch, and Christian Gross. "Many-body interferometry of a Rydberg-dressed spin lattice". In: *Nature Physics* 12.12 (Aug. 2016), pp. 1095–1099. ISSN: 1745-2481. DOI: 10.1038/nphys3835. URL: http://dx.doi.org/10.1038/nphys3835 (cit. on p. 8).

A Glossary

1. Set notation: for natural numbers $n \in \mathbb{N}$, $[n] = \{1, ..., n\}$.
2. Sums over sets: For a set S and a subset $A \subseteq S$, the sum $\sum_{B \supseteq A} f(B)$ is taken over all subsets $B \subseteq S$ such that $A \subseteq B$.
3. Asymptotics: for functions $f, g : \mathbb{N} \to \mathbb{R}$, we say
   (a) $f(n) = O(g(n))$ if there exists an $n_0 \in \mathbb{N}$ and a positive number c independent of n such that for all $n > n_0$, $f(n) \leq g(n)$.
   (b) $f(n) = \Omega(g(n))$ if $g(n) = O(f(g))$.
   (c) $f(n) = \Theta(g(n))$ if there exists an $n_0 \in \mathbb{N}$ and positive numbers a, b independent of n such that $a \cdot g(n) \leq f(n) \leq b \cdot g(n)$.

   We may use $O_p(\cdot)$, $\Omega_p(\cdot)$ and $\Theta_p(\cdot)$ to indicate that the numbers a, b and c may depend on some parameter p pertinent to the problem at hand.
4. (Circuit) Step: A single timestep in which each qubit may participate in only one gate.
5. (Circuit) Stage: This refers to the time interval in the circuit $C_n^{\mathcal{Q}}$ required to simulate one entangling gate. One stage has at most $\mathcal{T}_{\text{perm}}$ steps.
6. (Measurement) Round: A complete measurement of all the stabilizer generators of the code producing one outcome for each stabilizer generator.
7. $\mathcal{K}$ is the set of Clifford operations we use to construct syndrome-extraction circuits in 2 dimensions. It includes the following elements.
   (a) Initialization of new qubits in state $|0\rangle$ or $|+\rangle$,
   (b) Single-qubit Pauli gates,
   (c) Two-qubit Clifford gates CNOT and CZ between nearest-neighbor qubits,
   (d) Single-qubit Pauli X and Z measurements,
   (e) Physical SWAP operation with range R.
8. In the context of concatenated codes, $\mathcal{K}$ carries subscripts $\mathcal{K}_0$, $\mathcal{K}_1$ to refer to Level-0 (physical) and Level-1 (logical) Clifford operations.

B Constructing the ideal syndrome-extraction circuit $(C_n^{\mathcal{Q}})^{\text{ideal}}$

In this section, we return to the claim in Section 4.1. We prove that the syndrome-extraction circuit $(C_n^{\mathcal{Q}})^{\text{ideal}}$ for a $[\![n, k, d, \Delta_q, \Delta_g]\!]$ code can be constructed such that its depth is at most $s := 2\Delta + 4$, where $\Delta = 2\max(\Delta_q, \Delta_g)$.

Proof. By definition, each qubit participates in at most Δ_q stabilizer generators and each stabilizer generator contains at most Δ_g qubits in its support. We use the *Tanner graph* $\mathcal{T}(\mathcal{Q}_n) = (V \cup C^{\mathsf{X}} \cup C^{\mathsf{Z}}, E)$, a tripartite graph corresponding to the code $\mathcal{Q}_n$ where:

1. There is a vertex $v \in V$ for each qubit in the code. $|V| = n$.
2. There is a vertex $u_i^{\mathsf{X}} \in$ for each X-type generator $\mathsf{S}_i^{\mathsf{X}}$. $|C^{\mathsf{X}}| = m_{\mathsf{X}}$.
3. There is a vertex w_j^{Z} for each Z-type generator $\mathsf{S}_j^{\mathsf{Z}}$. $|C^{\mathsf{Z}}| = m_{\mathsf{Z}}$.

Consider the bipartite Tanner graph $\mathcal{T}^{\mathsf{X}} = (V \cup C^{\mathsf{X}}, E)$ that corresponds to the X-type generators of the code $\mathcal{Q}$.

In each step, each qubit can be involved in at most one gate. This can be phrased as a graph coloring problem: we color the edges of $\mathcal{T}^{\mathsf{X}}$ such that no two edges incident to a vertex have the same color. Since $\mathcal{T}^{\mathsf{X}}$ is bipartite, such an edge coloring can be computed efficiently using $\max(\Delta_q, \Delta_g)$ colors [Sch+03].

To measure the X-type syndromes, the first phase of the circuit $(C_n^{\mathcal{Q}})^{\text{ideal}}$ is partitioned into $\max(\Delta_q, \Delta_g)$ steps. In the t^{th} step, we perform the two-qubit gates corresponding to the edge color t.

Once completed, the same process is repeated for the Z-type syndromes. Following a similar line of reasoning, this requires $\Delta = \max(\Delta_q, \Delta_g)$ applications of two-qubit gates.

The circuit thus has two phases: first the X-type syndromes are measured followed by the Z-type syndromes [21] which completes a measurement of all stabilizer generators.

The total number of entangling stages is therefore 2Δ, where $\Delta = \max(\Delta_q, \Delta_g)$. Accounting for one stage for preparing and measuring ancilla qubits in each phase, we have a total of $s = 2\Delta + 4$ stages to measure syndromes. ■

[21] This is unlike the surface code where both types of syndromes are measured at once.