# Toward a 2D Local Implementation of Quantum Low-Density Parity-Check Codes

Noah Berthusen[,1,*] Dhruv Devulapalli[,1] Eddie Schoute[,2] Andrew M. Childs[,1,3]
Michael J. Gullans[,1] Alexey V. Gorshkov[,1,4] and Daniel Gottesman[1,3]

[1]*Joint Center for Quantum Information and Computer Science, National Institute of Standards and Technology
(NIST)–University of Maryland, College Park, Maryland 20742, USA*

[2]*Computer, Computational, and Statistical Sciences Division, Los Alamos National Laboratory,
Los Alamos, New Mexico 87545, USA*

[3]*Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland,
College Park, Maryland 20742, USA*

[4]*Joint Quantum Institute, National Institute of Standards and Technology (NIST)–University of Maryland,
College Park, Maryland 20742, USA*

Geometric locality is an important theoretical and practical factor for quantum low-density parity-check (qLDPC) codes that affects code performance and ease of physical realization. For device architectures restricted to two-dimensional (2D) local gates, naively implementing the high-rate codes suitable for low-overhead fault-tolerant quantum computing incurs prohibitive overhead. In this work, we present an error-correction protocol built on a bilayer architecture that aims to reduce operational overheads when restricted to 2D local gates by measuring some generators less frequently than others. We investigate the family of bivariate-bicycle qLDPC codes and show that they are well suited for a parallel syndrome-measurement scheme using fast routing with local operations and classical communication (LOCC). Through circuit-level simulations, we find that in some parameter regimes, bivariate-bicycle codes implemented with this protocol have logical error rates comparable to the surface code while using fewer physical qubits.

## I. INTRODUCTION

The surface code, despite showing promising theoretical and experimental performance [1–6], is poorly suited to large-scale fault-tolerant quantum computation due to its large qubit overhead [4,7,8]. As a result, there has been much effort on the development of high-rate quantum low-density parity-check (qLDPC) codes [9]. As these codes can encode multiple logical qubits, the required space resources are reduced, in some instances, to a constant [10].

One of the main drawbacks of these high-rate qLDPC codes is that many long-range connections are needed to implement their syndrome-extraction circuits [11–14]. This is a pressing issue for architectures such as superconducting qubits. There, many of the current designs only

allow two-qubit gates to be performed between qubits that are two-dimensional (2D) nearest neighbors, in which case implementing these long-range entangling gates incurs significant overhead [15,16]. Several recent proposals have attempted to alleviate this overhead by taking advantage of more complex electrical wiring of the superconducting circuits [17,18], employing code concatenation [19,20] or using bosonic cat qubits [21]. Implementing these long-range connections is less problematic in architectures such as neutral atoms, ion traps, or semiconductor spin qubits that can implement long-range gates through qubit movement [22–28]. However, since movement adds additional complications associated with qubit decoherence, heating, and loss, it is worthwhile to consider schemes that limit the amount of movement. In the extreme case, one can consider qubits that are fixed in space and solely use local interactions to perform entangling gates. Such studies provide additional insight into the trade-offs associated with engineering long-range connectivity through qubit motion or more complex electrical wiring.

In this paper, we present an approach to qLDPC codes that works without qubit motion or long-range couplers, inspired by the so-called stacked model [13,29]. In this

model, we assume that the high-rate qLDPC codes of interest have the property that after embedding the code into $\mathbb{Z}^2$, the majority of the stabilizer generators are local; i.e., their qubits are contained within a ball of constant radius. We claim that most of the work required to perform the syndrome-extraction circuit with 2D local gates comes from measuring the few nonlocal generators, so measuring these generators less frequently has the potential to significantly reduce the time overhead, ideally at only a minor cost to the error-correction performance of the code. It has been shown in Ref. [29] that, for quantum expander codes [30], neglecting to measure a large percentage of generators could be reasonably tolerated; however, the model considered in that work was narrow in scope, considering only a phenomenological noise model and neglecting the problem of embedding the codes. It is therefore unclear whether such codes lend themselves well to physical implementations. Nonetheless, the authors' results on partial error correction were optimistic and have motivated the investigation of the more realistic architecture developed in this work.

We propose and benchmark a realistic bilayer architecture suited for near- to medium-term superconducting devices and other platforms with restricted qubit movement. We find that the recently introduced bivariate-bicycle (BB) qLDPC codes [18], coming from the larger family of generalized bicycle qLDPC codes [31], are well suited for both the stacked model and the bilayer architecture. These codes have natural embeddings into $\mathbb{Z}^2$ where the generators have a repeated structure, and in some instances, a majority of the generators are geometrically small. The first property makes them amenable to a parallel syndrome-measurement scheme using routing with fast local operations and classical communication (LOCC) and the second property makes them good candidates for reducing overhead using the stacked model. More generally, we develop bounds on how quickly syndrome extraction can be performed in this manner and provide an algorithm to do so. Overall, we find that over multiple rounds of decoding, BB codes implemented in this architecture have error-correction performance comparable to that of the standard (rotated) surface code, albeit only when the parameters in the error model lie in certain regimes.

Our work stands as an alternative architecture that may be more practical for near-term quantum computers without the ability to move qubits. As such, it is not comparable to schemes such as Ref. [23,24], which allow for qubit movement. Several recent works have also proposed layered architectures [17,18]; however, their motivation is in minimizing the number of crossings in the two-qubit gate connectivity. They achieve this through the use of long-range connections, the elimination of which is the main imposed constraint of our work. References [19,20] have presented asymptotically well-performing concatenated schemes that use only local connectivity; however,

the required overheads likely make them infeasible for near- and medium-term quantum computers. In particular, in Ref. [20], it is estimated that approximately 600 physical qubits would be needed per each logical qubit, which is an order of magnitude more than what our architecture needs to implement the $[\![144, 12, 12]\!]$ Gross code [18], with approximately 48 physical qubits per logical qubit. Most closely comparable to our work is Ref. [15], which has aimed to implement quantum expander codes with local connectivity by using a similar teleportation-based scheme. Whereas the authors arrived at a negative result, the innovations in code choice, partial error correction, and syndrome extraction using entanglement purification presented here allow us to obtain more favorable performance. In general, our approach may be easier to implement, as it only requires a bilayer architecture, local connectivity, and relatively few qubits. Of course, it also comes with challenges, which we discuss later.

The paper is structured as follows. In Sec. II, we give the necessary background on quantum error correction and introduce the architecture and routing assumptions that we consider throughout the work. We also review the stacked model and motivate the use of masking. In Sec. III, we develop lower bounds on the routing time for our specific routing model and provide a greedy algorithm to use in implementations. In Sec. IV, we develop an error-correction protocol built on a bilayer architecture and then culminate with circuit-level simulations comparing the performance with the rotated surface code. We conclude in Sec. V with a discussion.

## II. BACKGROUND

### A. Quantum error correction

Quantum error-correcting codes [32] are believed to be necessary in order to run high-fidelity computations on noisy quantum computers. Without them, errors would accumulate throughout the course of a circuit and render the output unreliable. At a high level, quantum error-correcting codes allow us to redundantly encode quantum information in a subspace of the full $2^n$-dimensional Hilbert space and occasionally check to see if errors have caused the information to leave this logical subspace.

*Stabilizer codes* [33,34] are a class of quantum error-correcting codes defined by their *stabilizer*, an Abelian subgroup of the Pauli group on $n$ qubits that leaves the code space invariant. Equivalently, the code space of a stabilizer code is the joint $+1$ eigenspace of the generators of the stabilizer $S = \langle S_1, S_2, \ldots, S_r \rangle$. For a quantum $[\![n, k, d]\!]$ code with $n$ physical qubits, $k$ logical qubits, and distance $d$, the number of linearly independent generators is $r = n - k$. A stabilizer code is considered to be a quantum low-density parity-check (qLDPC) code if each qubit is in the support of at most $c$ stabilizer generators and each generator has weight at most $c$, where $c$ is a constant independent of $n$.
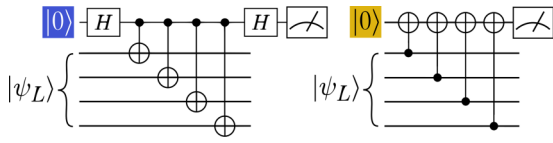
FIG. 1.   Circuits for measuring the eigenvalue of an $X$-type generator (blue) and a $Z$-type generator (yellow). The $Z$-type measurement presented here is a variation from the standard circuit that uses controlled-$Z$ (CZ) gates.

A stabilizer code is said to be a Calderbank-Shor-Steane (CSS) code [35,36] if each generator is a tensor product of $X$ and $I$ or a tensor product of $Z$ and $I$. Although the surface code is LDPC, the encoding rate $k/n$ vanishes in the limit as $n \to \infty$, contributing to its high overhead. Alternative qLDPC codes have asymptotically constant encoding rates while maintaining or improving the $\Theta(\sqrt{n})$ distance scaling of the surface code [37–43].

From a stabilizer description of a quantum error correcting code, one can define its *Tanner graph* $T(S) = (V_q \sqcup V_S, E)$. There is a vertex $q \in V_q$ for each data qubit and a vertex $s \in V_S$ for each stabilizer generator. Two vertices $q \in V_q, s \in V_S$ share an edge $(q,s) \in E$ if the generator $s$ acts nontrivially on qubit $q$. The Tanner graph of a qLDPC code has degree at most a constant $c$.

To determine whether the encoded quantum information has left the logical subspace, the eigenvalues of the stabilizer generators are measured. There are several ways to do this. The circuits depicted in Fig. 1 provide one of the most straightforward approaches, which we use throughout the paper. As the $n$ data qubits are assumed to be in a code state, we expect a $+1$ result when the ancillary check qubit is measured. A $-1$ result indicates an error that anticommutes with that specific generator. These measurement results constitute a classical syndrome that is then used as input to a decoding algorithm to correct the errors.

### B. Architecture

In this work, we consider an architecture in which qubits are located on the vertices of an $M \times M$ grid, where $M = \Theta(\sqrt{n})$. As is natural for current superconducting quantum computing platforms, we assume that two-qubit gates can only be performed between neighboring qubits on the grid. Any two-qubit gate that interacts qubits that are not neighboring is considered a *long-range* gate. Circuits that do not have access to long-range gates are called *2D local circuits* and architectures that are restricted to these circuits are called *2D local architectures*. This definition generalizes to architectures based on graphs other than the grid: given a connectivity graph $G = (V, E)$ with data qubits located on the vertices, the only allowed two-qubit gates are those between qubits $u, v \in V$ that share an edge $(u, v) \in E$. Similar restrictions arise if we disallow the slow movement of atoms in neutral-atom devices, in which case the

only available two-qubit gates are those performed through Rydberg-Rydberg interactions. This leads to an architecture that can perform entangling gates on qubits that are some distance $R$ away, where $R$ depends on the capabilities of the device. We do not investigate this ability in this work but we discuss it in Sec. V.

Implementing general quantum circuits on real architectures requires compilation into a form that respects the connectivity constraints of the device. For the 2D local architecture that we consider here, performing two-qubit operations on qubits that are not adjacent requires permuting them to be so. Doing this with SWAP gates requires a circuit depth proportional to the distance between the qubits. To implement stabilizer-generator measurements such as those shown in Fig. 1, this means that each data qubit must be moved to a position at which it can interact with the check qubit, so one must wait for these permutations to complete before the eigenvalue can be measured. This somewhat defeats the purpose of using qLDPC codes, since a single syndrome can no longer be extracted with a constant-depth circuit. As such, it is infeasible to perform long-range stabilizer-generator measurements in this way and we instead focus on an alternative method.

### C. Teleportation routing

*Routing* is the task of permuting packets of information, or tokens, on the vertices of a graph, using only interactions on edges of the graph. In quantum routing, the tokens are qubits and the graph is specified by the connectivity constraints of the architecture. Classical approaches to routing are typically built from SWAP gates [44–46], which can also be applied naturally to routing quantum data [47,48]. However, more general quantum operations can enable faster routing. In particular, measurement and classical feedback enable the use of entanglement swapping to distribute entanglement and perform quantum teleportation, which can achieve speed-ups over SWAP-based routing for many permutations and underlying graphs [49–51], with applications including error correction [52].

We assume the *LOCC routing* model described by Devulapalli *et al.* [49], in which arbitrary single-qubit and disjoint two-qubit quantum gates can be implemented in a single time step, and we have access to fast single-qubit midcircuit measurements and fast classical control of single-qubit gates. Additionally, there are a constant number of ancillary qubits for each data and check qubit, connected as attached ancillas [53,54] or through stacked vertical layers (see Sec. IV B). In LOCC routing, we can perform protocols such as entanglement swapping [55] and teleportation in constant depth. A specialization of LOCC routing that focuses on qubit and gate teleportation [56] is teleportation routing. During a single round of teleportation, we perform parallel entanglement swapping along multiple teleportation paths. Each vertex can be involved

in at most a constant number of paths, as we allow a constant number of ancillary qubits per vertex. In this work, we assume only one ancilla per data qubit and use the stacked-vertical-layers model. This model allows direct implementation of gates between ancillas and their corresponding data qubits, as well as between ancillas the data qubits of which are also directly connected [see Fig. 2(b)].

To perform long-range two-qubit gates, it is not necessary to actually teleport the participating qubits to adjacent locations; instead, it suffices to use the teleportation paths to implement a long-range gate with gate teleportation. The circuit shown in Fig. 2(a) allows us to implement arbitrarily long CNOT gates in constant quantum depth, avoiding the depth overhead of SWAP routing and any need to reverse the operation. At the cost of utilizing ancillary qubits, this lets us extract the syndrome of a single nonlocal generator using only a constant-depth circuit.

### D. Stacked model

The stacked model has recently been introduced as a potential avenue to reduce overhead when implementing qLDPC codes in architectures restricted to 2D local gates [13,29]. In the stacked model, the stabilizer generators of a quantum error-correcting code are partitioned into several layers depending on the size of the ball containing the qubits in its support. The lowest layer of the stack contains generators that are local and the higher layers contain nonlocal generators the interaction radius of which is some function of the system size. For certain codes, most of the generators are located at the bottom of the stack, i.e., are mostly local, whereas only a small fraction are large. When restricted to 2D local gates, the set of nonlocal generators takes much longer to route and measure than the local generators. Measuring the nonlocal generators less frequently than the local ones could significantly shorten the syndrome-extraction time, at the cost of potentially reduced error-correction capabilities. Note that the layers in the stack do not correspond to physical layers on hardware; instead, they are a conceptual tool for partitioning the generators into sets based on their geometric size.

The concept of *masking* [29,57] formalizes using an incomplete set of generators to perform error correction. Measuring a subset of stabilizer generators corresponds to choosing a subgroup of the stabilizer $T \subseteq S$ and the stabilizer generators that are not measured, $S\backslash T$, are considered to be masked. Error-correction performance may be degraded since the resulting syndrome may have less information about the error than would be available by measuring the full set of generators. During a circuit with $t = 1, \ldots, \tau$ error-correction rounds, we specify a subgroup $T_t \subseteq S$ for each round; or, equivalently, we specify the generators of $S\backslash T_t$ that are masked. For generators that were previously masked, unmasking them adds them

into the current subgroup and their eigenvalues are able to be measured. A single generator may be masked and subsequently unmasked many times over the course of a circuit.

An important consideration for this model is the specific assignment of physical qubits in the architecture to data and check qubits in the code, which can be considered a type of qubit placement [58] or qubit allocation [59]. This assignment can be thought of as an *embedding* of the Tanner graph of the code in the architecture, where an embedding for a graph $G = (V, E)$ is a map $\eta : V \to \mathbb{Z}^D$. As an example, the Tanner graph for the surface code has a natural embedding into $\mathbb{Z}^2$ that allows for all of its generators to act on qubits within a constant radius; however, one could instead assign data and check qubits to physical qubits randomly, yielding generators that still have weight 4 but are no longer local. The difficulty of implementing syndrome-extraction circuits is closely related to the chosen embedding. In Sec. IV A, we discuss the embedding problem for a specific class of codes.

To study the impact of nonlocality on the cost of performing syndrome measurements, we must quantify the notion of generator size and size frequency. We parametrize the size of a given generator as $M^\gamma$, where $0 \leq \gamma \leq 1$ and $M$ is the linear size of the grid. For local generators, $M^\gamma = O(1)$ implies a constant interaction radius, while the largest generators can have interaction radii $\frac{\sqrt{2}}{2}M \in \Theta(M)$ (i.e., $\gamma = 1$). For stabilizer codes, the number of independent stabilizer generators $r$ is related to the number of physical and logical qubits in the code as $n - r = k$. For constant-rate codes, there are thus $O(n) = O(M^2)$ independent generators, which can be parametrized like $M^{2\beta}$, with $0 \leq \beta \leq 1$. With $\beta = 1$, we are considering the problem of measuring every generator and with $\beta < 1$ we only consider some subset. We can describe the set of generators as a whole by defining a function $f(\gamma)$ to characterize the distribution of generators having size $M^\gamma$. The only constraint on $f(\gamma)$ is that it is a valid probability distribution over the domain of $\gamma$, $\int_0^1 f(\gamma)d\gamma = 1$. In practice, $f(\gamma)$ will depend on the architecture, embedding, and parameters of the code family of interest [13,14].

A rough estimate of the amount of work required to perform the syndrome-extraction circuits for a given set of generators is simply to count the two-qubit gates, which in many cases is the leading contributor to the error budget. In our routing model, this value is proportional to the total length of the teleportation paths when implementing long-range CNOT gates, which can be approximated as

$$\text{total path length} \approx M^2 \int_0^1 f(\gamma)M^\gamma d\gamma. \tag{1}$$

Here, the $M^2$ factor comes from the fact that there are $O(M^2)$ generators to measure in total and a single generator of size $\gamma$ requires a path length of $M^\gamma$. If we choose to
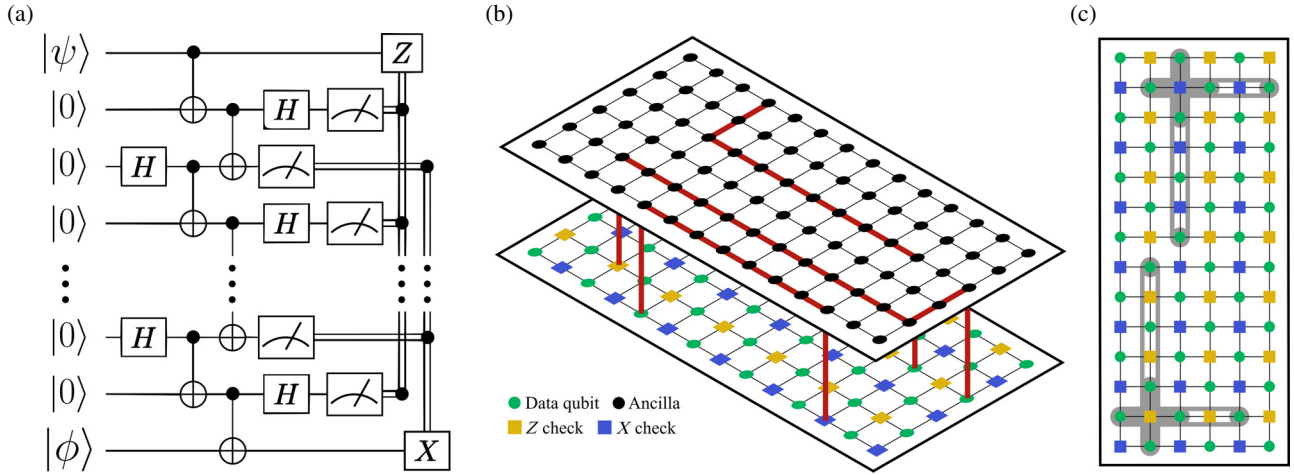
FIG. 2. (a) The circuit to teleport a controlled-NOT (CNOT) gate through a chain of $n$ qubits using only 2D local gates. The depth of the circuit is constant regardless of the length of the chain. (b) The proposed architecture to implement nonlocal high-rate qLDPC codes using only 2D local gates. The architecture consists of two qubit layers: the bottom layer contains the data qubits and ancilla qubits allocated to perform syndrome measurements, while the top layer contains extra ancilla qubits used to perform long-range CNOT gates. Each layer has only 2D local connections and the only connections between the layers are between qubits that are vertically adjacent. To perform a CNOT gate on two spatially distant qubits, the circuit from (a) is used along the paths of qubits highlighted in red. Multiple long-range CNOT gates may be performed in parallel, as long as the paths act on disjoint sets of qubits. (c) An example of the embedding for a $[\![42, 12, 2]\!]$ BB (error-detecting) code constructed with $\ell = 7, m = 3$ and by matrices $A = 1 + y^2 + y, B = 1 + x^5 + x$. The check structure, which is identical for all checks of both types up to mirroring, translation, and boundary conditions, is highlighted in gray.

only measure generators below a certain size $\gamma'$, this corresponds to simply evaluating the integral up to $\gamma'$. We might also want to consider measuring the smallest $x\%$ of generators, in which case one can solve $x = 100 \int_0^{\gamma'} f(\gamma) d\gamma$ to find the appropriate value of $\gamma'$ and then proceed in the same way.

## III. ROUTING BOUNDS

Previous work by Delfosse *et al.* [15] has developed lower bounds on the depth of Clifford circuits required to measure commuting Pauli operators. In this section, we derive similar bounds, taking advantage of additional information about the geometric size of the operators. These bounds do not hold in general, but are instead specific to the teleportation routing model discussed in Sec. II C. We assume that there is a fixed layout of the data and check qubits that gives rise to a specific generator-size distribution $f(\gamma)$. This is to avoid scenarios such as scrambled surface codes, where the difficulty of implementing the syndrome-extraction circuits could be greatly reduced by permuting the qubits.

*Claim 1.* Let $C$ be a 2D local circuit measuring $M^{2\beta}$ commuting Pauli operators the radii of which are greater than $M^{\gamma}$ after embedding them in an $M \times M$ grid. Then, for teleportation routing,

$$\text{depth}(C) = \Omega\left(M^{2\beta+\gamma-2}\right). \qquad (2)$$

*Proof.* In our routing model, the maximum total length of the teleportation paths in a single time step is $O(M^2)$, since only a constant number of ancillary qubits per data qubit are allowed, and there are $\Theta(M^2)$ edges in the grid graph. The cost of measuring an operator of size $\Omega(M^{\gamma})$ is dominated by implementing the long-range CNOT gate between its two furthest qubits. Although this can be done in constant depth using a dynamic circuit [Fig. 2(a)], it requires a teleportation path of length $\Omega(M^{\gamma})$. Consequently, routing and measuring this one operator uses $\Omega(M^{\gamma})$ edges of the $O(M^2)$ available edges. Measuring all $M^{2\beta}$ operators thus requires $\Omega(M^{2\beta+\gamma})$ edges. In the best case, we utilize all available edges in each circuit layer, giving a circuit depth of $\Omega(M^{2\beta+\gamma-2})$. ∎

In practice, it will often be the case that the edges are not optimally used, as illustrated in Fig. 3. We can extend this idea to the general case of an arbitrary distribution of generator sizes.

*Claim 2.* Let $C$ be a 2D local circuit measuring $M^{2\beta}$ commuting Pauli operators the radii of which follow a probability distribution $f(\gamma)$ after embedding them in an $M \times M$ grid. Then, for teleportation routing,

$$\text{depth}(C) = \Omega\left(M^{2\beta-2} \int_0^1 f(\gamma) M^{\gamma} d\gamma\right). \qquad (3)$$

*Proof.* Just as in Claim 1, we can lower bound the circuit depth by summing the lengths of the teleportation
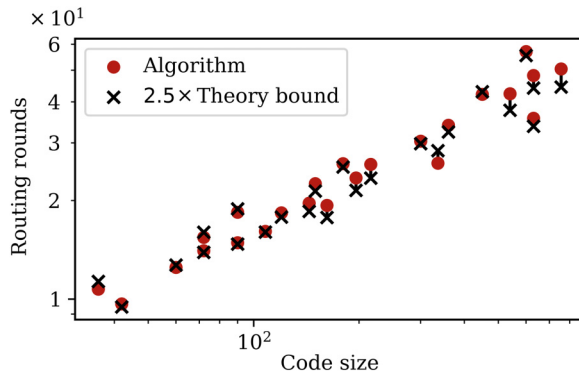
FIG. 3. The depth from greedy routing versus 2.5 times the theoretical optimal depth to route the $X$-type generators using a single layer of ancillary qubits. The code examples are drawn randomly from the family of BB qLDPC codes (see Sec. IV A).

ALGORITHM 1. Greedy routing.

1: **while** there are still operators to measure **do**
2:     Sort the operators in decreasing order according to how many of their qubits have completed routing.
3:     **for** incomplete operator $o_i = 1, 2, \ldots$ **do**
4:         **for** qubits $j = 1, 2, \ldots$ of operator $o_i$ **do**
5:             Use breadth-first search to find a teleportation path for qubit $j$ to the corresponding readout ancilla qubit.
6:             If no path exists, continue.
7:         **end for**
8:     **end for**
9:     Perform long-range entangling gates on qubits that found a teleportation path.
10:     Measure the readout qubit of operators that have completed routing.
11: **end while**

paths required to measure the set of operators. We now have operators of different sizes, where the fraction of operators of a certain size is determined by the probability distribution $f(\gamma)$.

Thus, for a given $\gamma$, there are a number of operators proportional to $f(\gamma)M^{2\beta}$ that each require $M^\gamma$ edges to measure. Since $0 \leq \gamma \leq 1$, the total teleportation path length needed to route and measure every operator is

$$M^{2\beta} \int_0^1 f(\gamma) M^\gamma d\gamma. \qquad (4)$$

Since we again have $O(M^2)$ edges in the grid available in each layer of the circuit, the total circuit depth is lower bounded as in Eq. (3), as desired. ∎

### A. Greedy routing

SWAP routing is a straightforward approach to compiling circuits for quantum hardware with interaction constraints. In practice, this can be done using an algorithm that tries to perform the circuit using as few SWAP gates as possible [47,60–62]. As midcircuit measurement and long-range entanglement generation become more reliable [63], teleportation routing may become a more viable option to move qubits and perform long-range gates. Here, we present a simple greedy algorithm to route an arbitrary set of operators under the routing and architecture assumptions of Secs. II C and II B, respectively. An operator consisting of a tensor product of single-qubit Paulis, such as a stabilizer generator, can only be measured once each qubit in its support has been routed. That is, a teleportation path is prepared and a long-range entangling gate is applied between the qubit and a readout ancilla qubit. Once all required gates have been applied, the operator is said to have completed routing and the readout qubit can be measured to obtain the eigenvalue of the operator. The algorithm is described below in Algorithm 1.

The circuit operations of a single iteration can be executed in parallel, so each iteration performs only a constant-depth circuit. Therefore, the total circuit depth of the routing procedure is proportional to the number of iterations. Instead of minimizing the gate count, the intent of this algorithm is to minimize the circuit depth—and saturate the bound of Claim 2—by maximizing the usage of teleportation paths. This is only possible if the partial measurements between iterations commute, such as when measuring the generators of a single type in a CSS code, in the standard surface-code syndrome-extraction circuit [64] or in the depth-7 BB-code measurement circuit [18]. The syndrome-extraction circuits that we use route every $Z$-type check and then route every $X$-type check.

To benchmark the performance of the algorithm, we draw random examples of BB codes (see Sec. IV A) and route the $X$-type generators while restricted to a single layer of ancillary qubits. For comparison, we compute the optimal routing depth according to Claim 2. In Fig. 3, we show the results of these simulations, providing evidence that the greedy routing algorithm nearly saturates Eq. (3). To obtain the constant multiple in Fig. 3, we consider the smallest eight code instances and perform a fit between the asymptotic lower bound and the depth returned from the greedy routing algorithm. This constant times the theory lower bound matches closely with the routing time of small code instances; however, we begin to see the algorithm routing depth deviating as we increase the block length, indicating nonoptimal performance. For code sizes of practical interest, this algorithm may be a viable option to optimize teleportation routing. Certain codes, such as the BB codes that we discuss in Sec. IV, have additional structure that allows us to manually find routing schedules that outperform those found by the greedy algorithm.

## IV. BILAYER ARCHITECTURE

### A. Bivariate-bicycle codes

In this work, we investigate the recently introduced BB qLDPC codes [18], which come from the wider family of generalized bicycle codes [31]. Let $\mathbb{I}_\ell$ be the $\ell \times \ell$ identity matrix and let $S_\ell$ be the $\ell \times \ell$ cyclic permutation matrix, which is obtained by shifting the columns of $\mathbb{I}_\ell$ one position to the right. Also let

$$x = S_\ell \otimes \mathbb{I}_m \quad \text{and} \quad y = \mathbb{I}_\ell \otimes S_m \quad (5)$$

for integers $\ell, m$. We then define two matrices,

$$A = A_1 + A_2 + A_3 \quad \text{and} \quad B = B_1 + B_2 + B_3, \quad (6)$$

where $A_i$ and $B_i$ are powers of $x$ or $y$. Here, we perform all arithmetic over $\mathbb{Z}_2$. Using $A$ and $B$, we can construct the CSS-type BB code $QC(A, B)$ with $X$- and $Z$-parity checks that, respectively, take the form

$$H_X = [A|B] \quad \text{and} \quad H_Z = [B^T|A^T]. \quad (7)$$

To define a valid stabilizer code, we require that all $X$-type checks commute with all $Z$-type checks, which translates to the condition $H_X \cdot H_Z^T = AB + BA = 0$. Since $[x, y] = 0$, this condition is satisfied.

For certain choices of $A_i$ and $B_i$, the resulting BB code has an embedding into $\mathbb{Z}^2$ that yields checks that act on four nearest-neighbor qubits and two distant qubits (see Appendix A). Another useful property of generalized bicycle codes is the repeated parity-check structure: given one check, other checks of the same type can be obtained with vertical and horizontal shifts on the grid, up to periodic boundary conditions. Opposite-type checks are obtained by mirroring and again performing horizontal and vertical shifts. In Fig. 2(c), we show an example of an embedding for a $[\![42, 12, 2]\!]$ code constructed with $\ell = 7$ and $m = 3$ and by matrices $A = 1 + y^2 + y$ and $B = 1 + x^5 + x$. The check structure for the weight-6 $X$- and $Z$-type generators is indicated by the gray outline.

These natural embeddings make it straightforward to search for codes where the check structure is geometrically small. While the checks are not entirely local due to the two nonlocal qubits in their support, appropriately choosing $\ell$ and $m$ can make the periodic boundary conditions induce generators that are comparatively much larger. This can be done by letting $\ell \gg m$, as illustrated in Fig. 10. In the resulting generator distribution, the majority of the checks are geometrically small. In the context of the stacked model, the generators that are induced by the boundary conditions are those that are measured less frequently.

Table I lists BB codes found by computer search which, through simulations similar to those of Ref. [29], display good numerical performance. For each code, every valid

embedding has been simulated in a simplified version of the protocol in Sec. IV D in order to find the embedding that yields the best masked error-correction performance. Choosing an embedding has determined the percentage of generators induced by the long boundary. This percentage is listed in Table I in the "Mask percentage" column. To our knowledge, the codes presented here are new, with the exception of the $[\![144, 12, 12]\!]$ code, which has been reported in Ref. [18].

### B. Syndrome-extraction circuits

As detailed in Sec. II B, the main difficulty in implementing nonlocal qLDPC codes on 2D local architectures is the need to perform nonlocal two-qubit operations. To address this issue, we propose a physical implementation based on the teleportation-routing model described in Sec. II C. The architecture, as depicted in Fig. 2(b), consists of two layers of qubits. The bottom layer contains the data qubits and ancillary qubits to perform syndrome measurements (check qubits), laid out using an embedding that maximizes the decoding performance while minimizing the number of long-range generators. The top layer contains ancilla qubits to aid in the implementation of long-range CNOT gates. In each layer, the only allowed two-qubit operations are between neighboring qubits and operations between layers are only allowed between qubits that are vertically adjacent, i.e., at the same $(x, y)$ location.

A bilayer architecture is a feasible design requirement for several types of quantum computers. As discussed in Ref. [18], it is difficult, yet not unreasonably so, to modify the current generation of superconducting hardware to support a second layer. In movement-restricted neutral-atom devices, one option is to use dual-species Rydberg arrays [66–68], where the data layer is made up of one species and the ancilla layer is made up of the other. Alternatively, for single-species arrays, it may be practical to store multiple qubits per atom, using a combination of nuclear and electronic [69,70] or motional qubits [71].

To implement a CNOT gate between a data qubit and a distant check qubit, we use the constant-depth circuit shown in Fig. 2(a). A number of ancilla qubits equal to the length of the CNOT gate are needed and so qubits from the upper layer are utilized, as illustrated in Fig. 2(b). Multiple long-range CNOT gates may be performed in parallel as long as the paths act on disjoint sets of qubits. Given a set of CNOT gates to perform, an order that attempts to minimize the total depth of the circuits can be found using the greedy routing algorithm introduced in Sec. III A. Alternatively, we can utilize the repeated check structure of the BB codes to manually come up with highly parallelized orderings; Fig. 4 shows an example of how the Bell pairs needed for the long-range CNOT gates (red highlighted paths) can be implemented in parallel (see also Fig. 9). The "Routing steps" column in Table I indicates the number of routing

TABLE I.   Examples of BB qLDPC codes found through a computer search. The code distances have been computed using the QDistRnd GAP package [65], with 1000 information sets and MINDIST = 0 to obtain the actual distance. The "Embedding" column reports the specific embedding into $\mathbb{Z}^2$ used for that code (see Appendix A). The "Mask percentage" column denotes the percentage of generators that are "large," i.e., induced by the long boundary and masked during a portion of the error-correction rounds. The "Routing steps" column indicates the number of routing rounds required to route, purify, and measure the short-range and long-range generators, respectively. Algorithm 1 has not been used to determine the circuits; instead, the repeated generator structure of the BB codes has allowed us to find circuits by hand. The actual circuit depth is 11 times greater due to the Bell-pair generation (depth 6), purification (depth 2), and implementation of the long-range CNOT gate (depth 3).

| $[[n,k,d]]$ | $\ell, m$ | $A$ | $B$ | Embedding | Mask percentage (%) | Routing steps |
|---|---|---|---|---|---|---|
| $[[72,8,6]]$ | 12,3 | $x^9 + y^1 + y^2$ | $1 + x^1 + x^{11}$ | $\langle A_2 A_3^T, B_1 B_2^T \rangle$ | 25 | 11,6 |
| $[[90,8,6]]$ | 9,5 | $x^8 + y^4 + y$ | $y^5 + x^8 + x^7$ | $\langle A_2 A_3^T, B_2 B_1^T \rangle$ | 22.22 | 9,5 |
| $[[120,8,8]]$ | 12,5 | $x^{10} + y^4 + y$ | $1 + x + x^2$ | $\langle A_2 A_3^T, B_1 B_2^T \rangle$ | 25 | 11,6 |
| $[[150,8,8]]$ | 15,5 | $x^5 + y^2 + y^3$ | $y^2 + x^7 + x^6$ | $\langle A_1 A_2^T, B_1 B_3^T \rangle$ | 26.66 | 11,6 |
| $[[144,12,12]]$ | 12,6 | $x^3 + y + y^2$ | $y^3 + x + x^2$ | $\langle A_2 A_1^T, B_1 B_3^T \rangle$ | 33.33 | 12,8 |
| $[[196,12,8]]$ | 14,7 | $x^6 + y^5 + y^6$ | $1 + x^4 + x^{13}$ | $\langle A_2 A_3^T, B_1 B_2^T \rangle$ | 35.71 | 16,15 |

rounds required to route, purify, and measure the short-range and long-range generators, respectively, for these hand-designed orderings. This repeated parity-check structure is also useful for implementing generalized bicycle codes with reconfigurable atom arrays [24] and bosonic cat qubits [21].

Remote CNOT gates implemented in this way have an error rate proportional to the length of the chain. For short distances, the resulting error rate is not much worse than the native two-qubit CNOT error rate; however, larger chains will be prohibitively noisy. To remedy this, we can apply entanglement purification [72,73] to the noisy Bell pairs in the ancilla layer. In Fig. 4, we outline the original purification scheme as proposed by Bennett *et al.* [72]. The protocol uses additional "donor" Bell pairs (pink highlighted paths) to create "source" Bell pairs (red highlighted paths) with higher fidelity. This is done by performing CNOT gates between the ends of the source and donor pairs, measuring the ends of the donor pairs in the computational

basis, and then comparing the measurement results classically. If the results agree, the source Bell pair is kept; otherwise, it is discarded. Averaging over cases in which the source Bell pair is kept, it has a higher fidelity than an unpurified pair; however, in the cases in which it is discarded, the corresponding long-range CNOT gate cannot be performed. We have the option of either reattempting the purification process, implementing the long-range CNOT with the flawed Bell pair, or giving up on the long-range CNOT gate (and ultimately the corresponding generator-syndrome measurement) altogether. Since we already intend not to measure every generator at every error-correction round, this last option is most appropriate. In the context of the bilayer architecture, both donor and source Bell pairs are routed through the ancilla layer. In practice, this means that fewer long-range CNOT gates can be implemented in parallel, since the purification process uses additional teleportation paths.

Although we now have a way to implement long-range CNOT gates, measuring every stabilizer generator in this manner incurs additional overhead (see Sec. IV D). Instead, we can reduce the time overhead by applying the stacked model and choosing to measure the costly large generators less frequently than the smaller ones. The frequency with which the long-range generators are measured can be tuned, with more frequent measurements potentially correcting more errors but increasing the time needed to implement error correction.

In the phenomenological noise model, depolarizing errors are introduced with probability $p$ only at the beginning of each error-correction round. The syndrome is then noiselessly computed using the parity-check matrix and the randomly drawn errors. Additional errors may be introduced to the syndrome to represent measurement errors. To correct the qubit errors, the syndrome is given as input to a decoder that attempts to deduce the most likely error. Decoding is considered a success if the guessed error is equivalent to the actual error up to a stabilizer element.
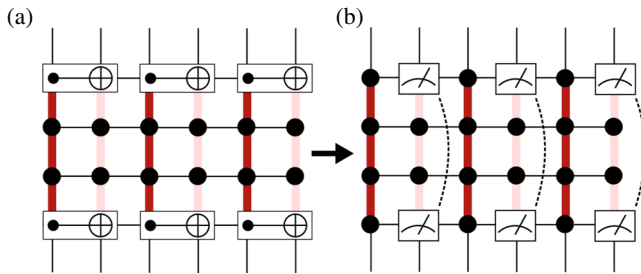


FIG. 4.   Implementing multiple long-range Bell pairs in parallel for a BB code. The "source" red highlighted Bell pairs are purified using the Bennett protocol [72]. (a) CNOT gates are performed between each end of the source and the pink "donor" Bell pairs. (b) Each end of the donor Bell pair is measured and the results are compared classically. If the measurements agree, the source Bell pair is kept and used; otherwise, it is discarded.

In reality, errors may occur at any operation in the syndrome-extraction circuit, including qubit initialization, single- and two-qubit gates, measurements, and idle locations. To model this, we instead consider the standard circuit-based depolarizing-noise model [3], where for each operation in the circuit, an error is introduced with some probability $p$. For example, an error arising from a CNOT gate is the gate followed by one of the possible 15 nonidentity two-qubit Pauli products on the control and target qubits. Although it is possible to decode circuit-level noise using the same method as for phenomenological noise, it has been shown to be advantageous to instead use a space-time circuit-level decoder [23,74]. Here, the goal is to guess the error at specific locations in the syndrome-extraction circuit. Decoding is considered a success if the guessed errors have the same effect on the logical observables as the actual error.

The input to the space-time decoder is not the syndrome of the error but, rather, the parities of the syndrome measurements between error-correction rounds. In the absence of errors, the syndrome between rounds should be constant, i.e., have parity of zero. A parity of one indicates that an error has occurred at some point in the previous error-correction round. Following the notation of STIM [75,76], we define the *i*th *detector* at time $t$ to be the parity of the syndrome of the current and previous rounds $D_i^{(t)} = \sigma_i^{(t)} \oplus \sigma_i^{(t-1)}$, where $\sigma_i^{(t)}$ is the $i$th bit of the syndrome at time $t$. However, in the stacked model, we have the possibility of neglecting to measure certain generators for some number of rounds, $t_m$. As such, detectors for these generators must compare the parities of the corresponding syndromes $t_m$ rounds apart, $D_i^{(t)} = \sigma_i^{(t)} \oplus \sigma_i^{(t-t_m)}$. Each detector allows us to determine whether errors have occurred in a specific detecting region [76] of the circuit. In Fig. 5(a), we show a simple example of a classical repetition-code circuit with its associated detectors and highlighted detecting regions.

### C. Space-time decoder

To correct for errors in the circuit-level model, we relate the detectors with errors in the circuit by constructing a bipartite graph. Let the detectors over $T$ rounds be the check nodes and let every possible single- and two-qubit error over the circuit make up the bit nodes. A detector and error are connected by an edge if the error causes the detector to activate. As a practical note, many errors have the same action on the detectors and logical observables, so they can be consolidated into a single node. Since each error in this set has the same action on the final logical observables, one can choose an arbitrary representative when checking for decoding success. Similarly, some errors will have no effect on the detectors or logical observables and as such are not included as a node in the bipartite graph. This bipartite graph can be considered the Tanner graph of a classical code and can be decoded
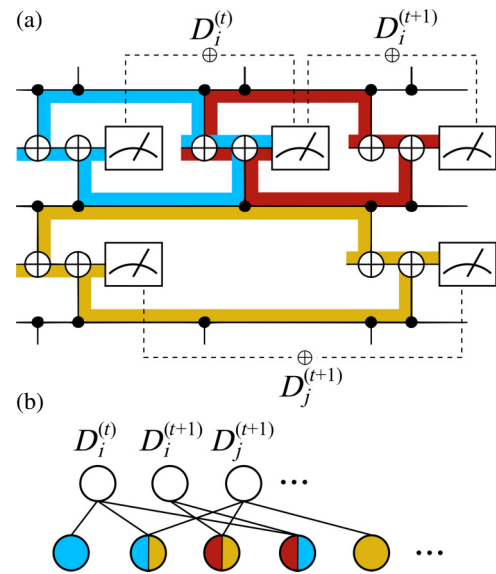


FIG. 5. (a) The detectors for a portion of a bit-flip repetition code. The highlighted regions represent the detecting region [76] of a detector, the set of errors that would cause the detector to be triggered. The corresponding detectors are then the parities of the measurements in that region. Since syndrome $j$ has been masked for a round, the detector now represents the parities of the measurements in the region that spans three rounds. (b) The bipartite space-time decoding graph of the circuit. The check nodes of this graph are the detectors and the bit nodes are possible errors during the execution of the circuit. A detector and error are connected by an edge if the error causes the detector to be activated. Errors on the boundary of two detecting regions cause both detectors to trigger.

by any appropriate decoder to deduce the errors that have occurred. In Fig. 5(b), we show the bipartite-decoding graph corresponding to the circuit of Fig. 5(a). The classes of equivalent errors from each detecting region constitute the bit nodes of the graph and are connected by edges to the appropriate detectors. For a more detailed discussion of the circuit-level noise decoding process, see Ref. [18].

### D. Circuit-level simulations

We now present the results of circuit-level error-correction simulations using the class of BB quantum LDPC codes and the architecture defined in Sec. II B. Previous simulations of BB codes have shown that they have greatly outperformed surface codes in terms of overhead under specific architecture assumptions [18,24]. Here, we show that BB codes implemented with 2D local gates in the proposed bilayer architecture have comparable performance to surface codes that encode the same number of logical qubits and have roughly the same number of physical qubits.

For the following simulations, we use STIM [75] to construct the circuits and build the space-time bipartite graph

used for decoding. As such, we consider a circuit-level noise model in which errors occur independently on different circuit operations. For a physical error rate $p$—in this work, we consider $p = 0.1\%$—single-qubit gates have probability $p/10$ of experiencing the single-qubit depolarizing channel; two-qubit gates have probability $p$ of experiencing the two-qubit depolarizing channel; measurement results have probability $p$ of being flipped; qubit reset operations have probability $p/10$ of preparing the $|1\rangle$ state instead of the $|0\rangle$ state; and idle qubits experience a depolarizing channel with probability $p/50$. The assumed single-qubit, two-qubit, and measurement error rates are comparable to the performance of current ion-trap [26,77,78] and superconducting [79] quantum computers. However, this last condition on the idle-qubit error rate is somewhat optimistic and is around an order of magnitude better than the idle error seen on production devices. We comment on this assumption in Sec. V.

For ease of implementation, we first separately perform circuit-level simulations of the entanglement-purification protocol. The simulation consists of implementing two noisy long-range Bell pairs using a circuit similar to that depicted in Fig. 2(a) and then performing the entanglement-purification protocol of Bennett *et al.* on the two pairs. In this simplest version of the protocol, failures are not reattempted and only a single donor Bell pair is used. Simulating the protocol many times allows us to estimate the probability that the purification protocol succeeds and, if so, the fidelity of the purified Bell pair. In Fig. 6(a), we display the results of these simulations for long-range Bell pairs of different lengths under the circuit-level error model described above.

During syndrome extraction, if the entanglement purification protocol for any of the long-range CNOT gates fails, we mask the corresponding generator instead of reattempting the purifications. We can then estimate the probability that the syndrome of a generator is available, i.e., all the required purifications for that generator succeed. If the purifications do succeed, then we can also estimate the error rate of the resulting long-range CNOT gate from the fidelity of the Bell pair. In the full circuit, we then implement a direct CNOT with this error rate to represent the entire procedure. In Fig. 6(b), we illustrate what this means in practice: assuming that the long-range generators are unmasked every five rounds, the first four rounds have these long-range generators masked (hatched fill). Additionally, due to failures of the entanglement-purification protocol, some short-range generators are also masked, even though we had planned for them to always be available. We note that these random failures are not expected to greatly impact the performance of the code, as it is unlikely that one generator will fail several rounds in a row. Thus, even if there are missed errors, they will likely be corrected when the generator does succeed in routing. In the fifth round, the long-range generators are unmasked
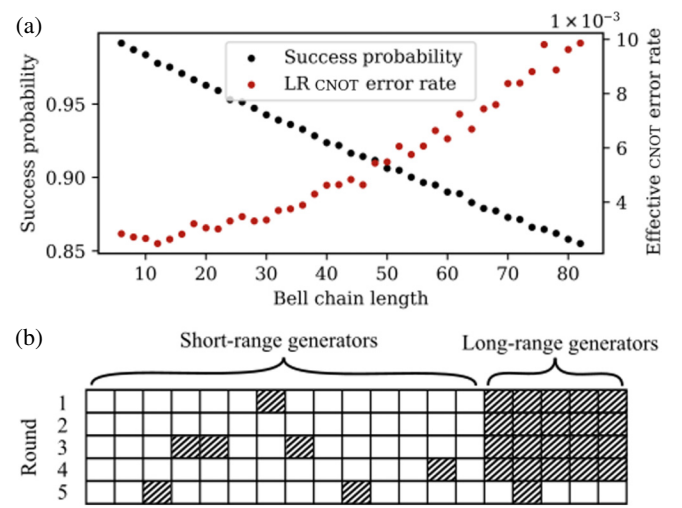


FIG. 6. (a) The results of circuit-level simulations of the entanglement-purification protocol of Ref. [72] for Bell pairs of increasing length. Two long-range Bell pairs are created using a noisy circuit similar to that of Fig. 2(a) and then purified with the noisy circuit depicted in Fig. 4. The success probability of the purification and the resulting Bell purity if successful is shown for 100 000 samples. (b) An example depiction of generator masking (indicated by a hatched fill) over several error-correction rounds being affected by the entanglement-purification protocol failing. In this example, the long-range generators are unmasked after five rounds.

and attempts are made to measure them, but it is only if purifications succeed that can we actually obtain their syndromes. Note that with this simple purification scheme, the long-range generators are less likely to succeed, since the necessary Bell pairs are between more distant qubits and more prone to failure.

For the full error-correction protocol, we begin each circuit with a single noiseless round to initialize the logical subspace. We then perform $t$ noisy error-correction rounds using the syndrome-extraction circuits defined in Sec. IV B. As the short-range generators are easier to measure, we attempt to measure them every round, whereas the costly long-range generators are unmasked and attempted every five rounds. As described above, we additionally mask both the short- and long-range generators with a probability equal to that of at least one of the required purifications failing. In the cases in which all purifications for a single generator succeed, we apply the two-qubit depolarizing channel after each CNOT gate with an error rate equal to that of a long-range CNOT gate performed using a Bell pair of the appropriate distance. Idling error rates are estimated using the number of steps needed to route and purify the source and donor Bell pairs for a given set of generators (see Fig. 9 and Table I). As each step consists of Bell pair generation (depth 6), purification (depth 2), and implementation of the long-range CNOT gate (depth 3), the actual circuit depth is $11\times$ greater. To

represent idling errors, a depolarizing channel is applied at the beginning of each error-correction round to every qubit with probability equal to the total circuit depth times the idle error rate. Additionally, a depolarizing channel is applied to every qubit with probability $p = 0.1\%$ at the beginning of each round. Before measuring the logical observables, we noiselessly extract the full syndrome one last time. The corresponding space-time bipartite graph is then generated and the errors are sampled and decoded.

In this work, we use a decoder based on belief propagation and ordered-statistics decoding (BP OSD) [80–82], which consists of the min-sum BP decoder followed by an order-10 combination-sweep OSD postprocessing step. Performing real-time decoding using BP and higher-order OSD postprocessing may be infeasible within the fast cycle time of superconducting quantum computers; however, it has been shown that good decoding performance for BB codes can be achieved while using less computationally expensive OSD parameters [83].

In Table II and Figs. 11(a) and 11(b), we show the results of these simulations for several codes listed in Table I. As a comparison, we perform the same simulations with the rotated surface code, which has parameters $[\![d^2, 1, d]\!]$. To decode, we follow the same process as described in Sec. IV C but instead use the minimum-weight perfect-matching decoder [84]. As the BB codes encode multiple logical qubits in a single block, multiple copies of the surface code must be used to achieve the same number of logical qubits. If $p_{SC,1}$ is the logical error rate of simulating a single rotated surface code for $t$ error-correction rounds, then $k$ copies of the surface code have a logical error rate

$$p_{SC,k} = 1 - (1 - p_{SC,1})^k. \quad (8)$$

In addition to the logical error rate, another important performance metric is the number of qubits used to achieve it. For the BB codes and the bilayer architecture, this includes the ancillary check qubits as well as the entire routing layer, which for an $[\![n, k, d]\!]$ code uses $4n$ qubits in total. The rotated surface code uses $d^2 - 1$ additional check qubits, which brings the total number of qubits to $2d^2 - 1$ for each copy. The total number of qubits used is listed together with the code parameters in Fig. 11. The error bars on the data points are calculated using the standard error when sampling from a binomial distribution $\sqrt{p_{\log}(1 - p_{\log})/N}$, where $N$ is the number of collected samples. Due to the large number of shots taken, $N \sim 10^5$, the error bars in Figs. 11 and 12 are nearly invisible. Additionally, we plot a fit of

$$p_{\log} = 1 - (1 - \epsilon_L)^t \quad (9)$$

for both the surface and the BB codes, from which we can extract the logical error rate per round, $\epsilon_L$.

TABLE II. The code parameters, the total number of qubits used, and $\epsilon_L$ as extracted from Eq. (9) for the simulations described in Sec. IV D. The code parameters shown in bold correspond to BB-code instances. The code parameters not in bold correspond to copies of the rotated surface code.

| $[\![n, k, d]\!]$ | Qubits | $\epsilon_L$ |
|---|---|---|
| $[\![\mathbf{128, 8, 4}]\!]$ | 248 | $1.4 \times 10^{-3} \pm 1.2 \times 10^{-6}$ |
| $[\![\mathbf{72, 8, 6}]\!]$ | 288 | $1.6 \times 10^{-3} \pm 3.0 \times 10^{-5}$ |
| $[\![\mathbf{90, 8, 6}]\!]$ | 360 | $8.9 \times 10^{-4} \pm 2.0 \times 10^{-5}$ |
| $[\![200, 8, 5]\!]$ | 392 | $2.0 \times 10^{-4} \pm 6.5 \times 10^{-7}$ |
| $[\![\mathbf{120, 8, 8}]\!]$ | 480 | $1.2 \times 10^{-4} \pm 2.0 \times 10^{-6}$ |
| $[\![\mathbf{288, 8, 6}]\!]$ | 568 | $9.5 \times 10^{-5} \pm 2.5 \times 10^{-7}$ |
| $[\![\mathbf{150, 8, 8}]\!]$ | 600 | $5.3 \times 10^{-5} \pm 1.3 \times 10^{-6}$ |
| $[\![392, 8, 7]\!]$ | 776 | $2.0 \times 10^{-5} \pm 1.5 \times 10^{-7}$ |
| $[\![\mathbf{144, 12, 12}]\!]$ | 576 | $1.6 \times 10^{-4} \pm 4.6 \times 10^{-6}$ |
| $[\![300, 12, 5]\!]$ | 588 | $3.0 \times 10^{-4} \pm 9.8 \times 10^{-7}$ |
| $[\![\mathbf{196, 12, 8}]\!]$ | 784 | $7.9 \times 10^{-5} \pm 2.3 \times 10^{-6}$ |
| $[\![432, 12, 6]\!]$ | 852 | $1.4 \times 10^{-4} \pm 3.7 \times 10^{-7}$ |
| $[\![588, 12, 7]\!]$ | 1164 | $2.9 \times 10^{-5} \pm 2.3 \times 10^{-7}$ |

The smallest BB codes encoding $k = 8$ logical qubits are outperformed by surface codes that use fewer physical qubits. However, increasing the block length yields BB codes that surpass the performance of similarly sized surface codes. This is illustrated in Fig. 7, where we see the BB codes achieving a lower logical error rate per round than the surface code while utilizing fewer qubits. Increasing the number of logical qubits to $k = 12$, the BB codes and the proposed architecture immediately outperform the surface codes in terms of logical error rate and space overhead. Compared to 12 patches of a $[\![36, 1, 6]\!]$ rotated surface code using a total of 852 physical qubits and a logical error rate per round of $\epsilon_L = 1.43 \times 10^{-4}$, we find a $[\![144, 12, 12]\!]$ BB code using 576 qubits that matches the performance, with $\epsilon_L = 1.56 \times 10^{-4}$. Additionally, we find a $[\![196, 12, 8]\!]$ code using 784 qubits that outperforms it with $\epsilon_L = 7.89 \times 10^{-5}$. At this scale, the improvements are not so drastic but we expect to see greater overhead benefits as the block length and number of logical qubits increase.

We now vary the interval at which the long-range generators are measured, the results of which are also shown in Fig. 11(c). For the $[\![90, 8, 6]\!]$ code, there are 44 (not necessarily independent) generators of a single type. Using a routing schedule that has been found by hand, all 35 short-range generators of a single type can be routed, purified, and measured in nine steps; whereas it takes five steps to route, purify, and measure the remaining ten long-range generators of the same type. Measuring the 35 short-range and ten long-range generators of the opposite type requires an additional nine and five steps, respectively. This means that measuring the long-range generators every five error-correction rounds requires a circuit depth that is 28.5% shorter than if the long-range generators were
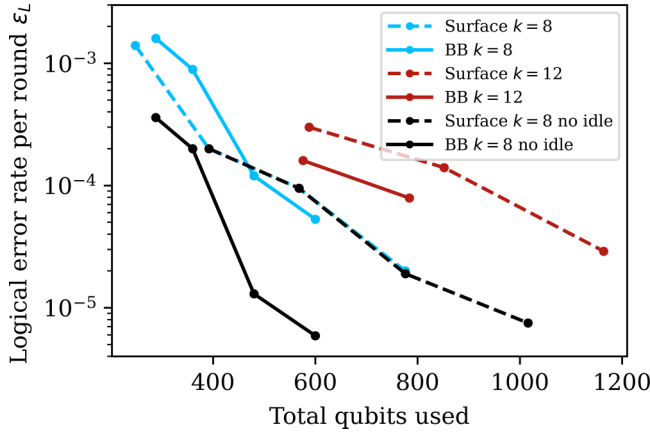
FIG. 7. The extracted logical error rate per round, $\epsilon_L$, as a function of the total number of qubits used (data qubits plus all ancilla qubits) for several BB- and surface-code instances. The data are tabulated in Table II. We also include simulation results in the no-idle error regime, as indicated by the black lines; these results are tabulated in Table III.
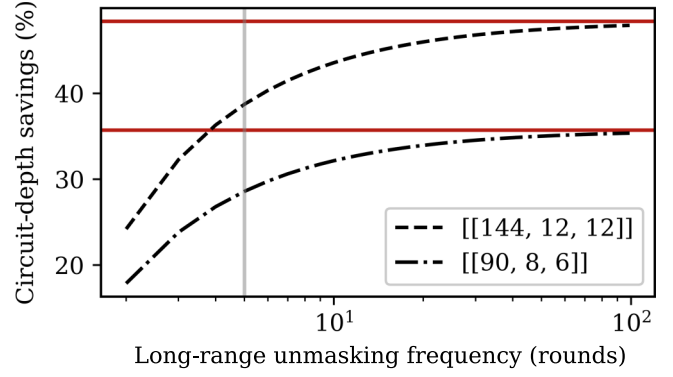


FIG. 8. The percentage change in the circuit depth, compared to a circuit that always measures every generator, as a function of the number of rounds that have elapsed between long-range generator measurements. The horizontal solid red lines indicate the potential maximum reduction in the circuit depth for the two BB-code instances. The gray vertical line highlights the depth savings achieved by measuring the long-range generators every five error-correction rounds, as done in Figs. 11(a), 11(b), and 12.

measured every round $(4 \cdot 2 \cdot 9 + 2 \cdot (5 + 9) = 100$ steps versus $5 \cdot 2 \cdot (5 + 9) = 140$ steps), at a negligible increase in the logical error rate per round from $\epsilon_L = 8.41 \times 10^{-4}$ to $8.92 \times 10^{-4}$. Increasing the size of the code to $[[196, 12, 8]]$, we again see negligible differences in logical error per round performance between the two measurement schedules, from $\epsilon_L = 7.41 \times 10^{-5}$ to $7.89 \times 10^{-5}$, with the additional benefit of a 32.0% decrease in the depth of the syndrome-extraction circuit when measured every five rounds. The two codes consist of 22.22% and 35.71% long-range generators, respectively, yet both remain consistent between long-range measurement intervals. As the block length increases, so does the discrepancy between the measurement times of the short- and long-range generators, increasing the circuit-depth savings. Additionally, this discrepancy would disproportionately introduce more

TABLE III. The code parameters, the total number of qubits used, and $\epsilon_L$ as extracted from Eq. (9) for the simulations described in Sec. IV D, albeit with no idle error. The code parameters shown in bold correspond to BB-code instances. The code parameters not in bold correspond to copies of the rotated surface code.

| $[[n, k, d]]$ | Qubits | $\epsilon_L$ |
|---|---|---|
| **$[[72, 8, 6]]$** | 288 | $3.6 \times 10^{-4} \pm 4.9 \times 10^{-6}$ |
| **$[[90, 8, 6]]$** | 360 | $2.0 \times 10^{-4} \pm 3.3 \times 10^{-6}$ |
| $[[200, 8, 5]]$ | 392 | $2.0 \times 10^{-4} \pm 6.5 \times 10^{-7}$ |
| **$[[120, 8, 8]]$** | 480 | $1.3 \times 10^{-5} \pm 8.2 \times 10^{-7}$ |
| $[[288, 8, 6]]$ | 568 | $9.5 \times 10^{-5} \pm 2.5 \times 10^{-7}$ |
| **$[[150, 8, 8]]$** | 600 | $5.9 \times 10^{-6} \pm 2.4 \times 10^{-7}$ |
| $[[392, 8, 7]]$ | 776 | $2.0 \times 10^{-5} \pm 1.5 \times 10^{-7}$ |
| $[[512, 8, 8]]$ | 1016 | $7.5 \times 10^{-6} \pm 4.3 \times 10^{-8}$ |

errors during the long-range measurement rounds, potentially making it more efficient to measure these large generators even less frequently. However, this behavior is highly dependent on the idle error rate and changing the idle error rate may cause the two curves to deviate.

Achieving more significant reductions in the circuit depth requires measuring the long-range generators much less frequently, as shown in Fig. 8. We display the potential circuit-depth savings for two BB-code instances as a function of how many error-correction rounds elapse between measurements of the long-range generators. The horizontal red lines indicate the maximum potential savings, corresponding to a schedule where the long-range generators are only measured once at the end of the circuit. For example, the $[[144, 12, 12]]$ BB code requires 16 (15) steps to measure the short- (long-) range generators of a single type, which gives a maximum depth saving of 48.4%. When measuring every five rounds, as in Figs. 11(a), 11(b) and 12, we see circuit-depth savings of 38.7%, indicated by the vertical gray line. Measuring the long-range generators very infrequently will significantly degrade the error-correction performance and may not be worth the reduced circuit depth. Instead, it may be more advantageous to measure the long-range generators relatively frequently, e.g., every two to five rounds; in that regime, we still see considerable circuit-depth savings (50–80% of the theoretical maximum) but the impact on the logical error rate is negligible.

Even with the reduced idle error rate that we consider here, idle errors are a significant source of error, especially on rounds where the long-range generators are measured. In Table III and Fig. 12, we perform the same simulations as described above but do not apply idling errors. Due to

their short syndrome-extraction circuit depths, the surface codes are unaffected by the decrease in the idle error. However, we now find that all BB-code instances achieve better logical error rates than surface codes while using fewer physical qubits, as shown in Fig. 7. Indeed, the $[[150, 8, 8]]$ code using 600 physical qubits sees an $8.8\times$ improvement in the logical error rate per round, from $\epsilon_L = 5.31 \times 10^{-5}$ to $5.9 \times 10^{-6}$, and now outperforms eight patches of a $[[64, 1, 8]]$ rotated surface code using 1016 physical qubits with $\epsilon_L = 7.5 \times 10^{-6}$. Achieving negligible idle error rates may not be feasible but this illustrates the regime in which our protocol performs best.

## V. DISCUSSION AND CONCLUSIONS

In this paper, we have presented a bilayer architecture for implementing nonlocal qLDPC codes on quantum devices that are restricted to 2D local gates. We have shown that BB codes are well suited for such an architecture and we have described a parallelizable syndrome-measurement scheme that makes use of the geometric parity-check structure of the codes. Through circuit-level simulations of a multiround decoding protocol, we have found that BB codes attain comparable logical error rates to that of the rotated surface code while using fewer physical qubits. Furthermore, by applying the stacked model and masking, we have achieved a significant decrease in syndrome-extraction time with a negligible impact on the error-correction performance.

However, there are a number of challenges that must be considered in a physical implementation of this protocol. Perhaps the most notable issue is the depth of the circuit required to perform even a single syndrome extraction. Implementing a single long-range CNOT gate requires constructing the long-range Bell pair, purifying it, and using it to implement a CNOT between a data qubit and check qubit. Although several CNOT gates can be implemented in parallel, doing this for the entire set of generators requires tens of routing steps, translating to a physical circuit with depth in the hundreds. One consequence of the depth of the circuit is that our protocol only performs well in the regime of low idle error rate. Furthermore, per Claim 2, as the block length increases, so too does the required routing time and, consequently, the physical circuit depth. This is in stark contrast to the implementation in Ref. [18], where the entire set of generators can be measured with a circuit of depth 7, albeit with the use of long-range connections. These long-range connections are a significant engineering challenge and it is unclear whether implementing high-fidelity gates in this way is feasible.

We do find BB codes where the same parity-check structure is shared between codes of increasing block length. For code families with this property, the number of steps in the syndrome-extraction circuit is constant, so the noise per syndrome-extraction cycle coming from idle error does not increase. However, this also means that the percentage of long-range generators and, by extension, the amount of nonlocality in the code, decreases. In Refs. [13,14], the authors have shown that it is impossible to beat the asymptotic scaling of the surface-code parameters without increasing the amount of nonlocality. Increasing the block length will yield larger $k$ and $d$ but the asymptotic scaling of these codes will approach that of the surface code; however, for finite sizes, we would still expect to see significant space-overhead savings compared to alternative lower-rate codes. Even for BB codes with increasing generator shape, it is feasible that the increased error-correction capabilities will outpace the increase in idle error. In particular, in Fig. 3, we show a power-law relationship between the block length and the routing depth. Assuming that the code is operating below threshold, the exponential suppression in the logical error should be sufficient to handle the increased effective idle error.

Another challenge is that the simple purification protocol presented here does not scale well, as increasing the block length would lead to low-fidelity Bell pairs and a high purification failure rate. Although there are many entanglement-purification protocols that improve the resulting Bell fidelity [85–88], using them would further increase the depth of the syndrome-extraction circuits or require additional ancillary qubits. The one potential saving factor is that the vast majority of the work is done by the upper routing layer to construct and purify the Bell pairs and the two layers interact in fewer than 1/10 of the circuit steps. If it were possible to sufficiently isolate the data layer, akin to what is done in ion traps or reconfigurable atom arrays, it might be possible to achieve the low idling error rates that would greatly improve the performance of the protocol.

If the aforementioned issues can be solved, then scaling up should increase the advantage of qLDPC codes over the surface code. One potential solution is to improve the circuit depth of the protocol. An architectural feature that could accomplish this is the ability to perform two-qubit gates on qubits that are some distance $R$ apart [19]. This is a natural operation on neutral-atom devices, where Rydberg-Rydberg interactions, especially dipolar ones [89], can be quite long range. Furthermore, Rydberg-Rydberg interactions can help with syndrome extraction by naturally realizing long-range generalized multicontrol multitarget CNOT gates [90]. At the same time, such long-range Rydberg-based gates may harm parallelism, since only one such gate can be implemented within the Rydberg-blockade radius at a time. Gates beyond the nearest neighbor could also be feasible in superconducting devices through the use of medium-range couplers or photonic interconnects [91]. When $R$ is a constant, the asymptotic behavior will remain unchanged; however, in practice this would mean that the short-range generators would be much easier to implement. With an appropriate
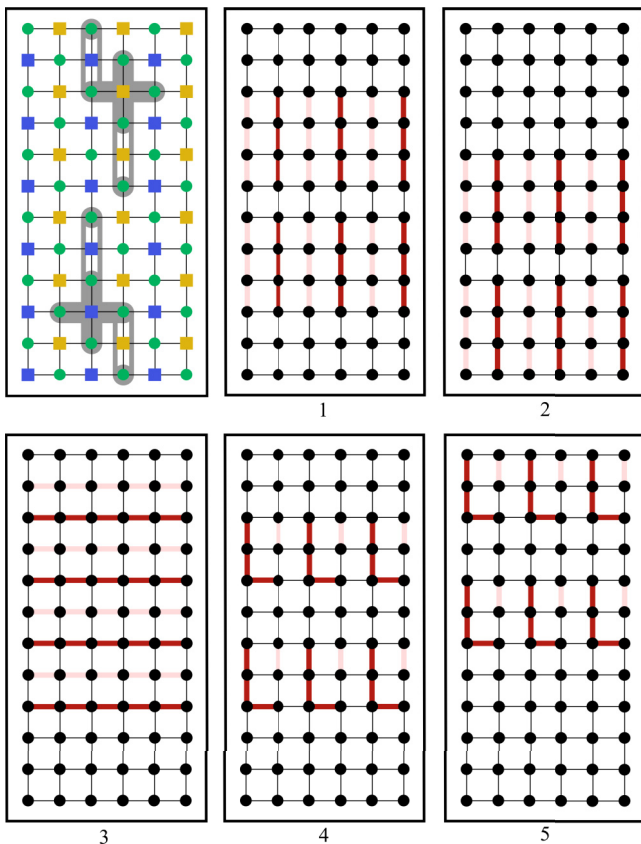
FIG. 9. An example five-step schedule to route and purify the Bell pairs needed to measure the short-range $Z$-type generators of a $[[36,4,4]]$ BB code constructed with $\ell = 6$ and $m = 3$ and by matrices $A = x + y^3 + y^2$ and $B = y^3 + x^5 + x^4$. The first panel shows the structure of the $Z$-type checks (yellow squares) and $X$-type checks (blue squares), as outlined in gray.
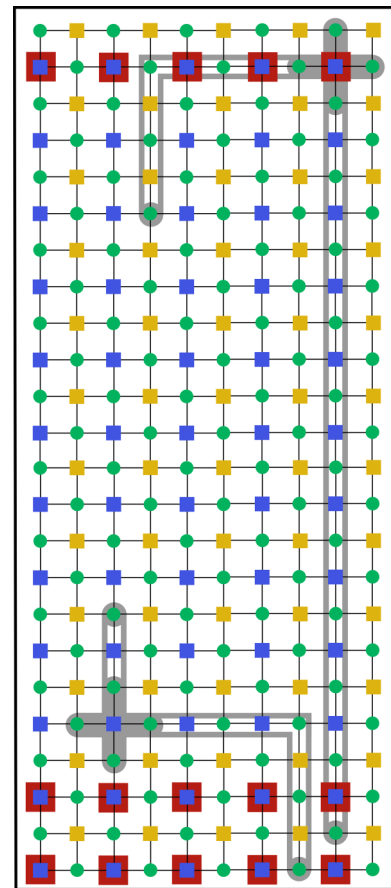


FIG. 10. An illustration of the long- and short-range generators of a single type for the $[[120,8,8]]$ BB code constructed with $\ell = 12$ and $m = 5$ and by matrices $A = x^{10} + y^4 + y$ and $B = 1 + x + x^2$. The qubits contained in the check are highlighted in gray. The checks (blue squares) highlighted in red are the long-range checks, as they cross the long boundary. For both $Z$- and $X$-type checks, there are 12 long-range checks out of a total of 48, yielding a mask percentage of 25%.
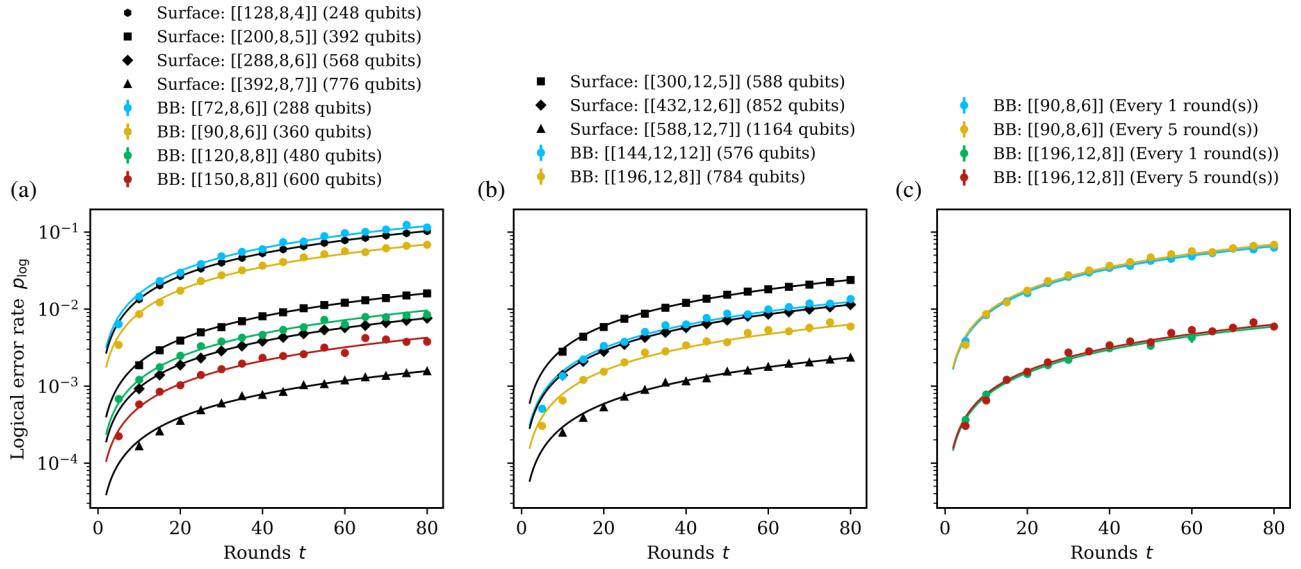
FIG. 11.    (a),(b) The logical error rate of performing $t$ rounds of error correction with BB codes with (a) $k = 8$ and (b) $k = 12$ on the proposed bilayer architecture. The logical error rate of the same simulations using $k$ copies of the rotated surface code is plotted as a comparison. (c) A comparison between the time intervals at which the long-range generators are measured. For the BB codes in (a) and (b), the long-range generators have been measured every five error-correction rounds. A fit of Eq. (9) is also shown, from which we extract the logical error rate per round, $\epsilon_L$.

## DATA AVAILABILITY

The source code and data to generate the figures in the paper are available at Ref. [92].

## APPENDIX: BIVARIATE-BICYCLE-CODE EMBEDDINGS

Here, we briefly describe the conditions for embedding BB codes in a 2D grid (for a more complete discussion, see Ref. [18]).

*Definition A1 (*Definition 1 in Ref. [18]). A code $QC(A, B)$ has a toric layout if its Tanner graph has a spanning subgraph isomorphic to the Cayley graph of $\mathbb{Z}_{2\mu} \times \mathbb{Z}_{2\lambda}$ for some integers $\mu$ and $\lambda$.

This is to say that codes with a toric layout have checks that act on the four nearest-neighbor qubits, and potentially on additional nonlocal qubits. The four nearest-neighbor qubits can be measured using a standard surface-code syndrome-extraction circuit [64], whereas the nonlocal qubits are measured using the proposed protocol. In the following, the order of an element $\mathrm{ord}(M)$ of a multiplicative matrix group is the smallest positive integer such that $M^{\mathrm{ord}(M)} = \mathbb{I}$, where $\mathbb{I}$ is the identity matrix of the same dimension as $M$.

A BB code $QC(A, B)$ depends on choices of matrices $A$ and $B$, as in Eq. (6), the terms of which are powers of $x$ or $y$, defined in Eq. (5). The matrices $x$ and $y$ depend on
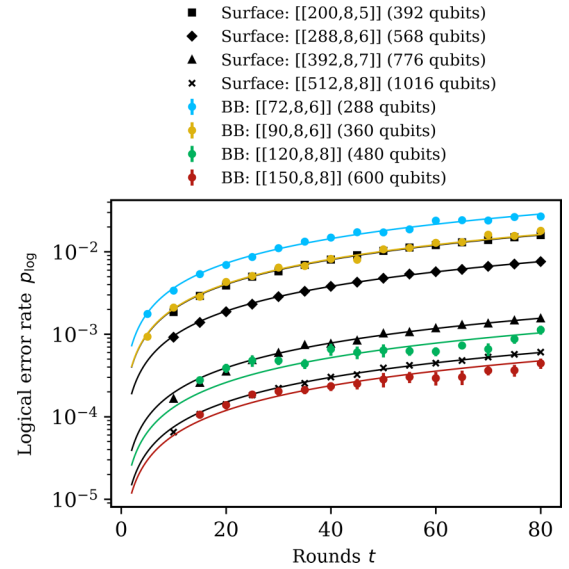


FIG. 12.    The logical error rate of performing $t$ rounds of error correction with BB codes with $k = 8$ logical qubits on the proposed bilayer architecture. For this plot, we consider the case in which the idle error rate is zero. The logical error rate of $k$ copies of the rotated surface code, calculated using Eq. (8), as well as the total number of physical qubits used, is again plotted as a comparison. A fit of Eq. (9) is also shown, from which we extract the logical error rate per round, $\epsilon_L$.

choices of positive integers $\ell$ and $m$, and they correspond to the dimensions of the grid in which the code $QC(A, B)$ is embedded should it satisfy Lemma A1. The $\mu$ and $\lambda$ of Definition A1 are $\ell$ and $m$, respectively. In this toric layout, qubits and checks can be labeled by $\mathcal{M}$, which can be considered to be a list of integers $\mathbb{Z}_{\ell m} = \{0, 1, \ldots, \ell m - 1\}$ that represent locations on the 2D grid.

*Lemma A1 (*Lemma 4 in Ref. [18]*)*. A code $QC(A, B)$ has a toric layout on a $2\ell \times 2m$ grid if there exist $i, j, g, h \in \{1, 2, 3\}$ such that

(1) $\langle A_i A_j^T, B_g B_h^T \rangle = \mathcal{M}$
(2) $\mathrm{ord}(A_i A_j^T)\, \mathrm{ord}(B_g B_h^T) = \ell m$

Here, $\langle A_i A_j^T, B_g B_h^T \rangle$ indicates the group generated by $A_i A_j^T$ and $B_g B_h^T$. The matrices $A_i A_j^T$ and $B_g B_h^T$ then correspond to horizontal and vertical translations, respectively, on the grid. To have a toric layout, these translations must visit the $\ell m$ $X$- and $Z$-type checks, as well as the two sets of $\ell m$ data qubits. In practice, this can be checked by multiplying $(B_g B_h^T)^b (A_i A_j^T)^a$ for $0 \le b < \mathrm{ord}(B_g B_h^T)$, $0 \le a < \mathrm{ord}(A_i A_j^T)$ with a basis vector of $\mathbb{F}_2^{\ell m}$ and seeing whether the other $\ell m - 1$ basis vectors can be obtained. Satisfying this is equivalent to satisfying condition (1). For a given choice of $A = A_1 + A_2 + A_3$ and $B = B_1 + B_2 + B_3$, there might not be assignments of $i, j, g, h$ such that Lemma A1 is satisfied. There may also be several satisfying assignments. Each satisfying assignment yields an embedding with a defined generator shape, which in turn determines the fraction of generators that cross the long boundary condition. Thus, the embedding controls the routing schedule and number of masked generators, both of which affect the overall error-correction performance of the code.

---

[1] S. B. Bravyi and A. Y. Kitaev, Quantum codes on a lattice with boundary, arXiv:quant-ph/9811052.

[2] A. Kitaev, Fault-tolerant quantum computation by anyons, Ann. Phys. (NY) **303**, 2 (2003).

[3] A. G. Fowler, A. M. Stephens, and P. Groszkowski, High-threshold universal quantum computation on the surface code, Phys. Rev. A **80**, 052312 (2009).

[4] A. G. Fowler, A. C. Whiteside, and L. C. L. Hollenberg, Towards practical classical processing for the surface code, Phys. Rev. Lett. **108**, 180501 (2012).

[5] Y. Zhao, Y. Ye, H.-L. Huang, Y. Zhang, D. Wu, H. Guan, Q. Zhu, Z. Wei, T. He, S. Cao, F. Chen, *et al.*, Realization of an error-correcting surface code with superconducting qubits, Phys. Rev. Lett. **129**, 030501 (2022).

[6] R. Acharya, I. Aleiner, R. Allen, T. I. Andersen, M. Ansmann, F. Arute, K. Arya, A. Asfaw, J. Atalaya, R. Babbush, D. Bacon, *et al.*, Suppressing quantum errors by scaling a surface code logical qubit, Nature **614**, 676 (2023).

[7] D. Litinski, A game of surface codes: Large-scale quantum computing with lattice surgery, Quantum **3**, 128 (2019).

[8] M. E. Beverland, P. Murali, M. Troyer, K. M. Svore, T. Hoefler, V. Kliuchnikov, G. H. Low, M. Soeken, A. Sundaram, and A. Vaschillo, Assessing requirements to scale to practical quantum advantage, arXiv:2211.07629.

[9] N. P. Breuckmann and J. N. Eberhardt, Quantum low-density parity-check codes, PRX Quantum **2**, 040101 (2021).

[10] D. Gottesman, Fault-tolerant quantum computation with constant overhead, Quantum Inf. Comput. **14**, 1338 (2014).

[11] S. Bravyi and B. Terhal, A no-go theorem for a two-dimensional self-correcting quantum memory based on stabilizer codes, New J. Phys. **11**, 043029 (2009).

[12] S. Bravyi, D. Poulin, and B. Terhal, Tradeoffs for reliable quantum information storage in 2D systems, Phys. Rev. Lett. **104**, 050503 (2010).

[13] N. Baspin and A. Krishna, Quantifying nonlocality: How outperforming local quantum codes is expensive, Phys. Rev. Lett. **129**, 050505 (2022).

[14] N. Baspin and A. Krishna, Connectivity constrains quantum codes, Quantum **6**, 711 (2022).

[15] N. Delfosse, M. E. Beverland, and M. A. Tremblay, Bounds on stabilizer measurement circuits and obstructions to local implementations of quantum LDPC codes, arXiv:2109.14599.

[16] N. Baspin, O. Fawzi, and A. Shayeghi, A lower bound on the overhead of quantum error correction in low dimensions, arXiv:2302.04317.

[17] M. A. Tremblay, N. Delfosse, and M. E. Beverland, Constant-overhead quantum error correction with thin planar connectivity, Phys. Rev. Lett. **129**, 050504 (2022).

[18] S. Bravyi, A. W. Cross, J. M. Gambetta, D. Maslov, P. Rall, and T. J. Yoder, High-threshold and low-overhead fault-tolerant quantum memory, Nature **627**, 778 (2024).

[19] C. A. Pattison, A. Krishna, and J. Preskill, Hierarchical memories: Simulating quantum LDPC codes with local gates, arXiv:2303.04798.

[20] C. Gidney, M. Newman, P. Brooks, and C. Jones, Yoked surface codes, arXiv:2312.04522.

[21] D. Ruiz, J. Guillaud, A. Leverrier, M. Mirrahimi, and C. Vuillot, LDPC-cat codes for low-overhead quantum computing in 2D, arXiv:2401.09541.

[22] C. Ryan-Anderson, N. C. Brown, M. S. Allman, B. Arkin, G. Asa-Attuah, C. Baldwin, J. Berg, J. G. Bohnet, S. Braxton, N. Burdick, *et al.*, Implementing fault-tolerant entangling gates on the five-qubit code and the color code, arXiv:2208.01863.

[23] Q. Xu, J. P. B. Ataides, C. A. Pattison, N. Raveendran, D. Bluvstein, J. Wurtz, B. Vasic, M. D. Lukin, L. Jiang, and H. Zhou, Constant-overhead fault-tolerant quantum computation with reconfigurable atom arrays, Nat. Phys. **20**, 1084 (2024).

[24] J. Viszlai, W. Yang, S. F. Lin, J. Liu, N. Nottingham, J. M. Baker, and F. T. Chong, Matching generalized-bicycle codes to neutral atoms for low-overhead fault-tolerance, arXiv:2311.16980.

[25] D. Bluvstein, S. J. Evered, A. A. Geim, S. H. Li, H. Zhou, T. Manovitz, S. Ebadi, M. Cain, M. Kalinowski, D. Hangleiter, *et al.*, Logical quantum processor based on reconfigurable atom arrays, Nature **626**, 58 (2024).

[26] S. A. Moses, C. H. Baldwin, M. S. Allman, R. Ancona, L. Ascarrunz, C. Barnes, J. Bartolotta, B. Bjork, P. Blanchard, M. Bohn, *et al.*, A race-track trapped-ion quantum processor, Phys. Rev. X **13**, 041052 (2023).

[27] G. Burkard, T. D. Ladd, A. Pan, J. M. Nichol, and J. R. Petta, Semiconductor spin qubits, Rev. Mod. Phys. **95**, 025003 (2023).

[28] M. D. Smet, Y. Matsumoto, A.-M. J. Zwerver, L. Tryputen, S. L. de Snoo, S. V. Amitonov, A. Sammak, N. Samkharadze, Önder Gül, R. N. M. Wasserman, *et al.*, High-fidelity single-spin shuttling in silicon, arXiv:2406.07267.

[29] N. Berthusen and D. Gottesman, Partial syndrome measurement for hypergraph product codes, Quantum **8**, 1345 (2024).

[30] A. Leverrier, J.-P. Tillich, and G. Zémor, in *2015 IEEE 56th Annual Symposium on Foundations of Computer Science* (Berkeley, CA, USA, 2015), p. 810.

[31] A. A. Kovalev and L. P. Pryadko, Quantum Kronecker sum-product low-density parity-check codes with finite rate, Phys. Rev. A **88**, 012311 (2013).

[32] P. W. Shor, Scheme for reducing decoherence in quantum computer memory, Phys. Rev. A **52**, R2493 (1995).

[33] D. Gottesman, Stabilizer codes and quantum error correction, arXiv:quant-ph/9705052.

[34] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane, Quantum error correction and orthogonal geometry, Phys. Rev. Lett. **78**, 405 (1997).

[35] A. R. Calderbank and P. W. Shor, Good quantum error-correcting codes exist, Phys. Rev. A **54**, 1098 (1996).

[36] A. Steane, Multiple particle interference and quantum error correction, Proc. R. Soc. Lond. Ser. A **452**, 2551 (1996).

[37] J.-P. Tillich and G. Zemor, Quantum LDPC codes with positive rate and minimum distance proportional to the square root of the blocklength, IEEE Trans. Inf. Theory **60**, 1193 (2014).

[38] M. B. Hastings, J. Haah, and R. O'Donnell, in *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC)* (Association for Computing Machinery, New York, NY, USA, 2021), p. 1276.

[39] N. P. Breuckmann and J. N. Eberhardt, Balanced product quantum codes, IEEE Trans. Inf. Theory **67**, 6653 (2021).

[40] P. Panteleev and G. Kalachev, Quantum LDPC codes with almost linear minimum distance, IEEE Trans. Inf. Theory **68**, 213 (2022).

[41] P. Panteleev and G. Kalachev, in *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing (STOC)* (Association for Computing Machinery, New York, NY, USA, 2022), p. 375.

[42] A. Leverrier and G. Zemor, in *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)* (Denver, CO, USA, 2022), p. 872.

[43] I. Dinur, M.-H. Hsieh, T.-C. Lin, and T. Vidick, in *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC)* (Association for Computing Machinery, New York, NY, USA, 2023), p. 905.

[44] N. Alon, F. R. K. Chung, and R. L. Graham, Routing permutations on graphs via matchings, SIAM J. Discrete Math. **7**, 513 (1994).

[45] L. Zhang, Optimal bounds for matching routing on trees, SIAM J. Discrete Math. **12**, 64 (1999).

[46] F. Chung, *Spectral Graph Theory* (American Mathematical Society, 1996).

[47] A. Cowtan, S. Dilkes, R. Duncan, A. Krajenbrink, W. Simmons, and S. Sivarajah, in *14th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2019)*, Leibniz International Proceedings in Informatics (LIPIcs) LIPIcs (Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019), Vol. 135, p. 5:1.

[48] A. M. Childs, E. Schoute, and C. M. Unsal, in *14th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2019)*, Leibniz International Proceedings in Informatics (LIPIcs) LIPIcs (Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019), Vol. 135, p. 3:1.

[49] D. Devulapalli, E. Schoute, A. Bapat, A. M. Childs, and A. V. Gorshkov, Quantum routing with teleportation, Phys. Rev. Res. **6**, 033313 (2024).

[50] D. J. Rosenbaum, in *8th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2013)*, Leibniz International Proceedings in Informatics (LIPIcs) (Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2013), Vol. 22, p. 294.

[51] S. Hillmich, A. Zulehner, and R. Wille, in *Proceedings of the 26th Asia and South Pacific Design Automation Conference, ASPDAC '21* (Association for Computing Machinery, New York, NY, USA, 2021), p. 792.

[52] M. Beverland, V. Kliuchnikov, and E. Schoute, Surface code compilation via edge-disjoint paths, PRX Quantum **3**, 020342 (2022).

[53] D. Loss and D. P. DiVincenzo, Quantum computation with quantum dots, Phys. Rev. A **57**, 120 (1998).

[54] M. V. G. Dutt, L. Childress, L. Jiang, E. Togan, J. Maze, F. Jelezko, A. S. Zibrov, P. R. Hemmer, and M. D. Lukin, Quantum register based on individual electronic and nuclear spin qubits in diamond, Science **316**, 1312 (2007).

[55] M. Żukowski, A. Zeilinger, M. A. Horne, and A. K. Ekert, "Event-ready-detectors" Bell experiment via entanglement swapping, Phys. Rev. Lett. **71**, 4287 (1993).

[56] D. Gottesman and I. L. Chuang, Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations, Nature **402**, 390 (1999).

[57] D. Gottesman, Opportunities and challenges in fault-tolerant quantum computation, arXiv:2210.15844.

[58] A. Shafaei, M. Saeedi, and M. Pedram, in *2014 19th Asia and South Pacific Design Automation Conference (ASP–DAC)* (Singapore, 2014), p. 495.

[59] M. Y. Siraichi, V. F. d. Santos, C. Collange, and F. M. Q. Pereira, in *Proceedings of the 2018 International Symposium on Code Generation and Optimization (CGO)* (Association for Computing Machinery, New York, NY, USA, 2018), p. 113.

[60] G. Li, Y. Ding, and Y. Xie, in *Proceedings of the Twenty–Fourth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)* (Association for Computing Machinery, New York, NY, USA, 2019), p. 1001.

[61] A. M. Childs, E. Schoute, and C. M. Unsal, in *14th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2019)*, Leibniz International Proceedings in Informatics (LIPIcs), (Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2019), Vol. 135, p. 3:1.

[62] S. Herbert, On the depth overhead incurred when running quantum algorithms on near-term quantum computers with limited qubit connectivity, arXiv:1805.12570.

[63] E. Bäumer, V. Tripathi, D. S. Wang, P. Rall, E. H. Chen, S. Majumder, A. Seif, and Z. K. Minev, Efficient long-range entanglement using dynamic circuits, arXiv:2308.13065.

[64] Y. Tomita and K. M. Svore, Low-distance surface codes under realistic quantum noise, Phys. Rev. A **90**, 062320 (2014).

[65] L. P. Pryadko, V. A. Shabashov, and V. K. Kozin, QDistRnd: A GAP package for computing the distance of quantum error-correcting codes, J. Open Source Softw. **7**, 4120 (2022).

[66] Y. Zeng, P. Xu, X. He, Y. Liu, M. Liu, J. Wang, D. J. Papoular, G. V. Shlyapnikov, and M. Zhan, Entangling two individual atoms of different isotopes via Rydberg blockade, Phys. Rev. Lett. **119**, 160502 (2017).

[67] K. Singh, S. Anand, A. Pocklington, J. T. Kemp, and H. Bernien, Dual-element, two-dimensional atom array with continuous-mode operation, Phys. Rev. X **12**, 011040 (2022).

[68] S. Anand, C. E. Bradley, R. White, V. Ramesh, K. Singh, and H. Bernien, A dual-species Rydberg array, Nat. Phys. **20**, 1744 (2024).

[69] A. V. Gorshkov, A. M. Rey, A. J. Daley, M. M. Boyd, J. Ye, P. Zoller, and M. D. Lukin, Alkaline-earth-metal atoms as few-qubit quantum registers, Phys. Rev. Lett. **102**, 110503 (2009).

[70] J. W. Lis, A. Senoo, W. F. McGrew, F. Rönchen, A. Jenkins, and A. M. Kaufman, Midcircuit operations using the omg architecture in neutral atom arrays, Phys. Rev. X **13**, 041035 (2023).

[71] P. Scholl, A. L. Shaw, R. Finkelstein, R. B.-S. Tsai, J. Choi, and M. Endres, Erasure-cooling, control, and hyperentanglement of motion in optical tweezers, arXiv:2311.15580.

[72] C. H. Bennett, G. Brassard, S. Popescu, B. Schumacher, J. A. Smolin, and W. K. Wootters, Purification of noisy entanglement and faithful teleportation via noisy channels, Phys. Rev. Lett. **76**, 722 (1996).

[73] D. Deutsch, A. Ekert, R. Jozsa, C. Macchiavello, S. Popescu, and A. Sanpera, Quantum privacy amplification and the security of quantum cryptography over noisy channels, Phys. Rev. Lett. **77**, 2818 (1996).

[74] D. S. Wang, A. G. Fowler, and L. C. L. Hollenberg, Surface code quantum computing with error rates over 1%, Phys. Rev. A **83**, 020302(R) (2011).

[75] C. Gidney, STIM: A fast stabilizer circuit simulator, Quantum **5**, 497 (2021).

[76] M. McEwen, D. Bacon, and C. Gidney, Relaxing hardware requirements for surface code circuits using time-dynamics, Quantum **7**, 1172 (2023).

[77] C. Baldwin, Quantinuum hardware specifications, https://github.com/CQCL/quantinuum-hardware-specifications, 2022.

[78] IonQ Aria: Practical performance, https://ionq.com/resources/ionq-aria-practical-performance, 2024.

[79] IBM Quantum, https://quantum.ibm.com/, 2021.

[80] P. Panteleev and G. Kalachev, Degenerate quantum LDPC codes with good finite length performance, Quantum **5**, 585 (2021).

[81] J. Roffe, D. R. White, S. Burton, and E. Campbell, Decoding across the quantum low-density parity-check code landscape, Phys. Rev. Res. **2**, 043423 (2020).

[82] J. Roffe, LDPC: PYTHON tools for low density parity check codes, https://pypi.org/project/ldpc/, 2022.

[83] T. R. Scruby, T. Hillmann, and J. Roffe, High-threshold, low-overhead and single-shot decodable fault-tolerant quantum memory, arXiv:2406.14445.

[84] O. Higgott and C. Gidney, Sparse blossom: Correcting a million errors per core second with minimum-weight matching, arXiv:2303.15933.

[85] L. Jiang, J. M. Taylor, N. Khaneja, and M. D. Lukin, Optimal approach to quantum communication using dynamic programming, Proc. Natl. Acad. Sci. **104**, 17291 (2007).

[86] W. Dür and H. J. Briegel, Entanglement purification and quantum error correction, Rep. Progr. Phys. **70**, 1381 (2007).

[87] S. Krastanov, V. V. Albert, and L. Jiang, Optimized entanglement purification, Quantum **3**, 123 (2019).

[88] C. Gidney, Tetrationally compact entanglement purification, arXiv:2311.10971.

[89] M. Saffman, T. G. Walker, and K. Mølmer, Quantum information with Rydberg atoms, Rev. Mod. Phys. **82**, 2313 (2010).

[90] J. T. Young, P. Bienias, R. Belyansky, A. M. Kaufman, and A. V. Gorshkov, Asymmetric blockade and multiqubit gates via dipole-dipole interactions, Phys. Rev. Lett. **127**, 120501 (2021).

[91] 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 quantum processors, npj Quantum Inf. **5**, 18 (2019).

[92] https://github.com/noahberthusen/qecc_routing