Constructions and Performance of Hyperbolic and Semi-Hyperbolic Floquet Codes

Oscar Higgott[*]
Department of Physics & Astronomy, University College London, London WC1E 6BT, United Kingdom

Nikolas P. Breuckmann[†]
School of Mathematics, University of Bristol, Fry Building, Woodland Road, Bristol BS8 1UG, United Kingdom

(Received 5 December 2023; revised 17 July 2024; accepted 18 October 2024; published 22 November 2024)

We construct families of Floquet codes derived from color-code tilings of closed hyperbolic surfaces. These codes have weight-two check operators, a finite encoding rate and can be decoded efficiently with minimum-weight perfect matching. We also construct semi-hyperbolic Floquet codes, which have improved distance scaling, and are obtained via a fine-graining procedure. Using a circuit-based noise model that assumes direct two-qubit measurements, we show that semi-hyperbolic Floquet codes can be 48 times more efficient than planar honeycomb codes and therefore over 100 times more efficient than alternative compilations of the surface code to two-qubit measurements, even at physical error rates of 0.3% to 1%. We further demonstrate that semi-hyperbolic Floquet codes can have a teraquop footprint of only 32 physical qubits per logical qubit at a noise strength of 0.1%. For standard circuit-level depolarizing noise at $p = 0.1\%$, we find a 30 times improvement over planar honeycomb codes and a 5.6 times improvement over surface codes. Finally, we analyze small instances that are amenable to near-term experiments, including a Floquet code derived from the Bolza surface that encodes four logical qubits into 16 physical qubits.

DOI: 10.1103/PRXQuantum.5.040327

I. INTRODUCTION

Recent breakthroughs in the development of quantum LDPC codes have led to significant improvements in the asymptotic parameters of quantum codes [1–6]. However, demonstrating a significant advantage over the surface code for realistic noise models and reasonable system sizes can be challenging. For example, Ref. [7] demonstrated a significant 14 times saving in resources relative to the surface code using hypergraph product codes for circuit-level noise, however this improvement was shown for very low physical error rates of around 0.01% and for very large system sizes (around 13 million qubits). This is in large part due to the high-weight stabilizer generators common in many families of quantum LDPC codes, resulting in deep and complex syndrome extraction circuits, which can introduce hook errors and lower their noise thresholds [7–10].

Since we ultimately need to construct a fault-tolerant circuit implementing a code, it is therefore highly desirable to use a code that supports low-weight check operators. A broad family of codes that can have this property are subsystem codes [11]. Subsystem codes are a generalization of stabilizer codes that can allow high-weight stabilizers to be measured by taking the product of measurement outcomes of low-weight anticommuting check operators (gauge operators). Several families of subsystem codes with low-weight check operators have been discovered [12–15], and in Ref. [16] we constructed finite-rate subsystem codes with weight-three check operators, derived from tessellations of closed hyperbolic surfaces. The low-weight checks and symmetries of the subsystem hyperbolic codes in Ref. [16] led to efficient syndrome extraction circuits, which are 4.3 times more efficient than surface-code circuits for 0.1% circuit-level noise and for a system size of around 19 000 physical qubits. However, it is natural to ask if we can reduce the check weight further and demonstrate larger resource savings relative to the surface code.

[*] Contact author: oscar.higgott.18@ucl.ac.uk
[†] Contact author: niko.breuckmann@bristol.ac.uk

We achieve further improvements in this work by constructing codes from the broader family of Floquet codes [17], which can be seen as a generalization of subsystem codes, see e.g., Ref. [18] for a discussion. Similar to subsystem codes, Floquet codes involve the measurement of low-weight anticommuting check operators. Unlike subsystem codes, however, Floquet codes do not require that the logical operators of the code remain static over time. Indeed, Floquet codes do not admit a static set of generators for all their logical Pauli operators, and the form of these logical operators instead evolves periodically during the syndrome extraction circuit. This additional flexibility leads to codes with weight-two check operators and good performance [19], especially in platforms that support direct two-qubit Pauli measurements, such as Majorana-based qubits [20]. Notably, the planar honeycomb code [20–22] is a specific Floquet code derived from a hexagonal lattice with open boundary conditions, and is therefore amenable to experimental realization on a quantum computer chip (e.g., a solid-state device such as a superconducting qubit architecture).

In Ref. [23], Vuillot showed how Floquet codes can be derived from any tiling suitable to define a color code [24], and furthermore observed that Floquet codes derived from color-code tilings of closed hyperbolic surfaces, which we refer to as *hyperbolic Floquet codes*, would have a constant encoding rate and logarithmic distance. However, there has not been prior work constructing explicit examples of hyperbolic Floquet codes or analyzing their performance.

In this work, we construct explicit families of hyperbolic Floquet codes (see Fig. 1) and analyze their performance numerically, comparing them to planar honeycomb codes and surface codes. We also construct Floquet codes derived from semi-hyperbolic lattices, which fine-grain hyperbolic lattices, leading to an improved distance scaling that enables exponential suppression of errors while retaining an advantage over honeycomb and surface codes. Furthermore, the thresholds of families of semi-hyperbolic Floquet codes are essentially the same as the threshold of the honeycomb code. All of our constructions have weight-two check operators and can be decoded efficiently using surface-code decoders, such as minimum-weight perfect matching [25–28].

For platforms that support direct two-qubit measurements, we find that semi-hyperbolic Floquet codes can require 48 times fewer physical qubits than honeycomb codes even for high physical error rates of 0.3% to 1% (for the "EM3" noise model) and for a system size of 21 504 physical qubits. These results imply that our constructions are over 100 times more efficient than alternative compilations of the surface code to two-qubit measurements [29,30], which have been shown to be less efficient than honeycomb codes for the same noise model [20,30]. We also show that semi-hyperbolic Floquet codes can require as few as 32 physical qubits per logical qubit to achieve logical failure rates below 10^{-12} per logical qubit at a physical error rate of 0.1%, whereas honeycomb codes instead require 600 [19] to 2000 [20] physical qubits per logical qubit in the same regime.

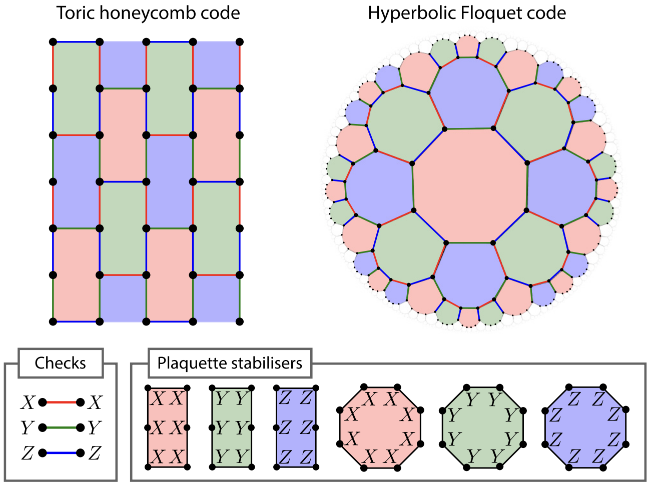

FIG. 1. Two of the families of Floquet codes studied in this work. A data qubit is associated with each vertex of a lattice, a two-qubit check operator is associated with each edge and a plaquette stabilizer generator is associated with each face. Left: a toric honeycomb code, which is defined on a hexagonal lattice with periodic boundary conditions and three-colorable faces [17]. Here we consider a patch with four columns of data qubits and six rows (dimensions 4 × 6). Opposite sides of the lattice are identified. Right: a Floquet code derived from a tiling of a hyperbolic surface with three octogans meeting at each vertex (vertex configuration 8.8.8) and three-colorable faces. Here we draw a region of the tiled hyperbolic plane defining the code family, however each Floquet code in the family is derived from a tiling of a closed surface.

We also consider a standard circuit-level depolarizing noise model ("SD6"), for which two-qubit measurements are implemented using an ancilla, CNOT gates and single-qubit rotations. For this noise model, our semi-hyperbolic Floquet codes are 30 times more efficient than planar honeycomb codes and over 5.6 times more efficient than conventional surface codes for physical error rates of around 0.1% and below.

Finally, we construct small examples of hyperbolic Floquet codes that are amenable to near-term experiments. This includes codes derived from the Bolza surface, using as few as 16 physical qubits, and which we show are three to six times more efficient than their Euclidean counterparts.

Although these (semi-)hyperbolic Floquet codes cannot be implemented using geometrically local connections in a planar Euclidean architecture, we show how they can instead be implemented using a *biplanar* or *modular* architecture. A biplanar architecture uses two layers of qubits, where connections within each layer do not cross, but may be long range. A modular architecture consists of many small modules, where each module requires only local two-dimensional (2D) planar Euclidean connectivity and connections between modules may be long range. The long-range connections between modules could be mediated via photonic links in a trapped-ion architecture [31], for example.

We note that some related works on LDPC codes were posted to arXiv shortly after the first version of our paper. References [32,33] demonstrate an approximately 10 times reduction in qubit overhead relative to the surface code for circuit-level noise of around 0.1% using bivariate-bicycle (BB) and lifted-product (LP) codes, respectively, and Ref. [34] also studied hyperbolic Floquet codes.

Our work complements Refs. [32,33], since our hyperbolic Floquet codes excel for the EM3 noise model (pair-measurement gate set) that is likely challenging for the BB and LP codes. Specifically, we achieve an approximately 10–20 times qubit reduction relative to surface codes for the EM3 noise model using a similar system size to Refs. [32,33] (a few hundred qubits), with the improvement rising to an approximately 100 times reduction in qubit overhead for larger system sizes. Given the check weight of the BB and LP codes it seems unlikely that a compilation to pair measurements would perform well, although this has not been explored. On the other hand, the codes in Refs. [32,33] perform better for a standard circuit-level noise model, for which we obtain an approximately 2.6 times qubit saving relative to the surface code using similar system sizes (hundreds of qubits) as those used for their approximately 10 times saving. Our improvement rises to a 5.6 times saving for larger system sizes. Furthermore, unlike Refs. [32,33], our codes are amenable to fast decoders based on matching and our connectivity graph has a lower degree of qubit connectivity (degree three, compared to 6 or more in Refs. [32,33]).

Our work also complements Ref. [34], which also studies hyperbolic Floquet codes. Both our work and theirs considers 6.6.6 tilings, however we find some instances not considered in Ref. [34], and likewise they construct some 6.6.6 tilings not considered in our work. Additionally, our work also considers 4.10.10 and 4.8.10 tilings as well as semi-hyperbolic tilings, which are necessary to suppress errors exponentially and obtain reasonable teraquop footprint estimates.

Our paper is organized as follows. We start by introducing Euclidean, hyperbolic, and semi-hyperbolic color-code tilings in Sec. II. In Sec. III we review Floquet codes, including how we construct Floquet-code circuits from the color-code tilings of Sec. II, as well as how these circuits can be decoded. Section III also describes the EM3 and SD6 noise models we use in our simulations. In Sec. IV we present an analysis of the (semi-)hyperbolic Floquet codes we have constructed, including a study of their parameters (Sec. IV A) and simulations comparing them to honeycomb and surface codes (Sec. IV B). We then conclude in Sec. V with a summary of our results and a discussion of future work.

II. COLOR-CODE TILINGS

Floquet codes [17] can be defined on any tiling of a surface that is also suitable to define a 2D color code [24]; namely, it is sufficient that the faces are three colorable and the vertices are three valent [23]. We will refer to such a tiling as a *color-code tiling*. In this section, we will review color-code tilings and describe the tilings used to construct Floquet codes in this work.

We denote by $\mathcal{T} = (V, E, F)$ a color-code tiling with vertices V, edges $E \subset V^2$, and faces $F \subset 2^V$. Each face $f \in F$ is assigned a color $C(f) \in \{R, G, B\}$, which can be red, green, or blue (R, G, or B), such that two faces that share an edge have different colors:

$$f_1 \cap f_2 \in E \implies C(f_1) \neq C(f_2), \quad \forall (f_1, f_2) \in F^2. \qquad (1)$$

We will also assign a color $C(e) \in \{R, G, B\}$ to each edge e, given by the color of the faces that it links. Equivalently, the color of an edge is the complement of the colors of the two faces it borders. See Fig. 1 for two examples of color-code tilings.

We now introduce some definitions from homology, which are applicable to any tiling $\mathcal{T} = (V, E, F)$ of a closed surface, including tilings that are not color-code tilings. The *dual* of $\mathcal{T}$, which we denote $\mathcal{T}^* = (V^*, E^*, F^*)$, has a vertex for each face of $\mathcal{T}$ and two vertices in $\mathcal{T}^*$ are connected by an edge if the corresponding faces in $\mathcal{T}$ share an edge. A *cycle* in $\mathcal{T}$ is a set of edges (a subset of E) that forms a collection of closed paths in $\mathcal{T}$ (it has no boundary). A *cocycle* in $\mathcal{T}$ is a set of edges (also a subset of E) that corresponds to a cycle in $\mathcal{T}^*$. A cycle is *homologically nontrivial* if it is not the boundary of a set of faces in $\mathcal{T}$ and is homologically trivial otherwise. Likewise, a cocycle is homologically nontrivial if it does not correspond to the boundary of a set of faces in $\mathcal{T}^*$ and is homologically trivial otherwise. Two cycles belong to the same homology class if the symmetric difference of the two cycles is a homologically trivial cycle and the dimension of the first homology group of the tiling, $\dim H_1$, is the number of distinct homology classes.

A. Hyperbolic surfaces with color-code tilings

Closed hyperbolic surfaces can be obtained via a compactification procedure. Consider the infinite hyperbolic plane $\mathbb{H}^2$, which is the universal covering space of hyperbolic surfaces, and its group of isometries $\mathrm{PSL}_2(\mathbb{R}) \rtimes \mathbb{Z}_2$. Taking a subgroup $\Gamma \subset \mathrm{PSL}_2(\mathbb{R}) \rtimes \mathbb{Z}_2$, which acts as fixed-point-free and discontinuous on $\mathbb{H}^2$, we obtain a closed hyperbolic surface by taking the quotient $X =$

$\mathbb{H}^2/\Gamma$. Note that due to the Gauß-Bonnet theorem, the area of the surface $\text{area}(X)$ is proportional to the dimension of the first homology group $\dim H_1(X)$.

This idea extends to tiled surfaces: consider a periodic tiling on $\mathbb{H}^2$ and a group Γ, which preserves the tiling structure, then the quotient surface $X = \mathbb{H}^2/\Gamma$ supports the same tiling. In particular, if we consider a periodic color-code tiling of $\mathbb{H}^2$ and a group $\Gamma \subset \text{PSL}_2(\mathbb{R}) \rtimes \mathbb{Z}_2$ that is compatible with it, we obtain a closed hyperbolic surface with the desired tiling.

B. Wythoff construction

We can obtain suitable color-code tilings of the hyperbolic plane using Wythoff's kaleidoscopic construction. This approach has been used in Ref. [35] to define hyperbolic color codes and pin codes. We now review this method and explain how we use it to generate hyperbolic color-code tilings.

We choose a *fundamental triangle* with internal angles π/p, π/q and π/t. We can generate an infinite tiling through reflections in the sides of the triangle (we refer to each side of the triangle as a mirror). The group generated by these reflections (a symmetry group of the tiling) is the triangle group $\Delta(p,q,t)$, an example of a Coxeter group. We place a vertex, called a generator point, within the fundamental triangle, and from each mirror we draw a perpendicular line to the generator point. These lines are mapped by the reflections to edges in the tiling, and the generator point is similarly mapped to vertices of the tiling. For a given triangle group $\Delta(p,q,t)$ we can generate different tilings depending on the choice of generator point; in this work we place the generator point in the interior of the fundamental triangle to generate color-code tilings, as shown in Fig. 2 for $\Delta(2,4,5)$.

If we denote a reflection in each of the three sides by a, b, and c, where the angles between the pairs of mirrors (a,b), (b,c), and (c,a) are π/t, π/q and π/p, respectively, then the triangle group $\Delta(p,q,t)$ has presentation

$$\Delta(p,q,t) = \langle a,b,c \,|\, a^2 = b^2 = c^2 = (ab)^t = (bc)^q = (ca)^p = e\rangle, \tag{2}$$

where e is the identity element. The relations for $(ab)^t$, $(bc)^q$, and $(ca)^p$ can be understood from the fact that ab, bc, and ca are rotations by $2\pi/t$, $2\pi/q$, and $2\pi/p$, respectively. We note that if $1/p + 1/q + 1/t < 1$ then we obtain a color-code tiling of the hyperbolic plane.

As discussed in Sec. II A, we can use a compactification procedure to construct a finite tiling of a closed hyperbolic surface from an infinite tiling [36,37]. Concretely, for a triangle group $\Delta(p,q,t)$ we can find a normal subgroup $\Gamma \subset \Delta(p,q,t)$ that has no fixed points and gives a finite quotient group $\Delta(p,q,t)/\Gamma$, see Refs. [36,37]. The quotient group $G^+_{p,q,t} := \Delta(p,q,t)/\Gamma$ can then be used to define uniform tilings using the Wythoff construction, and has presentation

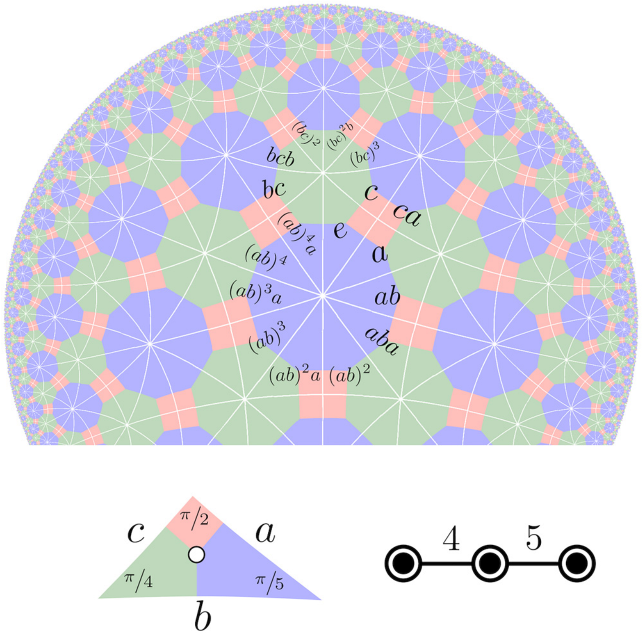

FIG. 2. A hyperbolic color-code tiling generated by the hyperbolic triangle group $\Delta(2,4,5)$ using Wythoff's kaleidoscopic construction. The tiling is generated through reflections across the sides of a fundamental triangle with internal angles $\pi/2$, $\pi/4$ and $\pi/5$. We show a white cut along each reflection line. Below the tiling we show the fundamental triangle and Coxeter diagram. A generator point is placed in the interior of the fundamental triangle and from each mirror a perpendicular line is drawn to the generator point. This generates a uniform tiling with vertex configuration 4.8.10 and three-colorable faces.

$$G^+_{p,q,t} = \langle a,b,c \,|\, a^2 = b^2 = c^2 = (ab)^t = (bc)^q = (ca)^p = r_1 = \cdots = r_v = e\rangle, \tag{3}$$

where here the additional relations $r_1,\ldots,r_v$ are the generators of Γ.

When $\Gamma \subset \Delta(p,q,t)$ contains only elements consisting of an even number of reflections, the resulting surface is orientable and the graph of the tiling is bipartite. This is because every vertex is labeled by an element of the reflection group $h \in \Delta(p,q,t)/\Gamma$ and, assuming that Γ contains only even-parity elements, the assignment of a parity to each h is well defined. An example of such a finite tiling (of the Bolza surface) is given in Fig. 3, where the parity of reflections of vertices are given by the color black and white.

C. Properties of uniform tilings

Color-code tilings obtained using the Wythoff construction as described in Sec. II B are uniform tilings, which can be described by their *vertex configuration*, a sequence

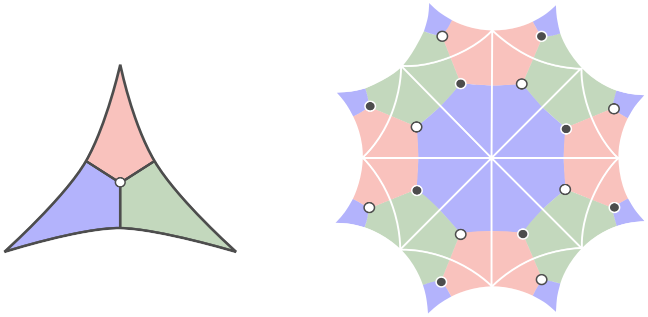

FIG. 3. Construction of color-code tilings via the Wythoff construction. Left: fundamental triangle with internal angles $\pi/4$, $\pi/4$, and $\pi/4$ belonging to the hyperbolic triangle group $\Delta(4,4,4)$. Right: the fundamental domain of the group Γ_B defining the Bolza surface $\mathbb{H}^2/\Gamma_B$. In the figure opposite sides are identified, so that $\dim H_1(\mathbb{H}^2/\Gamma_B) = 4$. The Bolza surface supports the three-colorable 8.8.8 tiling. Neighboring fundamental triangles of the tiling are related by a reflection along the triangle's side. The vertices in the middle of each fundamental triangle swap between black and white with each reflection.

of numbers giving the number of sides of the faces around a vertex. For example, each vertex of a degree-three uniform tiling with vertex configuration $r.g.b$ has three faces around it, with r, g, and b sides. When the tiling is constructed using the triangle group $\Delta(p,q,t)$ with a generator point in the middle of the fundamental triangle, we have that $r = 2p$, $g = 2q$, and $b = 2t$. Uniform tilings of hyperbolic surfaces satisfy $1/r + 1/g + 1/b < 1/2$, whereas a Euclidean tiling (e.g., 6.6.6 or 4.8.8) satisfies $1/r + 1/g + 1/b = 1/2$. Since we consider only color-code tilings, we can equivalently define r, g, and b to be the number of sides that red, green, and blue faces have in the uniform tiling, respectively. We construct Floquet codes from 6.6.6 (honeycomb) tilings, as well as 8.8.8, 4.8.10, and 4.10.10 hyperbolic tilings in this work; see Fig. 1 for an illustration of 6.6.6 and 8.8.8 tilings as well as Fig. 2 (left) for a 4.8.10 tiling. Let us denote the set of faces with r, g and b sides in an $r.g.b$ uniform tiling as F_r, F_g, and F_b, respectively. Note that this partition of the faces $F = F_r \cup F_g \cup F_b$ is well defined by the face coloring even if r, g, and b are not unique. We also denote the edge set by E and the vertex set by V. The tiling has $|E| = 3|V|/2$ edges, and since $r|F_r| = g|F_g| = b|F_b| = |V|$, we have that

$$|F| = |V|\left(\frac{1}{r} + \frac{1}{g} + \frac{1}{b}\right). \tag{4}$$

If a tiling $\mathcal{T} = (V,E,F)$ has a single connected component and tiles a closed surface, it can be shown that the dimension of the first homology group of the cell complex associated with the tiling has dimension

$$\dim H_1 = 2 - \chi = 2 - |V| + |E| - |F| \tag{5}$$

where here $\chi := |V| - |E| + |F|$ is the Euler characteristic of the tiling. We refer the reader to Ref. [37] for further background on homology theory relevant to quantum codes derived from tilings. If $\mathcal{T}$ is further a uniform $r.g.b$ color-code tiling we have that

$$\begin{aligned}\dim H_1 &= 2 - |V| + |E| - |F| \\ &= |V|\left(\frac{1}{2} - \frac{1}{r} - \frac{1}{g} - \frac{1}{b}\right) + 2.\end{aligned} \tag{6}$$

Note that for an orientable surface with genus w, it also holds that $\dim H_1 = 2w$. The derived Floquet code will encode $k = \dim H_1$ logical qubits into $n = |V|$ physical qubits. Hence, Eq. (6) implies that this family of codes has a finite rate and that the proportionality depends on the plaquette stabilizer weights r, g, b.

D. Semi-hyperbolic color-code tilings

We also construct *semi-hyperbolic* color-code tilings, which interpolate between hyperbolic and Euclidean tilings, using a fine-graining procedure. Our semi-hyperbolic color-code tilings are inspired by the semi-hyperbolic tilings introduced in Ref. [38], which used a slightly different method of fine graining applied to 4.4.4.4.4 tilings (instead of color-code tilings) to improve the distance scaling of standard hyperbolic surface codes. A similar method was also used to construct subsystem semi-hyperbolic codes in Ref. [16]. As we will show in Sec. IV, these tilings can be used to obtain Floquet codes with improved distance scaling relative to those derived from purely hyperbolic tilings, while retaining an advantage over honeycomb codes.

A semi-hyperbolic color-code tiling $\mathcal{T}_l = (V_l, E_l, F_l)$ is defined from a *seed* color-code tiling $\mathcal{T} = (V,E,F)$ as well as a parameter l, which determines the amount of fine graining. To construct $\mathcal{T}_l$ we first take the dual of the seed tiling $\mathcal{T}$, which we recall we denote by $\mathcal{T}^* = (V^*, E^*, F^*)$. Since $\mathcal{T}$ is a color-code tiling and therefore three valent, all the faces of $\mathcal{T}^*$ are triangles, and we have that $|V^*| = |F|$, $|E^*| = |E|$, and $|F^*| = |V|$. If $\mathcal{T}$ is an 8.8.8 (also denoted 8^3) tiling then $\mathcal{T}^*$ is a 3^8 uniform tiling, i.e., with eight triangles meeting at each vertex. We construct a new tiling $\mathcal{T}_l^*$, which fine grains $\mathcal{T}^*$ by tiling each face of $\mathcal{T}^*$ with a triangular lattice, as shown in Fig. 4, such that each edge in $\mathcal{T}^*$ is subdivided into l edges in $\mathcal{T}_l^*$. Finally, we take the dual of $\mathcal{T}_l^*$ to obtain our semi-hyperbolic color-code tiling $\mathcal{T}_l$. The example of this procedure given in Fig. 4 starts from $\mathcal{T}^*$ (here a 3^8 tiling) using $l = 3$ to obtain a color-code tiling of hexagons and octagons.

Given an $r.g.b$ uniform color-code tiling $\mathcal{T}$, the semi-hyperbolic color-code tiling $\mathcal{T}_l$ derived from it using this procedure has a number of vertices, edges and faces

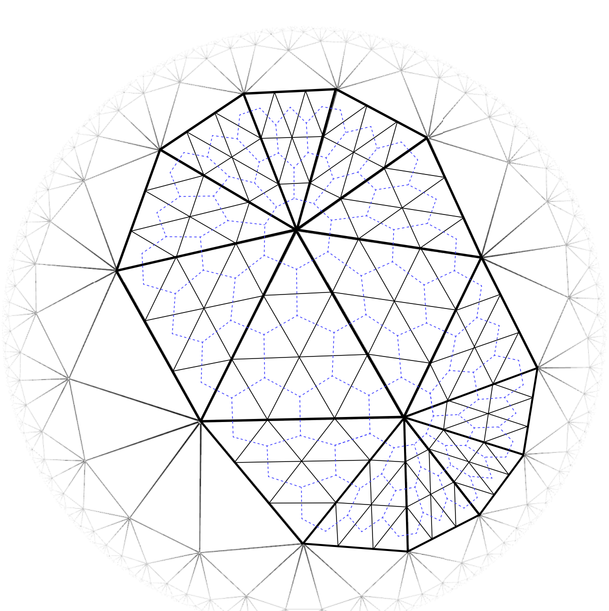

FIG. 4. Obtaining a semi-hyperbolic color-code tiling by fine graining a 3^8 tiling and then taking its dual. We fine grain by tiling each face of the 3^8 tiling with a triangular lattice, such that each side of each face in the 3^8 tiling becomes subdivided into l edges (shown for a subset of the faces here, with $l = 3$). We take the dual of this lattice (blue dashed lines) to obtain a semi-hyperbolic color-code tiling of hexagons and octagons.

given by

$$|V_l| = l^2|V|, \tag{7}$$

$$|E_l| = \frac{3l^2|V|}{2}, \tag{8}$$

$$|F_l| = \left(\frac{l^2}{2} - \frac{1}{2} + \frac{1}{r} + \frac{1}{g} + \frac{1}{b}\right)|V|. \tag{9}$$

Here Eq. (7) can be understood by noticing that the semi-hyperbolic construction replaces each face in the dual tiling $\mathcal{T}*$ (i.e., a vertex in $\mathcal{T}$) with a triangular tiling of l^2 faces in $\mathcal{T}_l^*$ (each a vertex in $\mathcal{T}_l$). Equation (8) follows from the fact that $\mathcal{T}_l$ is trivalent and hence $3|V_l| = 2|E_l|$. We derive Eq. (9) using the dual lattice, an example of which is shown in Fig. 4. In other words, we will find the number of vertices $|V_l^*|$ in $\mathcal{T}_l^*$ in terms of the number of faces $|F^*|$ in $\mathcal{T}^*$. We note that $\mathcal{T}_l^*$ still contains $|F^*|(1/r + 1/g + 1/b)$ vertices [see Eq. (4)] corresponding to the vertices from $\mathcal{T}^*$. Furthermore, we have added $l - 1$ additional vertices along each of the $|E^*| = 3|F^*|/2$ edges of $\mathcal{T}^*$, as well as $(l-1)(l-2)/2$ vertices fully contained within the interior of each of the original faces of $\mathcal{T}^*$. Combining these contributions and switching back to the primal tiling $\mathcal{T}_l$, we obtain Eq. (9).

Since we have not modified the topology of the surface, the dimension of the first homology group $\dim H_1 = 2 - |V_l| + |E_l| - |F_l|$ is independent of l and still given by Eq. (6).

III. FLOQUET CODES

Floquet codes, introduced by Hastings and Haah [17], are quantum error-correcting codes that are implemented by measuring two-qubit Pauli operators (they have weight-two checks). In Ref. [17] the authors introduced and studied the honeycomb code, which is a Floquet code defined from a hexagonal tiling of a torus, inspired by Kitaev's honeycomb model [39]. Shortly after, Vuillot showed that Floquet codes can be defined from any color-code tiling, and observed that the use of hyperbolic color-code tilings would lead to Floquet codes with a finite encoding rate and logarithmic distance [23].

We will focus our attention on Floquet codes derived from color-code tilings of *closed* surfaces. It is also possible to construct planar Floquet codes; we refer the reader to Refs. [20–22,40,41] for details. The definition of Floquet codes we use is consistent with Refs. [19,23], which is related to the original definition of Ref. [17] by a local Clifford unitary.

A. Checks and stabilizers

We use the color-code tiling $\mathcal{T}$ to define a Floquet code, which we will denote by $\mathcal{F}(\mathcal{T})$. Each vertex in $\mathcal{T}$ represents a qubit in $\mathcal{F}(\mathcal{T})$ and each edge in $\mathcal{T}$ represents a two-qubit check operator. There are three types of check operators: X checks (red edges), Y checks (green edges), and Z checks (blue edges). Specifically, the check operator for an edge $e = (v_1, v_2)$ is defined to be $P_e := P_{v_1}^{C(e)} P_{v_2}^{C(e)}$, where here P_q^c denotes a Pauli operator acting on qubit q labeled by the color $c \in \{R, G, B\}$, which determines the Pauli type. In other words, P_q^B, P_q^G and P_q^R denote Pauli operators X_q, Y_q, and Z_q, respectively. For the Floquet codes we consider in this work, the Pauli type of an edge is fixed and is determined by its color, and we will sometimes refer to checks of a given color as c checks, for $c \in \{R, G, B\}$. However, we note that alternative variants of Floquet codes have been introduced for which the Pauli type of an edge measurement is allowed to vary in the schedule (e.g., Floquet color codes [42]).

Each stabilizer generator of the Floquet code consists of a cycle of checks surrounding a face. In other words, we associate a stabilizer generator $P_f := \prod_{e \subset f} P_e$ with each face f. Stabilizer generators on red, green, and blue faces are X type, Y type, and Z type, respectively. We sometimes refer to these stabilizer generators as the *plaquette stabilizers* for clarity. Note that the plaquette stabilizers commute with all the checks. See Fig. 1 for examples of the checks and stabilizers for Floquet codes defined on hexagonal (6.6.6) and hyperbolic (8.8.8) lattices.

B. The schedule and instantaneous stabilizer group

The Floquet code is implemented by repeating a *round* of measurements, where each round consists of measuring all red checks, then all green checks and finally all blue checks. We refer to the measurement of all checks of a given color as a *subround* (so each round contains three subrounds).

Once the steady state is reached, after fault-tolerant initialization of the Floquet code, the state is always in the joint $+1$ eigenspace of the plaquette stabilizers defined in Sec. III A, regardless of which subround was just measured. This property allows measurements of the plaquette stabilizers to be used to define detectors for decoding (see Sec. III G). However, the full stabilizer group of the code at a given instant includes additional stabilizers corresponding to the check operators measured in the most-recent subround. We refer to the full stabilizer group of the code after a given subround as the *instantaneous stabilizer group* (ISG) [17]. In the steady state of the code, just after measuring a subround of checks with color $c \in R, G, B$, the ISG is generated by all the plaquette stabilizers, as well as the check operators of color c. The phase of each check operator in the ISG is given by its measurement outcome in the most-recent subround. We will sometimes refer to the ISG immediately after measuring c checks as ISG(c).

C. The embedded homological code

The state of a Floquet code can be mapped by a constant-depth unitary circuit to a 2D homological code, which we refer to as the embedded homological code. This mapping is shown in Ref. [17] for the honeycomb code and in Ref. [23] for general color-code tilings. The embedded homological code can be useful to understand and construct the logical operators of the code, as we will explain in Sec. III D. As an example, consider the state of the Floquet code immediately after a red subround, during which any red edge $e = (u, v)$ has participated in an X_uX_v measurement. This X_uX_v measurement projects the qubits u and v into a two-dimensional subspace, which we can consider an effective qubit in an embedded 2D homological code. The effective X and Z operators associated with this effective qubit are defined to be $\hat{X}_e := Y_uY_v$ (or Z_uZ_v) and $\hat{Z}_e := X_uI_v$ (or I_uX_v), respectively.

More concretely, we denote by $\mathcal{T}^*$ the dual of the color-code tiling $\mathcal{T}$, where we define a vertex in $\mathcal{T}^*$ for each face of $\mathcal{T}$, and two nodes in $\mathcal{T}^*$ are connected by an edge if and only if their corresponding faces in $\mathcal{T}$ share an edge. Furthermore, the color of a node in $\mathcal{T}^*$ is given by the color of the corresponding face in $\mathcal{T}$. We then define the restricted lattice $\mathcal{T}_c^*$ for $c \in \{R, G, B\}$ to be the subgraph of $\mathcal{T}^*$ where all nodes of color c (and their adjacent edges) have been removed [43]. We refer to $\mathcal{T}_R^*$ as the red restricted lattice (and similarly for green and blue). See Fig. 5 for examples of the red, green, and blue restricted lattices for an 8.8.8 hyperbolic tiling.

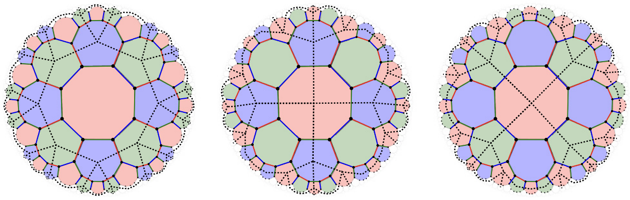

FIG. 5. The dashed lines in the left, middle, and right diagrams of the 8.8.8 tiling show the red ($\mathcal{T}_R^*$), green ($\mathcal{T}_G^*$), and blue ($\mathcal{T}_B^*$) restricted lattices, respectively. The restricted lattice $\mathcal{T}_c^*$ of color $c \in \{R, G, B\}$ defines the embedded homological code after the c checks are measured. Each edge, face, and vertex in the restricted lattice corresponds to an effective qubit, plaquette stabilizer or site stabilizer in the embedded homological code, respectively. All three restricted lattices are regular tessellations, with four octagons meeting at each vertex.

The embedded 2D homological code $\mathcal{C}(\mathcal{T}_c^*)$ associated with the Floquet code after measuring checks of color c is defined from $\mathcal{T}_c^*$ by associating qubits with the edges, Z checks with the faces and X checks with the vertices. An effective $\hat{X}$ operator on an effective qubit in $\mathcal{T}_c^*$ corresponds to a physical $P^c \otimes I$ operator on the Floquet code data qubits and similarly an effective $\hat{Z}$ operator corresponds to a physical $P^{\bar{c}} \otimes P^{\bar{c}}$ operator, where here $\bar{c} \in \{R, G, B\} \setminus \{c\}$. As an example of this mapping, notice that a red $X^{\otimes 8}$ plaquette stabilizer in Fig. 5 corresponds to a $\hat{Z}^{\otimes 8}$ plaquette operator in $\mathcal{C}(\mathcal{T}_R^*)$ and to an $\hat{X}^{\otimes 4}$ site operator in $\mathcal{C}(\mathcal{T}_G^*)$ or $\mathcal{C}(\mathcal{T}_B^*)$.

D. Logical operators

We can derive a set of logical operators of a Floquet code from the logical operators of the embedded 2D homological code [17,23], as shown in Fig. 6 for the toric honeycomb code. An unusual property of Floquet codes is that the logical operators generally do not commute with all of the checks. For example, none of the logical operators shown in Fig. 6 (left) commute with the blue checks. Fortunately we can still preserve the logical operators throughout the schedule by updating the logical operators after each subround, multiplying an element of the ISG into each of them, such that they commute with the next subround of check operator measurements. In this section we will give a suitable basis for the logical operators and show how they are updated in each subround to commute with the checks. This was explained in Ref. [19] for the toric honeycomb code (see their Fig. 1) whereas our description here is applicable to any Floquet code derived from a color-code tiling of a closed surface (Euclidean or hyperbolic).

We will define a basis for the logical operators of a Floquet code $\mathcal{F}(\mathcal{T})$ from the logical operators of $\mathcal{C}(\mathcal{T}_R^*)$, the

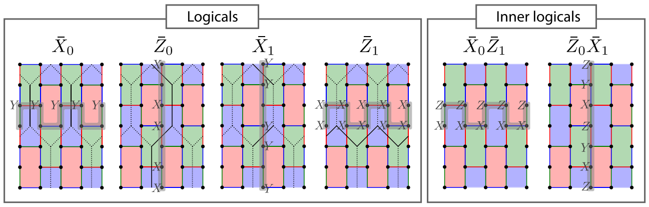

FIG. 6. Left: logical operators of the two logical qubits of the toric honeycomb code after measuring the red checks. These representatives of the logical operators commute with the round of green checks that follows. The overlaid black lattice (with dotted and solid lines) is the tiling of the corresponding embedded toric code, which has an effective qubit associated with each edge. The gray highlighted region is the logical path of the corresponding logical operator. The nontrivial support of the logical operator lies along its logical path in all subrounds, even though the Pauli operator itself changes. Right: the inner logical operators, which are products of the logical operators shown on the left. Each inner logical is a homologically nontrivial path of check operators, which commutes with all check operators.

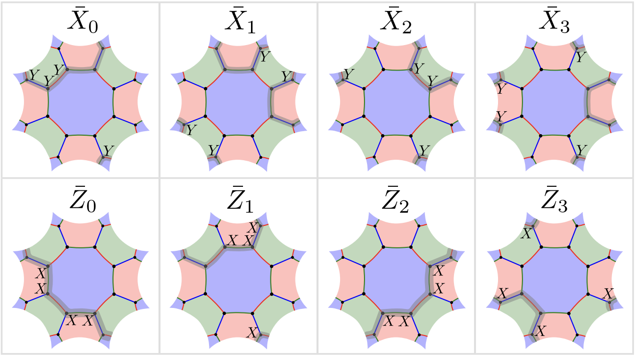

FIG. 7. A symplectic basis for the logical operators of the hyperbolic Floquet code derived from the 8.8.8 tiling of the Bolza surface, which has genus 2 and encodes four logical qubits into 16 physical data qubits. Opposite sides of the tiling are identified. The logical $\bar{X}$ and $\bar{Z}$ operators of logical qubit i are denoted by $\bar{X}_i$ and $\bar{Z}_i$, respectively. For each logical, the gray highlighted path is its associated homologically nontrivial logical path, which defines how the logical is updated in each subround (see Fig. 8).

embedded homological code immediately after a red subround. After the first red subround of $\mathcal{F}(\mathcal{T})$, each $\bar{X}_i$ logical of $\mathcal{F}(\mathcal{T})$ is defined to be an $\bar{X}_i$ logical of $\mathcal{C}(\mathcal{T}_R^*)$ (a homologically nontrivial cocycle), and each $\bar{Z}_i$ logical of $\mathcal{F}(\mathcal{T})$ is defined to be an $\bar{Z}_i$ logical of $\mathcal{C}(\mathcal{T}_R^*)$ (a homologically nontrivial cycle). See Fig. 6 (left) for an example using the toric honeycomb code. With this choice of logical basis, none of the representatives of the $\bar{X}_i$ or $\bar{Z}_i$ operators commute with all the check operators, so they are what Ref. [17] refers to as *outer logical operators*.

We need representatives of logical $\bar{X}_i$ and $\bar{Z}_i$ operators that commute with the green subround that immediately follows, however, the definition we have just given does not guarantee this property. This is because our choice of logicals so far is only defined up to an arbitrary element of ISG(R), which is generated by plaquette stabilizers as well as red checks, and the red checks do not generally commute with the green checks. We will now show how we can choose a representative for each $\bar{X}_i$ and $\bar{Z}_i$ operator that is guaranteed to commute with the subsequent green checks.

For each logical $\bar{X}_i$ or $\bar{Z}_i$ operator we associate a *logical path* $P(\bar{X}_i)$ or $P(\bar{Z}_i)$ (highlighted in gray in Figs. 6 and 7), a homologically nontrivial cycle on $\mathcal{T}$ that includes the nontrivial support of the logical operator. We can find each logical path using a procedure that lifts the homologically nontrivial cocycle (for $\bar{X}_i$) or cycle (for $\bar{Z}_i$) of the logical operator from the restricted lattice to a homologically nontrivial cycle on the color-code tiling $\mathcal{T}$. We will use the red restricted lattice $\mathcal{T}_R^*$ as an example. We can lift a *cocycle* Q in $\mathcal{T}_R^*$ to a logical path $P(X_i)$ in $\mathcal{T}$ by choosing a path that passes only through the red edges that Q crosses and around the borders of red faces. We can lift a *cycle* W in $\mathcal{T}_R^*$ to a logical path $P(Z_i)$ in $\mathcal{T}$ by finding a homologically nontrivial path in $\mathcal{T}$ that passes only through the borders of the blue and green faces in $\mathcal{T}$ associated with the nodes in W. Examples are shown in Fig. 6.

We have now explained how we obtain a logical path (a homologically nontrivial cycle in $\mathcal{T}$) for each logical operator of the Floquet code. This logical path of a logical operator is fixed throughout the measurement schedule, however the form of the logical Pauli operator does change after the measurement of each subround while keeping its support within its logical path. We now define the form of the logical Pauli operator by defining a mapping $\mathcal{V}_r^B$ which, for a given subround $r \in \{0, \ldots, 5\}$ and logical basis $B \in \{X, Z\}$, maps a cycle (or set of cycles) in $\mathcal{T}$ to a physical Pauli operator. We define $\mathcal{V}_r^B$ using the following table:

r (mod 6)	$\mathcal{V}_r^X$	$\mathcal{V}_r^Z$
0	Y on red edges	X on green edges
1	Y on blue edges	Z on green edges
2	X on blue edges	Z on red edges
3	X on green edges	Y on red edges
4	Z on green edges	Y on blue edges
5	Z on red edges	X on blue edges

Note that we have $\mathcal{V}_i^X = \mathcal{V}_{i+3 \pmod 6}^Z$ and $\mathcal{V}_i^Z = \mathcal{V}_{i+3 \pmod 6}^X$. As an example, given some cycle (or set of cycles) P on the tiling $\mathcal{T}$, the mapping $\mathcal{V}_0^X$ assigns a Pauli-Y operator to each qubit at the endpoint of a red edge within P and assigns an identity operaetor to all other qubits.

We then use this mapping to define the logical operator from its logical path. Concretely, we choose the representative of logical $\bar{X}_i$ to be $\mathcal{V}_r^X(P(\bar{X}_i))$ and similarly of logical $\bar{Z}_i$ to be $\mathcal{V}_r^Z(P(\bar{Z}_i))$. It is straightforward to verify that these are valid representatives of the logical operators and that they always commute with the check operator measurements in the subround that immediately follows.

Furthermore, for any face f of $\mathcal{T}$, then both $\mathcal{V}_r^X(f)$ and $\mathcal{V}_r^Z(f)$ are each either the identity or an element of the ISG in subround r. Hence, for each of the logical paths $P(\bar{X}_i)$ and $P(\bar{Z}_i)$ we could have picked *any* homologically nontrivial cycle belonging to the same homology class. This is because any two such choices belonging to the same homology class would differ by a set of faces in $\mathcal{T}$, and hence the corresponding Pauli operators obtained using $\mathcal{V}_r^X$ or $\mathcal{V}_r^X$ would differ by an element of the ISG. The use of the restricted lattice is helpful to understand the logical operators and paths, and to lift logical operators from the embedded homological code to the Floquet code (as we did for the numerical simulations in this work), but is not strictly necessary for constructing them; we could instead find homologically nontrivial cycles on the color-code lattice directly. Logical operators and paths for the toric honeycomb code are shown on the left of Fig. 6. Similarly, logical operators and paths of the hyperbolic Floquet code derived from the Bolza surface (the Bolza Floquet code) are shown in Fig. 7.

We have so far only described the form of the logical operators, however we will now explain how the logical operators are updated after each subround. After measuring the checks of color $c \in \{R, G, B\}$, we multiply into each logical operator the c-colored checks that lie within its logical path. This maps a logical $\mathcal{V}_r^X(P)$ or $\mathcal{V}_r^Z(P)$ to $\mathcal{V}_{r+1 \pmod 6}^X(P)$ or $\mathcal{V}_{r+1 \pmod 6}^Z(P)$, respectively, and guarantees that it will commute with the subround of checks that immediately follows. The following table describes the form of the logical operators over the period of six subrounds:

r (mod 6)	ISG	Form of $\bar{X}_i$	$\bar{X}_i$ in $\mathcal{T}_c^*$	Form of $\bar{Z}_i$	$\bar{Z}_i$ in $\mathcal{T}_c^*$
0	R	$\mathcal{V}_0^X(P(\bar{X}_i))$	cocycle in $\mathcal{T}_R^*$	$\mathcal{V}_0^Z(P(\bar{Z}_i))$	cycle in $\mathcal{T}_R^*$
1	G	$\mathcal{V}_1^X(P(\bar{X}_i))$	cycle in $\mathcal{T}_G^*$	$\mathcal{V}_1^Z(P(\bar{Z}_i))$	cocycle in $\mathcal{T}_G^*$
2	B	$\mathcal{V}_2^X(P(\bar{X}_i))$	cocycle in $\mathcal{T}_B^*$	$\mathcal{V}_2^Z(P(\bar{Z}_i))$	cycle in $\mathcal{T}_B^*$
3	R	$\mathcal{V}_3^X(P(\bar{X}_i))$	cycle in $\mathcal{T}_R^*$	$\mathcal{V}_3^Z(P(\bar{Z}_i))$	cocycle in $\mathcal{T}_R^*$
4	G	$\mathcal{V}_4^X(P(\bar{X}_i))$	cocycle in $\mathcal{T}_G^*$	$\mathcal{V}_4^Z(P(\bar{Z}_i))$	cycle in $\mathcal{T}_G^*$
5	B	$\mathcal{V}_5^X(P(\bar{X}_i))$	cycle in $\mathcal{T}_B^*$	$\mathcal{V}_5^Z(P(\bar{Z}_i))$	cocycle in $\mathcal{T}_B^*$

where here each row describes the form of the logical operators on $\mathcal{T}$ and $\mathcal{T}_c^*$ after a given subround r, which depends on r (mod 6). See Fig. 8 for the cycle of a logical operator in the Bolza Floquet code, as well as Fig. 1 of Ref. [19] for a similar figure using the honeycomb code. An $\bar{X}$-type logical operator in ISG(R) is mapped to a $\bar{Z}$-type logical operator in ISG(R) (and vice versa) every three subrounds by an automorphism of ISG(R). The same applies to ISG(G) and ISG(B). Since we must multiply checks into the logicals in each subround, the measurement of a logical operator in a Floquet code includes the product of measurement outcomes spanning a sheet in space time, rather than just a string of measurements in the final round as done for transversal measurement in the surface code.

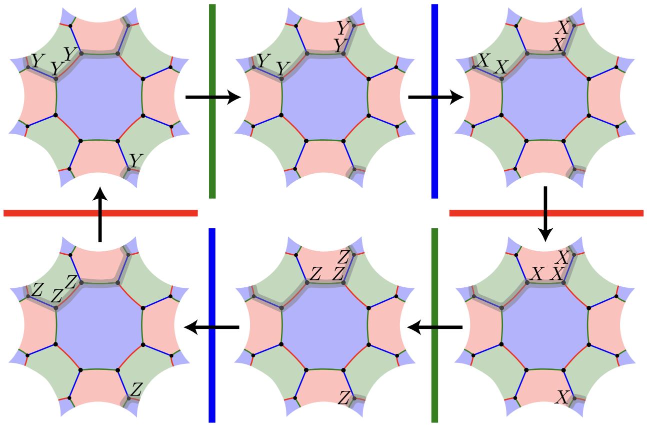

FIG. 8. Each logical operator changes over a periodic sequence of six subrounds (two full rounds), as shown here for a logical operator ($\bar{X}_0$ in Fig. 7) of the Bolza Floquet code. Opposite sides of the tiling are identified. Crossing a bar of color $c \in \{R, G, B\}$ means measuring the c checks and then multiplying into each logical operator the c checks that lie within its logical path. Note that the $\bar{X}_0$ logical operator (top left) is mapped to the $\bar{Z}_1$ logical operator (bottom right) every three subrounds and vice versa. This transition is an automorphism of ISG(R), which acts on each logical operator by multiplying by the inner logical operator that lies along the same logical path.

So far we have only considered outer logical operators, which move after each subround. However there are some logical operators, called *inner logical operators* in Ref. [17], which commute with all the checks. These inner logical operators act nontrivially on the encoded logical qubits of the Floquet code and are products of checks lying along homologically nontrivial cycles of the lattice. Each inner logical operator is formed from the product of an $\bar{X}$ and a $\bar{Z}$ logical operator associated with the same homologically nontrivial cycle of the lattice. For example, Fig. 6 (right) shows the two inner logical operators, $\bar{X}_0\bar{Z}_1$ and $\bar{Z}_0\bar{X}_1$, in the toric honeycomb code. One of the four inner logical operator of the Bolza Floquet code can be formed from the product of the $\bar{X}_0$ and $\bar{Z}_1$ logical operators in Fig. 7. However, even though the inner logical operators are formed from a product of check operators, the measurement schedule of the Floquet code is chosen such that it never measures the inner logical operators, only the plaquette stabilizers [17,23].

E. Circuits and noise models

In this work we consider two different noise models, corresponding to the EM3 and SD6 noise models used in Ref. [22]. Both noise models are characterized by a noise strength parameter $p \in [0, 1]$ (the physical error rate).

1. EM3 noise model

The EM3 noise model (entangling measurement three-step cycle) assumes that two-qubit Pauli measurements are available natively in the platform (and hence no ancilla qubits are needed), as is the case for Majorana-based architectures [20,29]. When implementing the measurement of a two-qubit Pauli operator $P^c \otimes P^c$, with probability p we insert an error chosen uniformly at random from the set $\{I, X, Y, Z\}^{\otimes 2} \times \{\text{flip}, \text{no flip}\}$. Here the two-qubit Pauli error in $\{I, X, Y, Z\}^{\otimes 2}$ is applied immediately before the measurement, and the "flip" operation flips the outcome of the measurement. We assume each measurement takes a single time step, and hence each cycle of three subrounds takes three time steps. Initialization of a qubit in the Z basis is followed by an X error, inserted with probability $p/2$. Similarly, the measurement of a data qubit in the Z basis is preceded by an X error, inserted with probability $p/2$. Single-qubit initialization and measurement in the X or Y basis is achieved using noisy Z-basis initialization or measurement and noiseless single-qubit Clifford gates. We do not need to define idling errors because data qubits are never idle in this noise model.

2. SD6 noise model

The SD6 noise model (standard depolarizing six-step cycle) facilitates the check measurements via an ancilla qubit for each edge of the color-code tiling. Therefore, this noise model introduces $|E| = 3|V|/2$ ancilla qubits and requires 2.5 times more qubits than the $EM3$ noise model for a given color-code tiling $\mathcal{T}$. Note that the distance in this noise model (the minimum number of independent error mechanisms required to flip a logical observable but no detectors) can be higher than $EM3$ for a given tiling, so we do not necessarily require 2.5 times more qubits for a given distance.

Each measurement of a two-qubit check operator $P^c \otimes P^c$ is implemented using CNOT gates and single-qubit Clifford gates, as well as measurement and reset of the ancilla qubit associated with the edge. We use the same circuit as in Ref. [19]. Each CNOT gate is followed by a two-qubit depolarizing channel, which with probability p inserts an error chosen uniformly at random from the set $\{I, X, Y, Z\}^{\otimes 2} \setminus \{I \otimes I\}$. Each single-qubit Clifford gates is followed by a single-qubit depolarizing channel, which with probability p inserts an error chosen uniformly at random from the set $\{X, Y, Z\}$. Single-qubit initialization is followed by an X error with probability p and single-qubit measurement is preceded by an X error with probability p. Each cycle of three subrounds takes six time steps, with CNOT gates, single-qubit gates, initialization and measurement all taking one time step each. In each time step in the bulk, each data qubit is either involved in a CNOT gate or a $C_{ZYX} := HS$ gate (which maps $Z \to Y \to X \to Z$ under conjugation) and therefore never idles.

F. Implementing hyperbolic and semi-hyperbolic connectivity

Although hyperbolic and semi-hyperbolic Floquet codes cannot be implemented using geometrically local connections in a planar Euclidean layout of qubits, they can instead be implemented using *biplanar* or *modular* architectures.

In a biplanar architecture, the connections (couplers) between qubits can be partitioned into two layers, such that the couplers within each layer do not cross. More concretely, consider the qubit connectivity graph $G_c = (V_c, E_c)$, which by definition contains an edge $(q_i, q_j) \in E_c$ if and only if qubits q_i and q_j directly interacted in the quantum circuit implementing the code. Recall that the *thickness* of a graph is the minimum number of planar graphs that the graph's edge set can be partitioned into. A graph is biplanar if it has thickness 2; in other words, G_c is biplanar if we can partition the edge set as $E_c = E_c^1 \cup E_c^2$, where the edge sets E_c^1 and E_c^2 each define a planar graph. It is known that the thickness of a graph with maximum degree d has thickness at most $t = \lceil d/2 \rceil$ (see Corollary 5 of Ref. [44], as well as Proposition 1 of Ref. [7] for this application to QEC). Therefore, since G_c has maximum degree 3 for any (semi-)hyperbolic Floquet code, it must also be biplanar. Note that the subsystem hyperbolic codes of Ref. [16] are also biplanar, since their qubit connectivity graphs always have degree 4. Biplanarity of the connectivity graph implies that we can fix the position of the qubits and have two layers of connections (e.g., above and below), each of which is planar. That both E_c^1 and E_c^2 can be planar with fixed qubit positions can be understood from Theorem 8 of Ref. [44], which shows that a planar graph always has a planar representation with nodes placed in arbitrary positions.

We can also implement (semi-)hyperbolic Floquet codes using a modular architecture [31], in which long-range links connect small modules of qubits. Examples of qubit platforms that could support modular architectures include ions [31,45,46], atoms [47,48], or superconducting qubits [49–53]. We break up the Floquet code into small modules, where each module supports planar Euclidean connectivity and long-range two-qubit Pauli measurements are used to connect different modules. For a hyperbolic Floquet code,

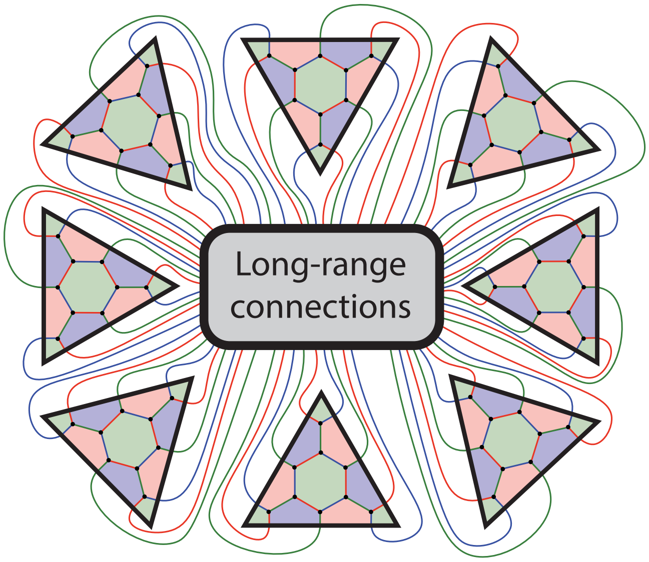

FIG. 9. A modular architecture for a semi-hyperbolic Floquet code. Each module has the connectivity of a planar Euclidean chip, and arbitrary connectivity is permitted between modules. In this example, each module is the result of fine graining the neighborhood of one vertex of the seed tiling $\mathcal{T}$ used to construct an $l = 3$ semi-hyperbolic Floquet code $\mathcal{F}(\mathcal{T}_3)$. Each long-range connection corresponds to a two-qubit measurement on qubits in two different modules. These inter-module connections may in general be long-range in order to facilitate the necessary connectivity of the semi-hyperbolic Floquet code.

variable length connections can be used to embed a small region of the lattice into each module [54]. For a semi-hyperbolic Floquet code, local regions of the code have identical connectivity to a honeycomb code; hence, each module can consist of a small hexagonal lattice of qubits obtained from fine graining the neighborhood of one vertex of the seed tiling $\mathcal{T}$ that the semi-hyperbolic Floquet code $\mathcal{F}(\mathcal{T}_l)$ was derived from (see Fig. 9). By connecting these modules using long-range two-qubit measurements, we can implement the connectivity required for the tiling of the closed hyperbolic surface. We note that additional connectivity might be necessary to implement logical gates and move logical qubits into and out of storage, but we leave this research problem to future work.

G. Detectors and decoding

In order to decode Floquet codes, we must make an appropriate choice of *detectors*. Following the definition in the Stim [55] documentation, a detector is a parity of measurement outcomes in the circuit that is deterministic in the absence of errors. In the literature, detectors have also been referred to as error-sensitive events [56] or checks [57,58]. In a Floquet code, each detector in the bulk of the schedule is formed by taking the parity of two consecutive measurements of a plaquette stabilizer. For example, in the honeycomb code, each plaquette stabilizer is the product of the six edge check measurements around a face, and so a detector is the parity of 12 measurement outcomes. For a Floquet code derived from an $r.g.b$ tiling, detectors are the parity of $2r$, $2g$ or $2b$ measurement outcomes. With this choice of detectors, it is known that any Floquet code can be decoded with minimum-weight perfect matching (MWPM) [25–28] or Union-Find (UF) [59,60] decoders [17,19,23] by defining a *detector graph* for a given noise model. Each node in the detector graph represents a detector and each edge represents an error mechanism that flips the detectors at its endpoints with probability p, and has an edge weight given by $\log((1-p)/p)$. Given this detector graph and an observed syndrome, MWPM or UF can be used to find a minimum-weight or low-weight set of edges consistent with the syndrome, respectively. Some additional accuracy can be achieved by also exploiting knowledge of hyperedge error mechanisms using a correlated matching [61] or belief-matching [62] decoder, however in this work we use the PyMatching implementation of MWPM [27] to decode and use Stim to construct the detector graphs and simulate the circuits [55].

IV. CONSTRUCTIONS

In this section, we present a performance analysis of our constructions of hyperbolic and semi-hyperbolic codes, including a comparison with honeycomb codes and surface codes. Section IV A contains our analysis of the code parameters, whereas simulations using EM3 and SD6 noise models are presented in Sec. IV B.

A. Code parameters

The number of data qubits $n = |V|$ in a Floquet code is determined by the number of vertices in the color-code tiling $\mathcal{T} = (V, E, F)$ and the number of logical qubits is given by the dimension of its first homology group $k = \dim H_1$. From Eqs. (6) and (7) we see that the encoding rate k/n is determined by the plaquette stabilizer weights and the fine-graining parameter l. No ancilla qubits are required for the EM3 noise model ($n_{\text{anc}} = 0$), whereas for the SD6 noise model we have $n_{\text{anc}} = |E| = 3n/2$ ancillas. We denote the total number of physical qubits by $n_{\text{tot}} := n + n_{\text{anc}}$.

The distance of a circuit implementing a Floquet code is the minimum number of error mechanisms required to flip at least one logical observable measurement outcome without flipping the outcome of any detectors. Therefore, the distance depends not only on the color-code tiling used to define the Floquet code, but also the choice of circuit and noise model. We consider three different types of distance of a Floquet code. One of these is a "circuit agnostic" distance, which we will refer to as the Floquet code's *embedded distance*. The embedded distance d of a Floquet code is defined to be the smallest distance of any of its three embedded 2D homological codes (the smallest homologically nontrivial cycle or cocycle in $\mathcal{T}_R^*$, $\mathcal{T}_G^*$, or $\mathcal{T}_B^*$). The embedded distance is more efficient to compute than the distance of a Floquet-code circuit, since it involves shortest-path searches on the 2D restricted lattices (using the method described in Appendix B of [38]), rather than on the much larger 3D detector graph associated with a specific circuit over many rounds (which we compute using `stim.Circuit.shortest_graphlike_error` in Stim [55]). We also consider the EM3 distance d_e of a Floquet code, which is the graphlike distance of its circuit for an EM3 noise model. Empirically, we find that the EM3 distance of a Floquet code almost always matches its embedded distance (see Table I), even though the embedded distance does not consider measurement errors or the details of a circuit-level noise model. Finally, we define the SD6 distance d_s to be the graphlike distance of the Floquet codes circuit for an SD6 noise model. While the SD6 noise model requires additional qubits (an ancilla qubit for each two-qubit check operator measurement), we note that the SD6 distance can in general be higher than the EM3 distance for a given Floquet code. This can be explained by the fact, for the EM3 noise model, a single error mechanism can result in any of the 15 nontrivial two-qubit Pauli operators acting on a pair of data qubits in the support of a pair measurement. In contrast, for the SD6 noise model, only a subset of these two-qubit Pauli operators can result from a single error mechanism.

TABLE I. Parameters of some of the hyperbolic Floquet codes we have constructed. Here n, k, and d are the number of physical data qubits, number of logical qubits, and embedded distance, respectively. The EM3 distance d_e and SD6 distance d_s are the graphlike distances of 16-round circuits using EM3 and SD6 noise models, respectively, computed using Stim [55]. We omit d_s where the calculation is too computationally expensive. We include only a code if d, d_e, or d_s is at least as large as the corresponding distance for all smaller codes in the family.

n	k	d	kd^2/n	d_e	d_s
8.8.8					
16	4	2	1.00	2	2
32	6	2	0.75	2	3
64	10	2	0.62	2	4
256	34	4	2.12	4	4
336	44	3	1.18	3	6
336	44	4	2.10	4	6
512	66	4	2.06	4	6
720	92	4	2.04	4	6
1024	130	4	2.03	4	6
1296	164	4	2.02	4	6
1344	170	6	4.55	...	6
2688	338	6	4.53	...	...
4896	614	6	4.51	...	...
5376	674	6	4.51	...	...
5760	722	6	4.51	...	...
4.10.10					
120	8	3	0.60	3	4
160	10	4	1.00	4	4
320	18	5	1.41	4	6
600	32	6	1.92	6	6
3600	182	8	3.24	...	...
10 240	514	9	4.07	...	...
19 200	962	10	5.01	...	...
4.8.10					
240	8	4	0.53	4	6
640	18	6	1.01	6	8
1440	38	8	1.69	8	8
7200	182	10	2.53	...	...

To achieve an EM3 distance d_e for either the toric or planar honeycomb code, we need at least $2d_e$ columns of qubits and $3d_e$ rows, and hence use a patch with dimensions $2d_e \times 3d_e$ using $n_{\text{tot}} = n = 6d_e^2$ physical qubits [19–22]. See Fig. 1 for an example of a 4×6 toric honeycomb code ($d_e = 2$).

1. Hyperbolic Floquet codes

Although planar and toric honeycomb codes can only encode one or two logical qubits, respectively, hyperbolic Floquet codes can encode a number of logical qubits k proportional to the number of physical qubits n. From Eq. (6) we see that a hyperbolic Floquet code has a finite encoding rate $k/n = \dim H_1/|V| > R$, where R is 1/8, 1/40 and 1/20 for the 8.8.8, 4.8.10 and 4.10.10 color-code tilings, respectively.

Hyperbolic surface codes are known to have $O(\log n)$ distance scaling [36,63–65] and it follows that the embedded distance of a hyperbolic Floquet code therefore also scales as $d = O(\log n)$ [23]. This leads to the parameters of hyperbolic Floquet codes satisfying $kd^2/n = O(\log^2(n))$, which is an asymptotic improvement over the parameters of Euclidean surface codes and Floquet codes, for which $kd^2/n = O(1)$.

We give the parameters of some of the hyperbolic Floquet codes we constructed in Table I. These belong to three families of hyperbolic Floquet codes, derived from 8.8.8, 4.10.10, and 4.8.10 tessellations. We have picked these tilings as they have the lowest stabilizer weight and give the highest relative distance, although they give the lowest encoding rate, see discussion in Ref. [36]. All of the codes constructed have kd^2/n exceeding that of planar and toric honeycomb codes for which kd^2/n is 1/6 and 1/3, respectively. The largest improvement in parameters is given by the $[[19200, 962, 10]]$ 4.10.10 hyperbolic Floquet code, which has $kd^2/n = 5.01$, a 30.1 times larger ratio than is obtained using the planar honeycomb code.

2. Semi-hyperbolic Floquet codes

One potential concern is that hyperbolic Floquet codes can only achieve error suppression that is polynomial in the system size, owing to their $O(\log n)$ distance scaling. However, in this section we will show that we can achieve exponential error suppression using the semi-hyperbolic color-code tilings we introduced in Sec. IV A 2, which fine grain the hyperbolic lattice and have $O(\sqrt{n})$ distance scaling, while still retaining an advantage over Euclidean Floquet codes.

We define a family of semi-hyperbolic Floquet codes from a hyperbolic color-code tiling $\mathcal{T}$ and the fine-graining parameter l to obtain a semi-hyperbolic tiling $\mathcal{T}_l$ (see Sec. II D), which we use to define the semi-hyperbolic Floquet code $\mathcal{F}(\mathcal{T}_l)$. Let us denote the parameters of the Floquet code $\mathcal{F}(\mathcal{T})$ derived from $\mathcal{T}$ by $[\![n,k,d]\!]$ and we denote the parameters of $\mathcal{F}(\mathcal{T}_l)$ by $[\![n_l,k_l,d_l]\!]$. The topology of the surface is unchanged by the fine-graining procedure so we have $k_l = k$ and from Eq. (7) we have that $n_l = l^2 n$. The minimum length of a homologically nontrivial cycle or cocycle in the restricted lattices of $\mathcal{T}_l$ will increase by a factor proportional to l, and so we have $d_l \geq C_1 l d$ for some constant C_1. This leads to parameters $[\![n_l,k_l,d_l]\!]$ of the semi-hyperbolic Floquet code $\mathcal{F}(\mathcal{T}_l)$ satisfying $k_l d_l^2/n_l \geq C_1^2 k d^2/n$. If we fix the tiling $\mathcal{T}$ from which we define a family of semi-hyperbolic Floquet codes $\mathcal{F}(\mathcal{T}_l)$ then we have $k_l d_l^2/n_l \geq C_2$ where $C_2 = C_1^2 k d^2/n$ is a constant determined by $\mathcal{T}$. Although a family of semi-hyperbolic Floquet codes defined this way does not have an *asymptotic* improvement over the parameters of the honeycomb code, it is still possible to obtain a constant factor improvement if $C_2 > 1/6$.

From our analysis of the parameters of the semi-hyperbolic Floquet codes we have constructed, we find that this constant factor improvement over the planar honeycomb code can be substantial. In Fig. 10 we show the parameters of families of semi-hyperbolic Floquet codes derived from 8.8.8 color-code tilings. For some families of semi-hyperbolic codes, we have $C_1 = 1$, i.e., the line is horizontal on the plot, at least for the system sizes we consider. For these semi-hyperbolic Floquet-code families, we can increase the distance by a factor of l using exactly l^2 times more physical qubits. Many of these families of semi-hyperbolic codes retain an order-of-magnitude reduction in qubit overhead relative to the planar honeycomb code even for very large distances.

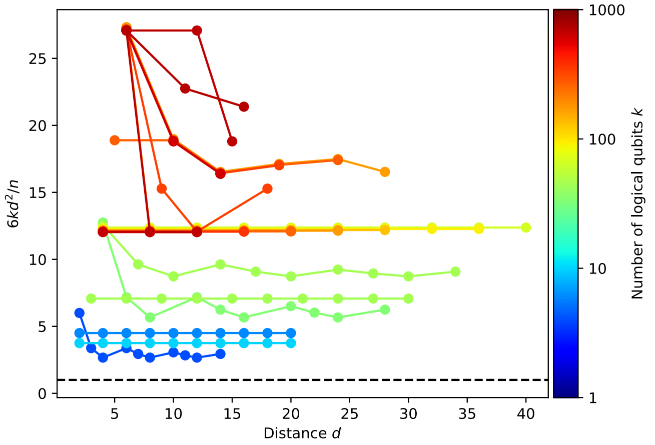

FIG. 10. Parameters of semi-hyperbolic Floquet codes. Each line corresponds to a family of semi-hyperbolic codes constructed by fine graining an 8.8.8 color-code tiling, with l increasing by one from left to right (starting at $l = 1$). We plot $6kd^2/n$ on the y axis against d on the x axis (where d is the embedded distance of the Floquet code). Note that $6kd^2/n = 1$ for the planar Floquet code (black dashed line) and hence this ratio corresponds to the multiplicative saving in the number of physical qubits relative to the planar Floquet code to obtain a fixed target k and d.

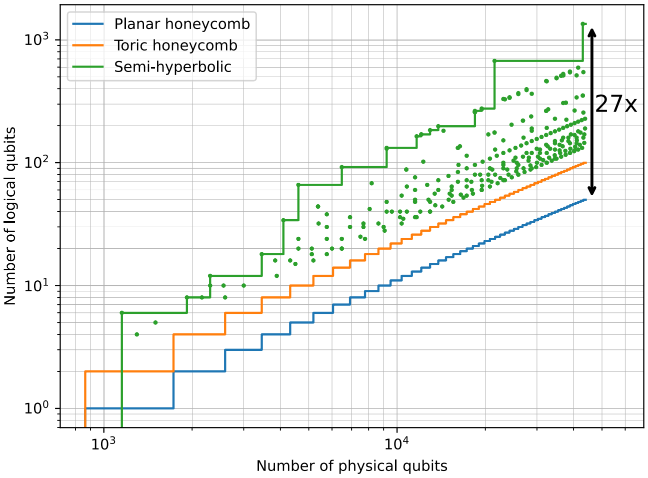

FIG. 11. The number of logical qubits that can be encoded with embedded distance d at least 12 using multiple copies of a semi-hyperbolic, toric honeycomb or planar honeycomb Floquet code. The EM3 distance is the distance of the code for the EM3 noise model, for which two-qubit Pauli operators can be measured directly without the need for ancillas. Each green circle corresponds to multiple copies of a semi-hyperbolic Floquet code, and the green line is the Pareto frontier for the copies of semi-hyperbolic codes considered.

In Fig. 11 we plot the number of logical qubits that can be encoded (with embedded distance at least 12) as a function of the number of physical qubits available, using multiple copies of semi-hyperbolic Floquet codes and honeycomb codes. We can encode up to 27 times more logical qubits using semi-hyperbolic Floquet codes relative to planar honeycomb codes, including a >10 times increase in the encoding rate for smaller system sizes using fewer than 10 000 physical qubits.

B. Simulations

So far we have focused our analysis on code parameters, however logical error rate performance for realistic noise models is more relevant to understand the practical utility of the constructions. In particular, hyperbolic Floquet codes derived from an $r.g.b$ tiling have higher plaquette stabilizer weights $2r$, $2g$, and $2b$, and each detector is formed from the parity of $2r$, $2g$ or $2b$ check operator measurements. We might be concerned that these higher stabilizer weights lead to worse logical error rate performance, despite their improved code parameters. However, by performing numerical simulations, we show in this section that hyperbolic Floquet codes have a logical error rate performance that is comparable to that of the honeycomb code even at high physical error rates, but with significantly reduced resource overheads.

Our numerical simulations compare the performance of semi-hyperbolic Floquet codes with planar honeycomb codes (for the EM3 and SD6 noise models) as well as rotated planar surface codes (for the SD6 noise models). All simulations were carried out using Stim [55] to simulate the circuits and construct matching graphs and PyMatching [27] to decode. For the simulations of planar honeycomb codes and surface codes, we used the open-source simulation software [66] written for Ref. [22]. In all figures, we define the logical error rate as the probability that *any* of the encoded logical qubits fail. Note that we have made some Stim circuits of hyperbolic and semi-hyperbolic Floquet codes, as well as tables of their parameters, available on GitHub [67].

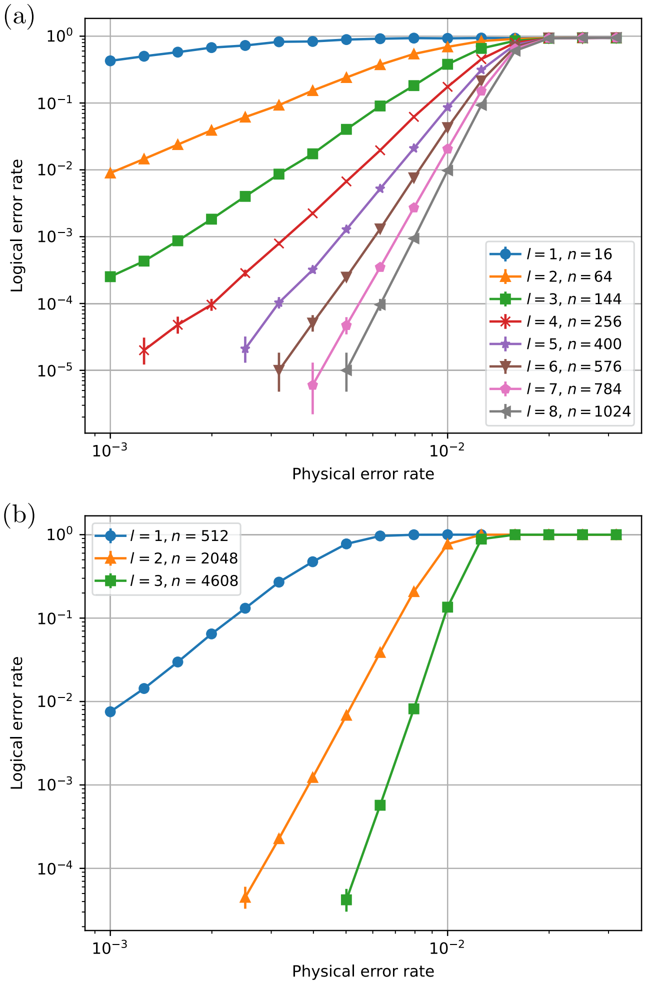

FIG. 12. Threshold plots of two families of semi-hyperbolic Floquet codes, using an EM3 noise model and 64 rounds. (a) A family of semi-hyperbolic Floquet codes derived from the Bolza surface (see Table II for code parameters). (b) A family of semi-hyperbolic Floquet codes derived from an 8.8.8 tiling of a genus-33 closed hyperbolic surface (with code parameters given in Table III). For both plots, the legend gives the fine-graining parameter l and number of physical qubits n and we observe a threshold of around 1.5%–2%, consistent with the threshold of the honeycomb code [19,20,22].

TABLE II. Parameters of a family of semi-hyperbolic Floquet codes derived from the Bolza surface. Here, n_{tot} is the total number of physical qubits (including ancillas) for the SD6 noise model. We verified that the EM3 distance d_e equals the embedded distance d reported here for all codes in the table.

l	k	n	d	kd^2/n	n_{tot}	d_s	kd_s^2/n_{tot}
1	4	16	2	1.00	40	2	0.40
2	4	64	3	0.56	160	4	0.40
3	4	144	4	0.44	360	6	0.40
4	4	256	6	0.56	640	8	0.40
5	4	400	7	0.49	1000	10	0.40
6	4	576	8	0.44	1440	12	0.40
7	4	784	10	0.51	1960	14	0.40
8	4	1024	11	0.47	2560	16	0.40
9	4	1296	12	0.44	3240	18	0.40
10	4	1600	14	0.49	4000	20	0.40

1. Thresholds

We expect families of semi-hyperbolic Floquet codes to have the same threshold as planar and toric honeycomb codes, since the bulk of a semi-hyperbolic Floquet code looks identical to a honeycomb code as we increase the fine-graining parameter l. We demonstrate this numerically for two families of semi-hyperbolic Floquet codes in Fig. 12, which each have a threshold of at least 1.5% to 2%,

TABLE III. Parameters of a family of semi-hyperbolic Floquet codes derived from an 8.8.8 tiling of a genus-33 closed hyperbolic surface.

l	k	n	d	kd^2/n
1	66	512	4	2.06
2	66	2048	8	2.06
3	66	4608	12	2.06
4	66	8192	16	2.06
5	66	12 800	20	2.06
6	66	18 432	24	2.06
7	66	25 088	28	2.06
8	66	32 768	32	2.06
9	66	41 472	36	2.06
10	66	51 200	40	2.06

consistent with the threshold of the honeycomb code [19,20,22].

2. Overhead reduction

In this section we study the reduction in qubit overhead that can be achieved using hyperbolic Floquet codes below threshold, for both EM3 and SD6 noise models.

Figure 13 analyzes the performance of hyperbolic Floquet codes requiring only a few hundred physical qubits. In Fig. 13(a) we simulate the performance of a $[[600, 32, d_e = 6]]$ hyperbolic Floquet code derived from a 4.10.10 tiling compared to 32 copies of planar honeycomb codes for the Majorana-inspired EM3 noise model, which assumes direct pair measurements. Achieving the same performance as the hyperbolic Floquet code using planar honeycomb codes requires 6912 physical qubits (11.5 times more qubits than the hyperbolic Floquet code). Furthermore, planar honeycomb codes are already known

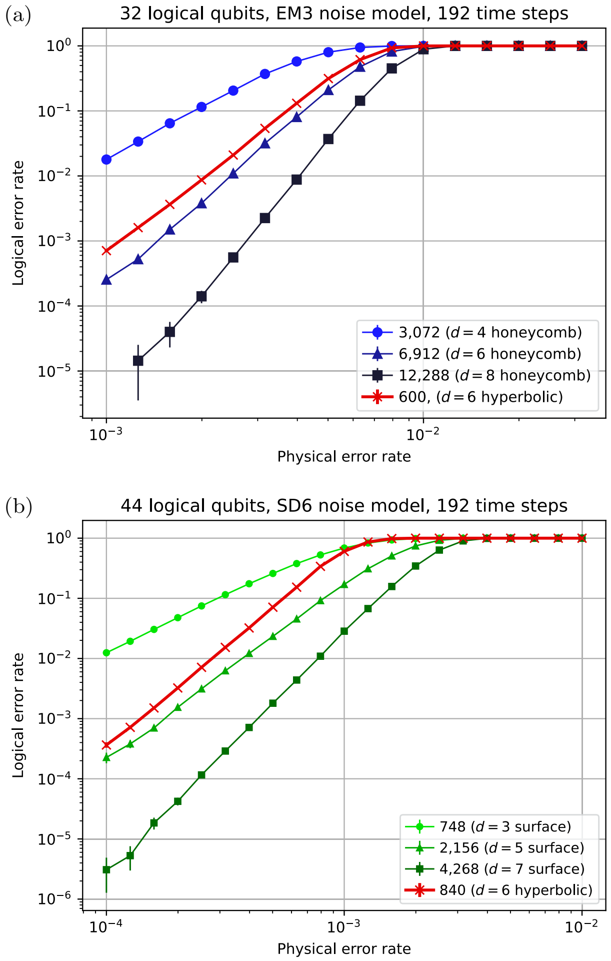

FIG. 13. (a) Performance of a $[[600, 32, d_e = 6]]$ hyperbolic Floquet code (derived from a 4.10.10 tiling) compared to 32 copies of planar honeycomb codes, for an EM3 noise model (direct pair measurements). For the honeycomb codes we plot $1 - (1 - p_{\log})^{32}$ where $p_{\log}$ is the logical error rate for a single patch (one logical qubit). (b) Performance of a $[[336, 44, d_s = 6]]$ hyperbolic Floquet code (from an 8.8.8 tiling) compared to 44 copies of rotated surface codes [i.e., plotting $1 - (1 - p_{\log})^{44}$], for an SD6 noise model. For both (a),(b), we simulate 192 time steps and give the average of the logical X and Z error rates on the y axis. The legend gives the total number of physical qubits (including ancillas for the SD6 noise model).

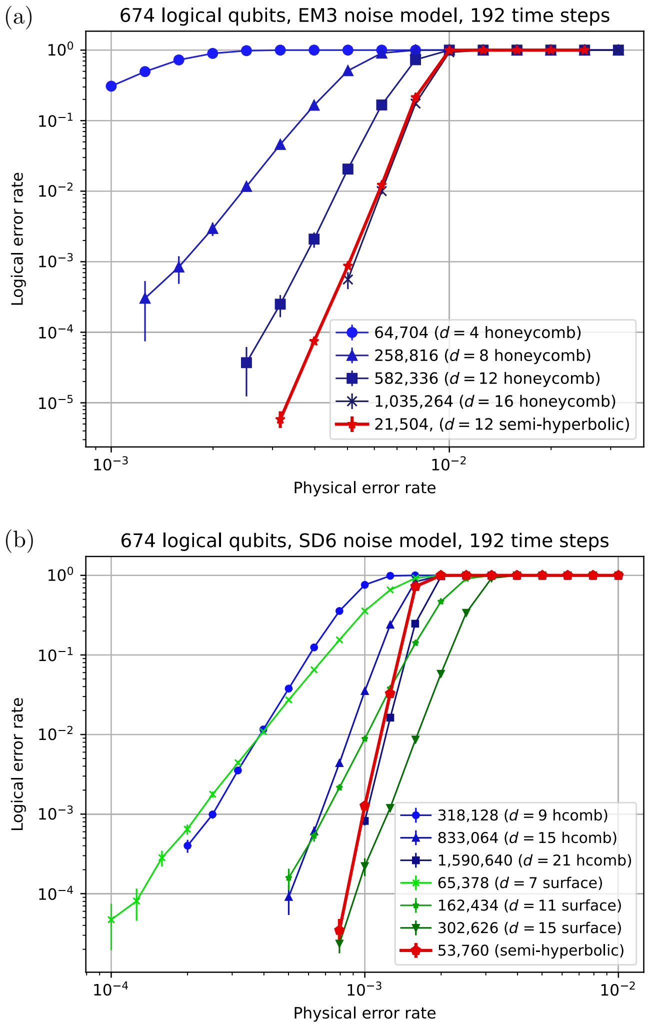

FIG. 14. Logical error rate vs physical error rate for protecting 674 logical qubits using 674 copies of planar honeycomb codes (shades of blue) and a $[[21504, 674, 12]]$ semi-hyperbolic Floquet code derived from a 8.8.8 tessellation. In (a) an EM3 noise model is used (direct pair measurements without the use of an ancilla) whereas in (b) an SD6 noise model (ancilla assisted) is used and we also compare with standard surface-code circuits (shades of green). For all codes, we simulate 192 time steps and give the average of the logical X and Z error rates on the y axis. For the honeycomb and surface codes, we plot $1 - (1 - p_{\log})^{674}$ where $p_{\log}$ is the logical error rate for a single patch (one logical qubit). The legend gives the total number of physical qubits (including ancillas for the SD6 noise model).

to be two to six times more efficient than surface codes for comparable noise models that assume compilation into direct two-qubit Pauli measurements [20,29,30].

We also show a comparison of a $[[336, 44, d_s = 6]]$ hyperbolic Floquet code (from an 8.8.8 tiling) in Fig. 13(b), compared to 44 copies of rotated surface codes, for the standard circuit-level depolarizing (SD6 noise model). Including ancillas, the hyperbolic Floquet code requires 840 physical qubits whereas the surface codes require 2156 physical qubits (2.6 times more than the hyperbolic Floquet code).

We consider larger examples, which achieve an even larger relative reduction in resources compared to Euclidean codes, in Fig. 14. In Fig. 14(a) we consider the EM3 noise model, which assumes noisy direct measurement of two-qubit Pauli operators. We simulate the performance of an $l = 2$ $[[21504, 674, 12]]$ semi-hyperbolic Floquet code, which belongs to the family of semi-hyperbolic Floquet codes described in Table IV. We compare the logical error rate performance of the $[[21504, 674, 12]]$ code with honeycomb codes for encoding 674 logical qubits over 192 time steps. We find that the semi-hyperbolic Floquet code matches the performance of $d = 16$ honeycomb codes using 1 035 264 physical qubits, corresponding to a 48 times reduction in qubit overhead. Therefore, given a platform permitting direct two-qubit Pauli measurements and semi-hyperbolic qubit connectivity (e.g., using a modular architecture), our results suggest that semi-hyperbolic Floquet codes can require over 100 times fewer physical qubits than the surface code circuits from Refs. [20,29,30] for an EM3 noise model at around $p = 0.1\%$.

Using a least-squares fit of the four left-most data points to extrapolate the red curve in Fig. 14(a) to lower physical error rates, we estimate that the $[[21504, 674, 12]]$ semi-hyperbolic Floquet code has a logical error rate of $\approx 1.8 \times 10^{-11}$ at $p = 0.1\%$ for the EM3 noise model. This implies a logical error rate per logical qubit of $\approx 2.7 \times 10^{-14}$, or $\approx 4.2 \times 10^{-16}$ per logical qubit per round. We therefore project that we can reach well below the "teraquop regime" [19] of 10^{-12} logical failure rates using only 32 physical qubits per logical qubit. In contrast, planar honeycomb codes have been shown to require from 600 [19] to 2000 [20] physical qubits per logical qubit to achieve the teraquop regime using the same noise model in prior work.

TABLE IV. Parameters of a family of semi-hyperbolic Floquet codes encoding 674 logical qubits, constructed by fine graining an 8.8.8 tiling of a genus-337 hyperbolic surface. Note that d here denotes the embedded distance.

l	k	n	d	kd^2/n
1	674	5376	6	4.51
2	674	21 504	12	4.51
3	674	48 384	15	3.13

In Fig. 14(b), we instead consider the SD6 noise model, which uses an ancilla qubit to measure each check operator in the presence of standard circuit-level depolarizing noise. We compare a semi-hyperbolic Floquet code [the same as used in Fig. 14(a)] with honeycomb and surface codes for encoding 674 logical qubits over 192 time steps. In this noise model, we find that the semi-hyperbolic Floquet code matches the performance of planar honeycomb codes with SD6 distance $d_s = 21$ which require $n_{\text{tot}} = 1\,590\,640$ qubits, whereas the semi-hyperbolic Floquet code only needs 53 760 qubits (including ancillas), a 29.6 times saving in resources. At a physical error rate slightly below 0.1%, our semi-hyperbolic Floquet code achieves a 5.6

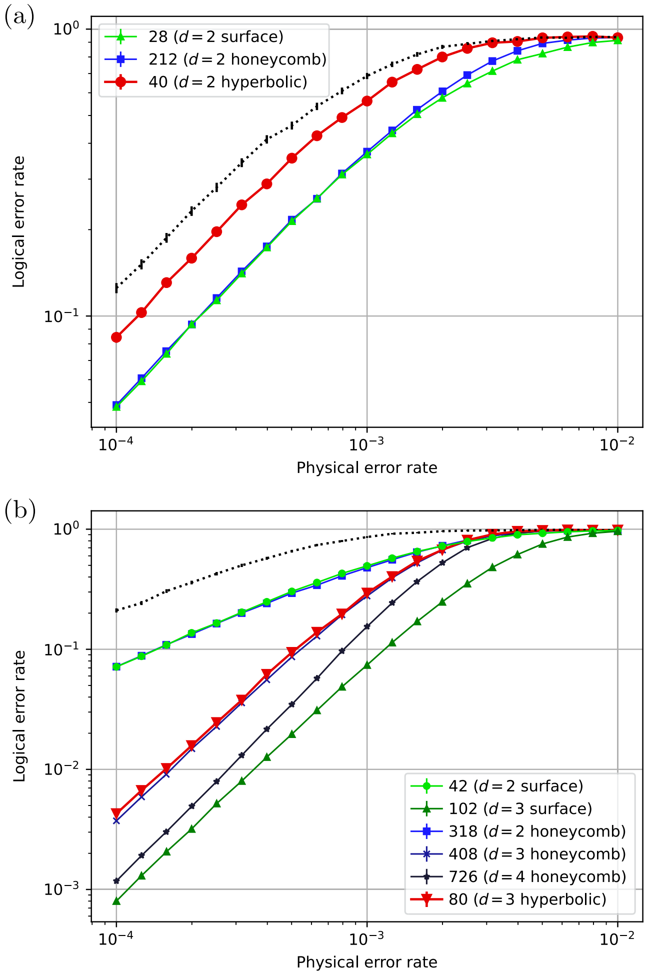

FIG. 15. Logical error rates vs physical error rates for encoding k logical qubits using k copies of small planar honeycomb or surface codes, compared to a hyperbolic Floquet code encoding k logical qubits, for 96 time steps using a standard depolarizing (SD6) noise model. In (a) we have $k = 4$ for the Bolza Floquet code, which has SD6 distance $d_s = 2$, whereas in (b) we have $k = 6$ for a hyperbolic Floquet code with $d_s = 3$. The hyperbolic Floquet codes in (a),(b) are both derived from 8.8.8 tilings. The legend gives the total number of qubits, with the SD6 distance given in brackets.

times reduction in qubit overhead relative to surface codes, matching the logical error rate of $d = 15$ surface codes, which use 302 626 physical qubits. Note that the curve for the $d = 15$ surface code has a shallower gradient than that of the semi-hyperbolic Floquet code, indicating that the semi-hyperbolic Floquet code has a higher SD6 distance.

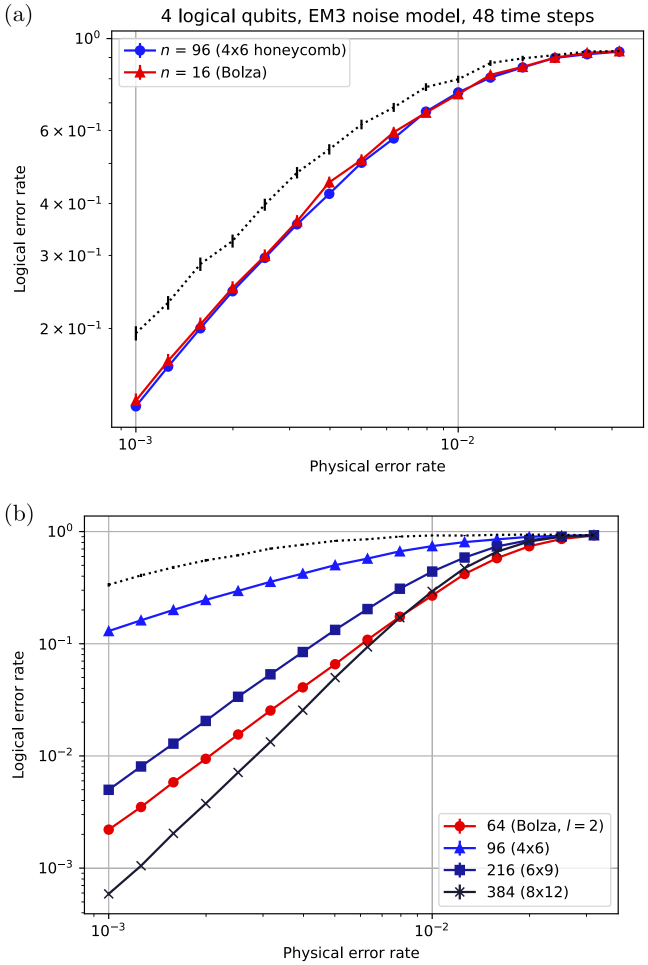

FIG. 16. Logical error rates vs physical error rates for encoding four logical qubits using four copies of small planar honeycomb codes (blue) or a (semi-)hyperbolic Floquet code encoding four logical qubits (red), for the EM3 noise model (check operators are measured directly without ancillas). In (a) we use a hyperbolic code derived from the Bolza surface that has distance 2 in this noise model, whereas in (b) we use an $l = 2$ semi-hyperbolic code derived from the Bolza surface that has distance 3. The legend gives the total number of qubits, with the dimension of the lattice given in parentheses for the planar honeycomb codes. All circuits use 16 rounds of measurements, and the y axis gives the average of the logical X and Z error rates. The black dotted line is the logical failure rate for the (semi-)hyperbolic code for a decoder, which always predicts that no logical observable has been flipped. Note that the 4 × 6, 6 × 9 and 8 × 12 planar honeycomb codes have distances 2, 3, and 4, respectively, for this EM3 noise model.

Therefore, we would expect that the semi-hyperbolic Floquet code will offer a bigger advantage over surface codes at lower physical error rates. If the semi-hyperbolic Floquet code has SD6 distance $d_s = 21$, then it would have similar performance to $d = 21$ surface codes at very low physical error rates, which would use 593 794 physical qubits (881 per patch), 11 times more than the 53 760 needed for our semi-hyperbolic Floquet code.

C. Small examples

We also study the performance of small hyperbolic and semi-hyperbolic Floquet codes that might be amenable to experimental realization in the near future. In Fig. 15 we show the performance of a small hyperbolic Floquet codes (with SD6 distance 2 and 3) for an SD6 noise model, compared to small planar honeycomb and surface codes. To achieve a given logical performance, the hyperbolic Floquet codes in both Figs. 15(a) and 15(b) use around five times fewer physical qubits than planar honeycomb codes but use a similar number of physical qubits to surface codes.

For the EM3 noise model, we compare to planar honeycomb codes in Fig. 16(a) and find that the $d = 2$ hyperbolic Floquet code derived from the Bolza surface (using 16 qubits) has identical performance to four copies of a $d = 2$ planar honeycomb code using six times more physical qubits. We also study the performance (again for the EM3 noise model) of an $l = 2$ semi-hyperbolic Floquet code derived from the Bolza surface in Fig. 16(b) and find it to be 3.4 times more efficient than four copies of $d = 3$ planar honeycomb codes achieving a similar logical error rate.

V. CONCLUSION

In this work, we have constructed Floquet codes derived from color-code tilings of closed hyperbolic surfaces. These constructions include hyperbolic Floquet codes, obtained from hyperbolic tilings, as well as semi-hyperbolic Floquet codes, which are derived from hyperbolic tilings via a fine-graining procedure. We have given explicit examples of hyperbolic Floquet codes with improved encoding rates relative to honeycomb codes and have shown how semi-hyperbolic Floquet codes can retain this advantage while enabling improved $\sqrt{n}$ distance scaling.

We have used numerical simulations to analyze the performance of our constructions for two noise models: a Majorana-inspired “EM3” noise model, which assumes direct noisy two-qubit measurements, and a standard circuit-level depolarising “SD6” noise model, which uses ancilla qubits to assist the measurement of check operators. For the EM3 noise model, we compare a semi-hyperbolic Floquet code that encodes 674 logical qubits into 21 504 physical qubits with 674 copies of planar honeycomb codes, and find that the semi-hyperbolic Floquet code uses 48 times fewer physical qubits at physical error rates as high as 0.3% to 1%. We demonstrate that this semi-hyperbolic Floquet code can achieve logical error rates below 10^{-12} at 0.1% EM3 noise using as few as 32 physical qubits per logical qubit. This is a significant improvement over planar honeycomb codes, that require 600 to 2000 physical qubits per logical qubit in the same regime [19,20]. For the SD6 noise model at a noise strength of 0.1%, we show that the same semi-hyperbolic Floquet code uses around 30 times fewer physical qubits than planar honeycomb codes, and around 5.6 times fewer qubits than surface codes. We also construct several small examples of hyperbolic and semi-hyperbolic Floquet codes amenable to near-term experiments, including a 16-qubit hyperbolic Floquet code derived from the Bolza surface. These small instances are around three to six times more efficient than planar honeycomb codes and are comparable to standard surface codes in the SD6 noise model.

An interesting avenue of future research might be to study the performance of other recent variants and generalizations of Floquet codes [18,40,42,68–73]. For example, the CSS Floquet code introduced in Refs. [42,68] is also defined from any color-code tiling, but with a modified choice of two-qubit check operators. CSS Floquet codes can therefore also be constructed from the hyperbolic and semi-hyperbolic colour code tilings in Sec. II and, since they have the same embedded homological codes as the Floquet codes we study here, we would expect them to offer similar advantages over CSS honeycomb codes.

Another possible research direction would be to study and design more concrete realisations of hyperbolic and semi-hyperbolic Floquet codes in modular architectures. This could include studying protocols for realizing these modular architectures where links between modules are more noisy or slow [74–77], motivated by physical systems that can be used to realise them, such as ions [31,45,46], atoms [47,48], or superconducting qubits [49–53].

Finally, we have focused here on error-corrected quantum *memories*, but more work is needed to demonstrate how our constructions can be used to save resources within the context of a quantum computation. We note that the Floquet code schedule applies a logical Hadamard to all logical qubits in the code every three subrounds; however, to be useful in a computation we must either be able to read and write logical qubits to the memory or address logical qubits individually with a universal gate set. We expect that reading and writing of logical qubits to and from honeycomb codes can be achieved using Dehn twists and lattice surgery by adapting the methods developed in Ref. [38] for hyperbolic surface codes to (semi-)hyperbolic Floquet codes, however, a more detailed analysis is required. Further work could also investigate (semi-)hyperbolic Floquet codes that admit additional transversal logical operations [78,79], such as fold-transversal logical gates [80], examples of which have been demonstrated for hyperbolic surface codes.

ACKNOWLEDGMENTS

O.H. would like to thank Craig Gidney for helpful discussions, including suggesting Fig. 11, as well as Barbara Terhal and Matt McEwen for discussions that improved the section on biplanar and modular architectures. O.H. acknowledges support from the Engineering and Physical Sciences Research Council (Grant No. EP/L015242/1) and a Google PhD fellowship.

APPENDIX: ADDITIONAL RESULTS

Figure 17 gives the number of logical qubits that can be encoded with embedded distance at least 20 and 30 using semi-hyperbolic Floquet codes or honeycomb codes (these plots are slight variants of Fig. 11).

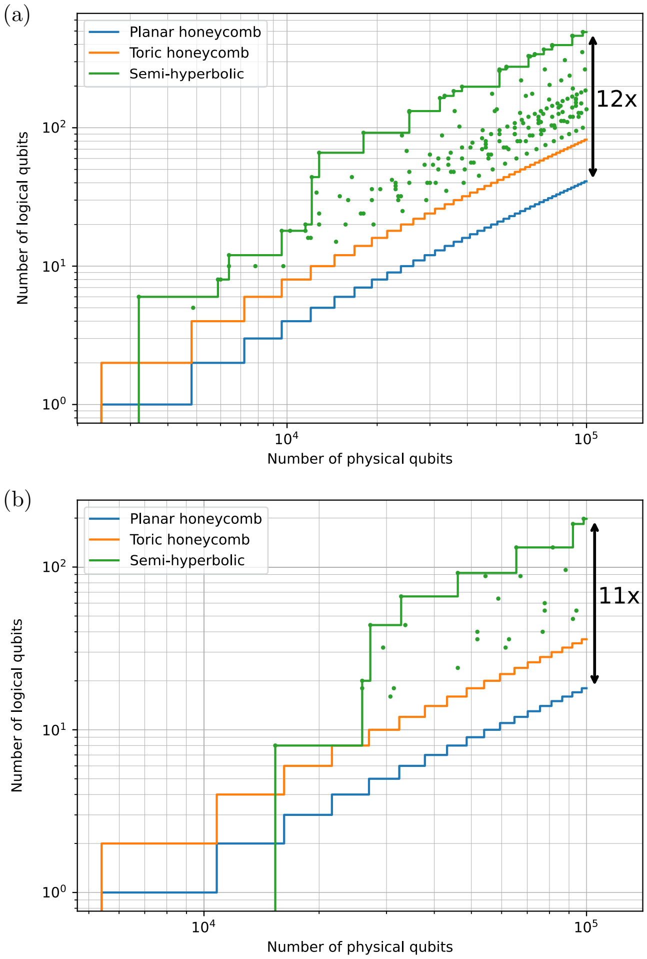

FIG. 17. The number of logical qubits that can be encoded using multiple copies of a semi-hyperbolic, toric honeycomb, or planar honeycomb Floquet code. In (a) the distance is required to be at least 20 for an EM3 noise model, whereas in (b) the distance is at least 30.

[1] Nikolas P. Breuckmann and Jens Niklas Eberhardt, Quantum low-density parity-check codes, PRX Quantum **2**, 040101 (2021).

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

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

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

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

[6] Anthony Leverrier and Gilles Zémor, in *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE, Denver, CO, USA, 2022), p. 872.

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

[8] J. Conrad, C. Chamberland, N. P. Breuckmann, and B. M. Terhal, The small stellated dodecahedron code and friends, Philos. Transact. R. Soc. A: Math. Phys. Eng. Sci. **376**, 20170323 (2018).

[9] Muyuan Li and Theodore J. Yoder, in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, Denver, CO, USA, 2020), p. 109.

[10] Christopher A. Pattison, Anirudh Krishna, and John Preskill, Hierarchical memories: Simulating quantum LDPC codes with local gates, ArXiv:2303.04798.

[11] David Poulin, Stabilizer formalism for operator quantum error correction, Phys. Rev. Lett. **95**, 230504 (2005).

[12] Dave Bacon, Operator quantum error-correcting subsystems for self-correcting quantum memories, Phys. Rev. A **73**, 012340 (2006).

[13] Martin Suchara, Sergey Bravyi, and Barbara Terhal, Constructions and noise threshold of topological subsystem codes, J. Phys. A: Math. Theor. **44**, 155301 (2011).

[14] Sergey Bravyi, Guillaume Duclos-Cianci, David Poulin, and Martin Suchara, Subsystem surface codes with three-qubit check operators, Quantum Inf. Comput. **13**, 963 (2013).

[15] Aleksander Kubica and Michael Vasmer, Single-shot quantum error correction with the three-dimensional subsystem toric code, Nat. Commun. **13**, 6272 (2022).

[16] Oscar Higgott and Nikolas P. Breuckmann, Subsystem codes with high thresholds by gauge fixing and reduced qubit overhead, Phys. Rev. X **11**, 031039 (2021).

[17] Matthew B. Hastings and Jeongwan Haah, Dynamically generated logical qubits, Quantum **5**, 564 (2021).

[18] Alex Townsend-Teague, Julio Magdalena de la Fuente, and Markus Kesselring, Floquetifying the colour code, ArXiv:2307.11136.

[19] Craig Gidney, Michael Newman, Austin Fowler, and Michael Broughton, A fault-tolerant honeycomb memory, Quantum **5**, 605 (2021).

[20] Adam Paetznick, Christina Knapp, Nicolas Delfosse, Bela Bauer, Jeongwan Haah, Matthew B. Hastings, and Marcus P. da Silva, Performance of planar Floquet codes with Majorana-based qubits, PRX Quantum **4**, 010310 (2023).

[21] Jeongwan Haah and Matthew B. Hastings, Boundaries for the honeycomb code, Quantum **6**, 693 (2022).

[22] Craig Gidney, Michael Newman, and Matt McEwen, Benchmarking the planar honeycomb code, Quantum **6**, 813 (2022).

[23] Christophe Vuillot, Planar Floquet codes, ArXiv:2110.05348.

[24] H. Bombin and M. A. Martin-Delgado, Topological quantum distillation, Phys. Rev. Lett. **97**, 180501 (2006).

[25] Eric Dennis, Alexei Kitaev, Andrew Landahl, and John Preskill, Topological quantum memory, J. Math. Phys. **43**, 4452 (2002).

[26] Austin G. Fowler, Minimum weight perfect matching of fault-tolerant topological quantum error correction in average $O(s1)$ parallel time, ArXiv:1307.1740.

[27] Oscar Higgott and Craig Gidney, Sparse blossom: Correcting a million errors per core second with minimum-weight matching, ArXiv:2303.15933.

[28] Yue Wu and Lin Zhong, in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, Bellevue, WA, USA, 2023), Vol. 1, p. 928.

[29] Rui Chao, Michael E. Beverland, Nicolas Delfosse, and Jeongwan Haah, Optimization of the surface code design for Majorana-based qubits, Quantum **4**, 352 (2020).

[30] Craig Gidney, A pair measurement surface code on pentagons, Quantum **7**, 1156 (2023).

[31] C. Monroe, R. Raussendorf, A. Ruthven, K. R. Brown, P. Maunz, L.-M. Duan, and J. Kim, Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects, Phys. Rev. A **89**, 022317 (2014).

[32] Sergey Bravyi, Andrew W. Cross, Jay M. Gambetta, Dmitri Maslov, Patrick Rall, and Theodore J. Yoder, High-threshold and low-overhead fault-tolerant quantum memory, Nature **627**, 778 (2024).

[33] Qian Xu, J. Pablo Bonilla Ataides, Christopher A. Pattison, Nithin Raveendran, Dolev Bluvstein, Jonathan Wurtz, Bane Vasić, Mikhail D. Lukin, Liang Jiang, and Hengyun Zhou, Constant-overhead fault-tolerant quantum computation with reconfigurable atom arrays, Nat. Phys. **20**, 1084 (2024).

[34] Ali Fahimniya, Hossein Dehghani, Kishor Bharti, Sheryl Mathew, Alicia J. Kollár, Alexey V. Gorshkov, and Michael J. Gullans, Fault-tolerant hyperbolic Floquet quantum error correcting codes, ArXiv:2309.10033.

[35] Christophe Vuillot and Nikolas P. Breuckmann, Quantum pin codes, IEEE Trans. Inf. Theory **68**, 5955 (2022).

[36] Nikolas P. Breuckmann and Barbara M. Terhal, Constructions and noise threshold of hyperbolic surface codes, IEEE Trans. Inf. Theory **62**, 3731 (2016).

[37] Nikolas P. Breuckmann, Homological quantum codes beyond the torsic code, PhD thesis (2018).

[38] Nikolas P. Breuckmann, Christophe Vuillot, Earl Campbell, Anirudh Krishna, and Barbara M. Terhal, Hyperbolic and semi-hyperbolic surface codes for quantum storage, Quantum Sci. Technol. **2**, 035007 (2017).

[39] Alexei Kitaev, Anyons in an exactly solved model and beyond, Ann. Phys. (N. Y.) **321**, 2 (2006). January Special Issue.

[40] Margarita Davydova, Nathanan Tantivasadakarn, Shankar Balasubramanian, and David Aasen, Quantum computation from dynamic automorphism codes, Quantum **8**, 1448 (2024).

[41] David Aasen, Jeongwan Haah, Zhi Li, and Roger S. K. Mong, Measurement quantum cellular automata and anomalies in Floquet codes, ArXiv:2304.01277.

[42] Markus S. Kesselring, Julio C. Magdalena de la Fuente, Felix Thomsen, Jens Eisert, Stephen D. Bartlett, and Benjamin J. Brown, Anyon condensation and the color code, PRX Quantum **5**, 010342 (2024).

[43] Aleksander Kubica and Nicolas Delfosse, Efficient color code decoders in $d \geq 2$ dimensions from toric code decoders, Quantum **7**, 929 (2023).

[44] John H. Halton, On the thickness of graphs of given degree, Inf. Sci. (N. Y.) **54**, 219 (1991).

[45] L. J. Stephenson, D. P. Nadlinger, B. C. Nichol, S. An, P. Drmota, T. G. Ballance, K. Thirumalai, J. F. Goodwin, D. M. Lucas, and C. J. Ballance, High-rate, high-fidelity entanglement of qubits across an elementary quantum network, Phys. Rev. Lett. **124**, 110501 (2020).

[46] Juan M. Pino, Jennifer M. Dreiling, Caroline Figgatt, John P. Gaebler, Steven A. Moses, M. S. Allman, C. H. Baldwin, Michael Foss-Feig, D. Hayes, K. Mayer, and et al, Demonstration of the trapped-ion quantum CCD computer architecture, Nature **592**, 209 (2021).

[47] Andreas Reiserer and Gerhard Rempe, Cavity-based quantum networks with single atoms and optical photons, Rev. Mod. Phys. **87**, 1379 (2015).

[48] Dolev Bluvstein, Harry Levine, Giulia Semeghini, Tout T. Wang, Sepehr Ebadi, Marcin Kalinowski, Alexander Keesling, Nishad Maskara, Hannes Pichler, Markus Greiner, *et al.*, A quantum processor based on coherent transport of entangled atom arrays, Nature **604**, 451 (2022).

[49] Rishabh Sahu, William Hease, Alfredo Rueda, Georg Arnold, Liu Qiu, and Johannes M. Fink, Quantum-enabled operation of a microwave-optical interface, Nat. Commun. **13**, 1276 (2022).

[50] Hai-Tao Tu, Kai-Yu Liao, Zuan-Xian Zhang, Xiao-Hong Liu, Shun-Yuan Zheng, Shu-Zhe Yang, Xin-Ding Zhang, Hui Yan, and Shi-Liang Zhu, High-efficiency coherent microwave-to-optics conversion via off-resonant scattering, Nat. Photonics **16**, 291 (2022).

[51] R. D. Delaney, M. D. Urmey, S. Mittal, B. M. Brubaker, J. M. Kindem, P. S. Burns, C. A. Regal, and K. W. Lehnert, Superconducting-qubit readout via low-backaction electro-optic transduction, Nature **606**, 489 (2022).

[52] Na Zhu, Xufeng Zhang, Xu Han, Chang-Ling Zou, Changchun Zhong, Chiao-Hsuan Wang, Liang Jiang, and Hong X. Tang, Waveguide cavity optomagnonics for microwave-to-optics conversion, Optica **7**, 1291 (2020).

[53] Poolad Imany, Zixuan Wang, Ryan A. DeCrescent, Robert C. Boutelle, Corey A. McDonald, Travis Autry, Samuel Berweger, Pavel Kabos, Sae Woo Nam, Richard P. Mirin, and Kevin L. Silverman, Quantum phase modulation with acoustic cavities and quantum dots, Optica **9**, 501 (2022).

[54] Alicia J. Kollár, Mattias Fitzpatrick, and Andrew A. Houck, Hyperbolic lattices in circuit quantum electrodynamics, Nature **571**, 45 (2019).

[55] Craig Gidney, Stim: A fast stabilizer circuit simulator, Quantum **5**, 497 (2021).

[56] Neereja Sundaresan, Theodore J. Yoder, Youngseok Kim, Muyuan Li, Edward H. Chen, Grace Harper, Ted Thorbeck, Andrew W. Cross, Antonio D. Córcoles, and Maika Takita, Demonstrating multi-round subsystem quantum error correction using matching and maximum likelihood decoders, Nat. Commun. **14**, 2852 (2023).

[57] Héctor Bombín, Chris Dawson, Ryan V. Mishmash, Naomi Nickerson, Fernando Pastawski, and Sam Roberts, Logical blocks for fault-tolerant topological quantum computation, PRX Quantum **4**, 020303 (2023).

[58] Nicolas Delfosse and Adam Paetznick, Spacetime codes of Clifford circuits, ArXiv:2304.05943.

[59] Nicolas Delfosse and Naomi H. Nickerson, Almost-linear time decoding algorithm for topological codes, Quantum **5**, 595 (2021).

[60] Shilin Huang, Michael Newman, and Kenneth R. Brown, Fault-tolerant weighted union-find decoding on the toric code, Phys. Rev. A **102**, 012419 (2020).

[61] Austin G. Fowler, Optimal complexity correction of correlated errors in the surface code, ArXiv:1310.0863.

[62] Oscar Higgott, Thomas C. Bohdanowicz, Aleksander Kubica, Steven T. Flammia, and Earl T. Campbell, Improved decoding of circuit noise and fragile boundaries of tailored surface codes, Phys. Rev. X **13**, 031007 (2023).

[63] Jozef Širáň, Triangle group representations and constructions of regular maps, Proc. London Math. Soc. **82**, 513 (2001).

[64] Ethan Fetaya, Bounding the distance of quantum surface codes, J. Math. Phys. **53**, 062202 (2012).

[65] Nicolas Delfosse, in *2013 IEEE International Symposium on Information Theory* (IEEE, Istanbul, Turkey, 2013), p. 917.

[66] Craig Gidney, Michael Newman, and Matt McEwen, Generation/simulation tools for the planar honeycomb code, https://github.com/Strilanc/honeycomb-boundaries, 2022.

[67] Oscar Higgott and Nikolas P. Breuckmann, Ancillary data for hyperbolic and semi-hyperbolic Floquet codes, https://github.com/oscarhiggott/hyperbolic-floquet-data, 2023.

[68] Margarita Davydova, Nathanan Tantivasadakarn, and Shankar Balasubramanian, Floquet codes without parent subsystem codes, PRX Quantum **4**, 020341 (2023).

[69] David Aasen, Zhenghan Wang, and Matthew B. Hastings, Adiabatic paths of hamiltonians, symmetries of topological order, and automorphism codes, Phys. Rev. B **106**, 085122 (2022).

[70] Zhehao Zhang, David Aasen, and Sagar Vijay, X-cube Floquet code: A dynamical quantum error correcting code with a subextensive number of logical qubits, Phys. Rev. B **108**, 205116 (2023).

[71] Andreas Bauer, Topological error correcting processes from fixed-point path integrals, Quantum **8**, 1288 (2024).

[72] Joseph Sullivan, Rui Wen, and Andrew C. Potter, Floquet codes and phases in twist-defect networks, Phys. Rev. B **108**, 195134 (2023).

[73] Arpit Dua, Nathanan Tantivasadakarn, Joseph Sullivan, and Tyler D. Ellison, Engineering 3d floquet codes by rewinding, PRX Quantum **5**, 020305 (2024).

[74] Austin G. Fowler, David S. Wang, Charles D. Hill, Thaddeus D. Ladd, Rodney Van Meter, and Lloyd C. L. Hollenberg, Surface code quantum communication, Phys. Rev. Lett. **104**, 180503 (2010).

[75] Naomi H. Nickerson, Ying Li, and Simon C. Benjamin, Topological quantum computing with a very noisy network and local error rates approaching one percent, Nat. Commun. **4**, 1756 (2013).

[76] Naomi H. Nickerson, Joseph F. Fitzsimons, and Simon C. Benjamin, Freely scalable quantum technologies using cells of 5-to-50 qubits with very lossy and noisy photonic links, Phys. Rev. X **4**, 041041 (2014).

[77] Joshua Ramette, Josiah Sinclair, Nikolas P. Breuckmann, and Vladan Vuletić, Fault-tolerant connection of error-corrected qubits with noisy links, npj Quantum Inf. **10**, 58 (2024).

[78] Armanda O. Quintavalle, Paul Webster, and Michael Vasmer, Partitioning qubits in hypergraph product codes to implement logical gates, Quantum **7**, 1153 (2023).

[79] Mark A. Webster, Armanda O. Quintavalle, and Stephen D. Bartlett, Transversal diagonal logical operators for stabiliser codes, New J. Phys. **25**, 103018 (2023).

[80] Nikolas P. Breuckmann and Simon Burton, Fold-transversal Clifford gates for quantum codes, Quantum **8**, 1372 (2024).