Fault-tolerant hyperbolic Floquet quantum error correcting codes

Ali Fahimniya[1,2], Hossein Dehghani[1,2], Kishor Bharti[1,2,3], Sheryl Mathew[1], Alicia J. Kollár[2], Alexey V. Gorshkov[1,2], and Michael J. Gullans[1]

[1] Joint Center for Quantum Information and Computer Science, NIST/University of Maryland, College Park, Maryland 20742, USA

[2] Joint Quantum Institute, NIST/University of Maryland, College Park, Maryland 20742, USA

[3] Institute of High Performance Computing (IHPC), Agency for Science, Technology and Research (A*STAR), 1 Fusionopolis Way, # 16-16 Connexis, Singapore 138632, Republic of Singapore

arXiv:2309.10033v4 [quant-ph] 25 Aug 2025

A central goal in quantum error correction is to reduce the overhead of fault-tolerant quantum computing by increasing noise thresholds and reducing the number of physical qubits required to sustain a logical qubit. We introduce a potential path towards this goal based on a family of dynamically generated quantum error correcting codes that we call "hyperbolic Floquet codes." These codes are defined by a specific sequence of non-commuting two-body measurements arranged periodically in time that stabilize a topological code on a hyperbolic manifold with negative curvature. We focus on a family of lattices for n qubits that, according to our prescription that defines the code, provably achieve a finite encoding rate $(1/8 + 2/n)$ while still requiring only two-body measurements. Similar to hyperbolic surface codes, the distance of the code at each time-step scales at most logarithmically in n. The family of lattices we choose indicates that this scaling is achievable in practice. We develop and benchmark an efficient matching-based decoder that provides evidence of a threshold near $0.1\,\%$ in a phenomenological noise model and $0.25\,\%$ in an entangling measurements noise model. Utilizing weight-two check operators and a qubit connectivity of 3, one of our hyperbolic Floquet codes uses 400 physical qubits to encode 52 logical qubits with a code distance of 8, i.e., it is a $[[400, 52, 8]]$ code. At small error rates, comparable logical error suppression to this code requires 5x as many physical qubits (1924) when using the honeycomb Floquet code with the same noise model and decoder.

Ali Fahimniya: fahim@umd.edu

1 Introduction

Large-scale quantum computing requires quantum error correction and fault-tolerance protocols to control the growth of errors. The surface code [1, 2, 3] is a promising route towards fault-tolerant quantum computing because of its high tolerance for noise and geometrically local implementation. These features have led to considerable interest in realizing small instances of the surface code, as demonstrated in recent experiments [4, 5, 6, 7, 8, 9, 10, 11]. In the conventional surface code, both the parity-check weight and the qubit degree are relatively low, specifically, each is only four. This means that each qubit is involved in just four parity checks, which are operations that help identify and correct errors. However, a drawback of the surface code is that it requires a large number of physical qubits to represent each logical qubit [12]. To improve on the performance of the surface code in general fault-tolerant protocols, one can attempt to optimize the syndrome extraction circuit by using weight-two check operators [13, 14, 15], tailor the error correction scheme to specific noise models [16, 17, 18, 19], or reduce the number of physical qubits required to achieve a certain number of logical qubits with a given code distance [20].

Current schemes either support weight-two check operators or maintain a constant ratio of

logical to physical qubits, but not both. For instance, recent research on quantum Low-Density Parity-Check (LDPC) codes [12] suggests that these codes can encode 12 logical qubits using 288 physical qubits. This compares favorably against surface codes which would require 4000 physical qubits to encode 12 logical qubits for similar error suppression. In an LDPC-type quantum error-correcting code, each check operator acts on only a few qubits, and each qubit is part of only a few checks [20, 21, 22, 23, 24, 12, 25]. However, these particular quantum LDPC codes [12] necessitate the use of weight-six checks, rendering their implementation challenging with current devices. Hyperbolic quantum codes [26, 27, 28, 29] that employ tessellations of closed hyperbolic surfaces have been proposed as a solution for achieving finite encoding rate with weight-three check operators. Recently, there has been considerable attention directed toward a new class of codes known as Floquet codes [13, 30, 31, 32, 33, 34, 35, 36, 37]. These codes feature weight-two check operators but come with the limitation of a vanishing encoding rate; thus, raising the question of whether it is possible to achieve a finite encoding rate using weight-two check operators instead.

In this work, we introduce a new family of codes, termed as hyperbolic Floquet codes, with a finite encoding rate of $1/8 + 2/n$ for n physical qubits and weight-two check operators. For example, with 400 physical qubits, our scheme can encode 52 logical qubits. See Table 1 for a summary of some of our hyperbolic Floquet quantum error correcting codes. The hyperbolic Floquet codes inherit some of the best features from quantum hyperbolic codes [26, 27, 28, 29] and Floquet codes [13], although they give up on geometric locality. Similar to Floquet codes, our check operators have a weight of two, while the encoding rate is finite, akin to that of quantum hyperbolic codes. The implementation of our code demands a reduced qubit connectivity of 3, in contrast to the recently introduced quantum LDPC codes [12] which may require a connectivity of 6.

Our approach excels in terms of the weight of the check operators, qubit connectivity, and encoding rate. However, a trade-off is that code distance (d) can scale at most logarithmically in the number of physical qubits. We find a family of hyperbolic lattices of increasing size whose associated hyperbolic Floquet codes achieve a distance scaling consistent with this expectation. We develop an efficient decoder based on matching of syndrome pairs similar to the surface code. Employing two different noise models, we see evidence for thresholds around 0.1 % and 0.25 %. The finite encoding rate implies that a total logical error suppression can be reached with an asymptotically vanishing fraction of the physical qubits that would be required using zero-rate topological codes. To further substantiate this scaling argument, we provide a detailed comparison of the logical failure rate of our codes against many independent copies of the honeycomb Floquet code under the same noise models and decoder.

The architecture of hyperbolic Floquet codes is elaborated upon in Section 2, where we explore various hyperbolic lattices, measurement protocols and logical operators. The properties related to error detection and correction are described in Section 3. In Section 4, we present numerical results of logical error rate suppression and evidence for a threshold, as well as compare hyperbolic Floquet codes in performance to honeycomb Floquet codes within the same noise model and decoder. Finally, we conclude with discussion and several open problems in Section 5.

2 The Hyperbolic Floquet Code

Lattice and Measurement Protocol. Hyperbolic Floquet codes utilize regular tessellations of negatively curved planes, employing even-sided polygons as building blocks with a degree of three for all vertices. The physical qubits are located at the vertices of the lattice. This code relies on a three-coloring of the polygons, and thus, it is required that they are even-sided. Figure 1 depicts a face three-colored lattice of octagons that is the basis for a 64-qubit hyperbolic Floquet code. This lattice is of the form $\{8, 3\}$ using Schäfli symbols [38], meaning that 3 octagons (8-gons) meet at each vertex. We refer to this lattice as the *octagonal lattice*, and employ it in the development of hyperbolic Floquet codes. Hyperbolic Floquet codes can also be derived from lattices with decagons or higher even-sided polygons, though such codes fall outside of the scope of this paper's in-depth analysis.

While the octagonal lattice is regular and all

vertices are equivalent, the projection of this lattice from the negatively-curved hyperbolic plane onto a flat plane, as in Fig. 1, results in distortions in the shapes of the displayed octagons. We employ the polar coordinates of the vertices for this projection. That is, each vertex's projected coordinates are obtained based on its distance from a reference vertex using the hyperbolic metric and its orientation with respect to a reference direction. The depicted lattice is a closed lattice, with complex periodic boundary conditions, tessellating a genus-5 manifold. The periodic boundary conditions identify each edge that does not seem to have two adjacent octagons with another such edge. Four of these boundary condition identifications are labeled with the letters a, b, c, and d. Edges with the same letter are identified as equivalent, thus representing the same edge. Similarly, the four vertices at the ends of the paired edges with matching labels are identified and represent only two distinct vertices.

The edge colors of the octagonal lattice in Fig. 1 are derived from the face colors, such that each edge inherits its color from the two octagonal faces that it connects and would extend into if elongated. In the Floquet code, these three-colored edges constitute the repetitive rounds of two-body check measurements of the physical qubits of the edges. At rounds $r = 3n$ ($r = 3n + 1$, $r = 3n + 2$) with nonnegative integers n, all red (green, blue) checks are measured, as illustrated in Fig. 2(a). Specifically, red, green, and blue checks correspond to measurements of XX, YY, and ZZ Pauli operators on the pairs of physical qubits at the ends of the respective edges. Although any two such measurement checks do not necessarily commute, all the checks within a single round do commute and can be considered to be measured simultaneously.

It is noteworthy that the original honeycomb Floquet code described by Hastings and Haah [13] requires a second edge labeling based on edge orientations, which dictates the type of two-body measurement. However, as later indicated by Gidney et al. [30], this second labeling is redundant, as the edge colors alone can determine the measurement type, yielding the same code. These two code definitions can be mapped onto one another through local Bloch rotations of the physical qubits. In this work, we adopt the version requiring solely the edge three-coloring to define hyperbolic Floquet codes. This approach is advantageous because edge orientation is inherently ambiguous in hyperbolic space.

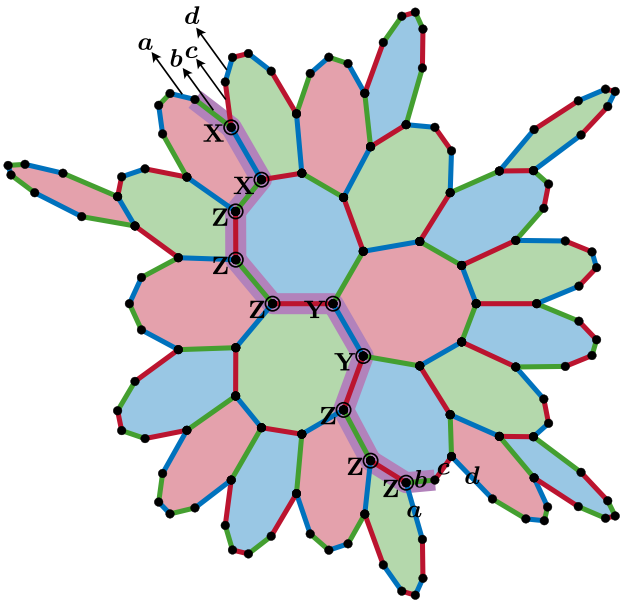


Figure 1: A hyperbolic Floquet code with 64 physical qubits. The code is based on an octagonal ({8, 3}) lattice, which is displayed using polar coordinates of its vertices. The 64 physical qubits are located at the vertices of the lattice. Identification of the edges at the boundary makes the lattice a periodic tessellation of a genus-5 manifold. Four such periodic boundary condition identifications are marked by letters a-d. The octagonal faces are three-colored such that the adjacent faces have different colors. Each edge inherits the color from the two faces it would enter if elongated. Red, green, and blue edges correspond to XX, YY, and ZZ measurement checks, respectively. The green edges at the ends of the highlighted purple path are identified as a periodic boundary condition, marked with letter b. The purple path therefore is a closed path consisting of 10 unique vertices, marked by circles. This closed path hosts a weight-ten noncontractible loop operator, Pauli operators of which are denoted next to its vertices.

Instantaneous Stabilizer Group. In this section, we assume errorless evolution of the quantum state of the physical qubits. We further consider that this state starts as a maximally mixed state prior to the round $r = 0$ of measurements. Then, after each measurement round, we examine the group of all Pauli operators that have an eigenvalue $+1$ on the current state of the physical qubits, i.e., stabilize the current state. This group is termed the *instantaneous stabilizer group* (ISG), and before any measurements, con-

sists of solely the identity operator. The check measurements, however, change the structure of the ISG. Figure 2(c) shows the evolution of the ISG through measurement rounds.

After rounds $r = 3n$ ($r = 3n + 1$, $r = 3n + 2$), each XX check (YY check, ZZ check) becomes a generator of the ISG, up to a sign determined by the respective measurement outcome (i.e., if the measurement result of a check is -1, the negative of that check joins the ISG). For every one of these two-body generators of the ISG, two checks in the subsequent measurement round anticommute with it. As a result, these two-body instantaneous stabilizer generators no longer stabilize the state following the next round. For example, consider the four red (i.e., XX) edges around the blue face in Fig. 2(b). After round $r = 3n$, the two-body checks of these edges, up to their measurement results' signs, are members of the ISG. The upcoming measurement round, $r = 3n + 1$, is green (i.e., YY) checks. Each red check anticommutes with two green checks from of the blue face, and hence, will be removed from the ISG upon measurement of any of those green checks. After round $r = 3n + 1$, each of these green checks, up to its measurement outcome sign, will be in the ISG.

Although the two-body checks are removed from the ISG following the next measurement round, certain products of them commute with the upcoming measurement checks, and therefore, stay in the ISG. In the example of the blue face and red checks in Fig. 2(b), the product of all four red checks commutes with any green check, ensuring its membership in the ISG after round $r = 3n + 1$. The product of these four red checks can be multiplied with the product of the four green checks of that blue face, deriving a weight-eight plaquette operator. This operator, after round $r = 3n + 1$, is part of the ISG. Since it is equal to a product of X and Y Paulis for each qubit, it is equivalent to a product of Z Paulis (up to a phase). In a similar vein, plaquette operators for red and green faces are products of X and Y Paulis, respectively. The plaquette operators are depicted in Fig. 2(b).

Each plaquette operator [Fig. 2(b)] shares the same Pauli operator as the two-body checks intersecting it at a single vertex. Conversely, it has a different Pauli operator than the two-body checks intersecting it at two vertices. Consequently, these plaquette operators commute with all check operators. When measured (during rounds $r = 2$, $r = 3$, and $r = 4$ corresponding to blue, red, and green faces), they will join the ISG, up to a sign derived from the measurements. Beyond round $r = 4$, the ISG mod phases reaches a periodic steady state with the logical qubits in a mixed state (i.e., stable against the noncommuting measurements), consistently generated by all plaquette operators and the most recent measurement checks, each acquiring a sign from the measurement outcomes.

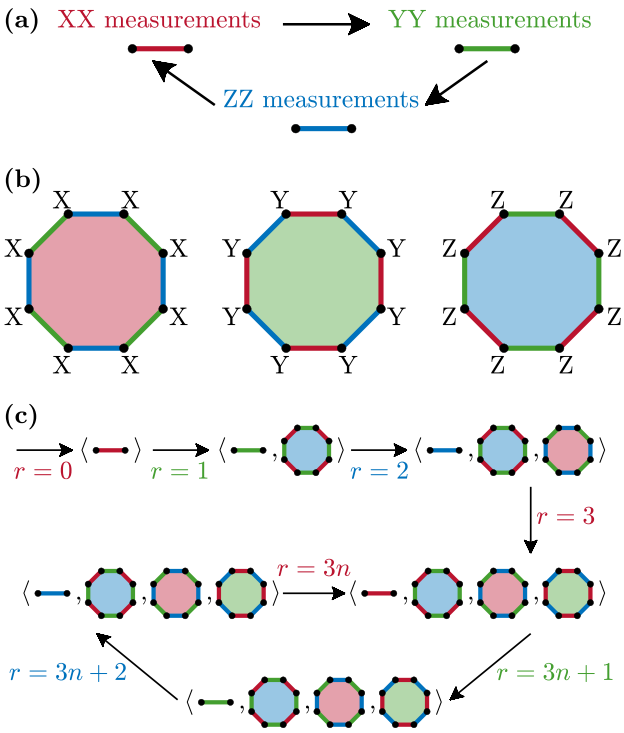


Figure 2: Measurement protocol and plaquette operators of hyperbolic Floquet codes. (a) One cycle of the Floquet code's measurement protocol. Each cycle consists of three measurement rounds, each round carried out on XX, YY, and ZZ checks, corresponding to red, green, and blue edges in Fig. 1, respectively. In each measurement, the weight-two Pauli operator of the two qubits at the ends of that edge are measured. (b) Plaquette operators that are products of checks along the edges of each octagon, for red, green, and blue plaquettes. They commute with all check operators on the lattice, and their value can be inferred using the measurement outcomes of the check operators. (c) The instantaneous stabilizer group (ISG) after each measurement round expressed as generated by its generators after each measurement round. The generators can be check operators (panel a) or plaquette operators (panel b). Each check or plaquette operator represents all similar operators with the same color up to a sign.

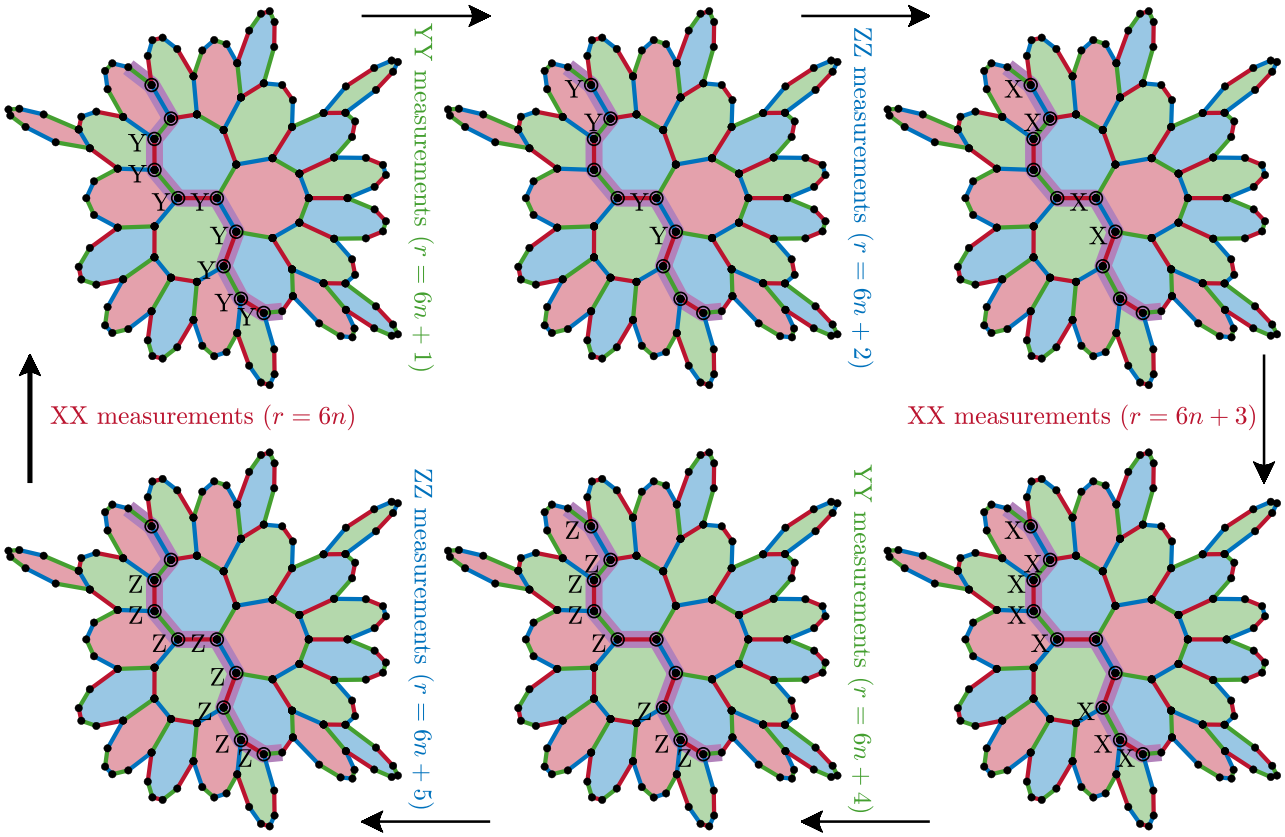


Figure 3: Evolution of a logical operator in the 64-qubit hyperbolic Floquet code through measurement rounds. The logical operator is contained within the noncontractible loop operator (see Fig. 1) of a closed path (highlighted in purple). The nonidentity Pauli operators of the logical operator are denoted next to the physical qubits. The logical operator commutes with the plaquette operators, the most recent measurement checks, and the upcoming measurement checks. After each round of measurements, the logical operator's expression is updated by being multiplied with the measured checks along the purple path. The update procedure results in changing a type one logical operator to a type two, and vice versa. Consequently, a periodicity of six for the logical operator's expression emerges. The weight of the logical operator varies through rounds, with a minimum value of four.

Logical Operators. Logical operators are the operators that commute with the ISG but are not part of it. Commuting with any two consecutive instances of the ISG is required for the logical information to not be destroyed by the measurement rounds [39]. However, operators that commute with the ISG but are always a part of it, i.e., the plaquette operators and their products, have determined values and cannot be used for restoring information about the logical qubits. The logical operators of the hyperbolic Floquet code are contained within noncontractible closed paths of the hyperbolic lattice. These are the paths that, utilizing the periodic boundary conditions, wrap around the lattice. One closed path of the hyperbolic lattice in Fig. 1 is highlighted in purple.

Just as the product of two-body checks around an octagon commutes with all two-body checks, the product along any loop does the same. We will call these *loop operators*. The contractible loop operators can be expressed as the product of the plaquette operators within them, and they are part of the ISG beyond round $r = 4$. On the other hand, the noncontractible loop operators will never integrate into the ISG. The closed path of one such loop is highlighted with purple in Fig. 1. The displayed periodic boundary condition identifications (edges labeled with letters a to d) serve as a reference to confirm that the purple path is indeed closed. The lattice's periodic boundary conditions identify the two green edges at the ends of the portrayed path, making the two vertices at each end non-unique. The distinct vertices of the weight-ten loop operator are marked with a circle, and the adjacent Pauli op-

erators represent the components of the loop operator. It is noteworthy that the Pauli operator of each physical qubit is determined by its outgoing edge that is not part of the loop; specifically, X, Y, or Z Paulis for red, green, or blue outgoing edges, respectively. Noncontractible loop operators comprise the Floquet code's logical operators. As we will discuss below, the number of noncontractible loop operators is twice the genus of the manifold.

Each noncontractible loop operator can be decomposed into the product of two operators, where both individually commute with the elements of the ISG. This decomposition, akin to the ISG, depends on the most recent measurement round. Assume that measurements of the red check operators (XX measurements) have just been carried out (round $r = 3n$). The next measurement round is the green checks (YY measurements). The two operators that commute with the ISG elements and make a noncontractible loop operator are the product of X Paulis on the vertices of the green edges along its closed path (bottom right panel in Fig. 3) and the product of Y Paulis on the vertices of the red edges along it (top left panel in Fig. 3). These two constructions are generalizable to any measurement round as *the most recent measurement Paulis on the edges of the upcoming round* and *the upcoming measurement Paulis on the edges of the most recent round*, and hereinafter will be referred to as the *type one* and *type two* logical operators.

Given that, after XX measurements, the type one logical operator has only X Paulis and the type two has two Paulis on each XX check, they commute with the two-body XX checks in the ISG. They also commute with plaquette operators within the ISG. Taking the type one operator as an example, each red and blue plaquette shares an integer number of green edges with the closed path. This means it has an even number of non-identity Paulis in common with the first operator. Moreover, since the path is closed and always leaves any octagon that it enters, an even number of its green edges are incident at any given green octagon. Therefore, there are again an even number of mutual non-identity Paulis between the operator and any green plaquette operator. Similar analysis can be carried out for the type two operator, establishing that both operators commute with the ISG. Further, these operators are independent operators and one of them cannot be obtained by multiplying the other one with some elements of the ISG. These are the two logical operators defined on the noncontractible loop.

So far, the constructed logical operators have been found to commute with the ISG after XX measurements. Notably, this construction ensures that they also commute with the upcoming measurements, specifically YY checks on green edges. To elaborate, the type one logical operator features two X Paulis on any green edge along its path and the type two operator is made of Y Paulis. Therefore, YY checks can be measured without destroying these logical operators. However, after YY measurements, some of the upcoming ZZ checks anticommute with the current construction of these logical operators and their measurement is going to affect the logical operators. Nonetheless, before ZZ measurements, we can update the expression of the logical operators by multiplying them with the measured YY checks along their noncontractible loops. The measurement outcomes of these YY checks must be factored into the sign of the updated logical operators.

Consider the type two logical operator (top left panel of Fig. 3). When we multiply this operator by YY checks on the closed path, the Y Paulis shift to the blue edges (top middle panel of Fig. 3). Note that this is a type one logical operator after YY (green) and prior to ZZ (blue) measurements. Following the next measurement round (ZZ measurements), we multiply this operator with the measured ZZ checks along the path, making it comprised of X Paulis on the blueedges (top right panel of Fig. 3). This is in turn a type two logical operator after ZZ and prior to XX measurements. As seen in Fig. 3, the logical operator switches its type with each measurement round, and reverts to its original expression after six rounds.

In Ref. [13], the noncontractible loop operators are called the inner logical operators, while the type one and type two logical operators are equivalent to the electric and magnetic outer logical operators defined therein. As the noncontractible loop operators are equal to the product of two logical operators, their existence does not add to the number of independent logical opera-

tors, and hence, the number of logical qubits of a hyperbolic Floquet code. In order to emphasize this, we do not use the term logical operators in referring to them.

Code Family. Now we focus on finding the lattices that give rise to the particular codes and calculate their code parameters. Each code is based on a periodic octagonal lattice. To identify lattices, we turn to the database of symmetric connected graphs of degree three up to 10,000 vertices, provided in [40][1]. Within this database, the graphs with a *girth* of eight qualify as periodic octagonal lattices. These lattices are symmetric (arc-transitive) graphs: for any pair of edges, there exists an automorphism of the graph that maps one edge to the other.

The number of vertices of each lattice determines the number of the physical qubits, n, for the corresponding code. The number of logical qubits, k, on the other hand, is determined by the genus of the manifold that the periodic lattice tessellates. In order to find the relation between the genus of this manifold and n, we need to look at the crystallography structure of the octagonal lattice [41].

An octagonal lattice has a Bravais unit cell comprising 16 vertices, which tessellates the surface of a genus-2 manifold (see Fig. 12 in Ref. [41]). Every additional set of 16 vertices in a given octagonal lattice corresponds to one unit cell and cover an area equal to -4π (in units of negative curvature) on the hyperbolic plane. Thus, the genus of the manifold covered by the entire lattice is given by:

$$g = \frac{n}{16} + 1. \tag{1}$$

Having a genus that grows with the number of physical qubits, hyperbolic error correcting codes exhibit a finite encoding rate. This stems from the fact that a manifold's number of independent noncontractible closed paths equals twice its genus. Two closed paths are dependent if the multiplication of one by a set of plaquettes yields the other. This relation divides the noncontractible loops into equivalence classes, where any pair of loops in the same class are dependent.

[1]Reference [40] includes methodology and graphs up to 768 vertices. Graphs with up to 10,000 vertices can be found here.

Code	Unit Cells #	Genus	[[n, k, d]]
H16	1	2	[[16, 4, 2]]
H64	4	5	[[64, 10, 4]]
H144	9	10	[[144, 20, 6]]
H400	25	26	[[400, 52, 8]]
H2160	135	136	[[2160, 272, 10]]

Table 1: Family of hyperbolic octagonal Floquet codes with the fewest number of physical qubits for each code distance. Each code is denoted by a prefix 'H' followed by the number of its physical qubits. The layout of the physical qubits corresponds to a tessellation of a negatively curved manifold with one Bravais unit cell per each 16 physical qubits. The number of handles (genus) of this manifold is one more than the number of unit cells [Eq. (1)]. The number of the logical qubits, k, is twice the genus [Eq. (2)], and the code distance, d, is determined by finding the logical operator with the lowest minimum weight.

The number of these classes is $2g$. Furthermore, as discussed above, each loop can be split into two distinct logical operators. Hence, the total number of logical qubits becomes:

$$k = 2g = \frac{n}{8} + 2, \tag{2}$$

and the resulting code's encoding rate is:

$$\frac{k}{n} = \frac{1}{8} + \frac{2}{n}, \tag{3}$$

which is always greater than $1/8$ and approaches $1/8$ as $n \to \infty$.

Although the choice of the octagonal lattices ensures local face three-coloring, not every periodic octagonal lattice is face-three-colorable. This limitation emerges from potential inconsistencies along noncontractible closed paths. Notably, larger lattices have more independent noncontractible closed paths, complicating the search for face-three-colorable large lattices among generic octagonal lattices. However, this is not an issue for the lattices that are symmetric (arc-transitive) graphs. In such lattices, any two edges map into each other under an automorphism of the lattice. As a result, classes of equivalent (nonindependent) noncontractible loops also exhibit mutual interchangeability. Therefore, if a lattice's faces can be three-colored along a single noncontractible loop, the three-coloring can be extended to the entire lattice.

The last relevant code parameter is the code distance. During the update process in Floquet codes (Fig. 3), each logical operator's weight

(number of nonidentity Paulis) changes through six rounds, assuming its minimum value in some of the rounds. For instance, the logical operator shown in Fig. 3 has a minimum weight of four. The code distance, d, is the lowest minimum weight found among all logical operators. Note that with the aforementioned constructions of the logical operators, each logical operator's minimum weight, and consequently the code distance, are always even numbers for a Floquet code.

We assess the periodic octagonal lattices from [40]. Each lattice in this list that is face-three-colorable is the basis for a hyperbolic Floquet code. As detailed above, the corresponding code parameters, $[[n, k, d]]$, can be calculated. Among the codes with a given distance, d, the one requiring the fewest physical qubits, n, is the most efficient, i.e., it uses the least amount of resources to achieve the same distance. Table 1 lists the code parameters and lattice properties for the most efficient hyperbolic Floquet codes for even distances from 2 to 10 [2]. The maximum code distance achieved based on octagonal lattices in [40] is 10. We designate these distance-maximizing codes using the prefix 'H' followed by their respective number of physical qubits.

3 Error Models and Decoding Method

Error Models. We employ two error models. A *phenomenological* model comprising unbiased depolarization errors affecting individual physical qubits and measurement errors associated with the weight-two measurements of the checks, and an *entangling measurements* model which imitates circuit-level noise for future platforms with native two-qubit measurements, such as Majorana fermions [42]. Considering other noise models and optimizing the code/decoder to the noise is an interesting avenue for future work.

The strength of the error in both models is set by a single parameter, p, that is a measure of the physical error rate per measurement round. In the *phenomenological* error model, an unbiased depolarization channel of strength p applies an X, Y, or Z Pauli operator to each physical qubit between consecutive rounds, each with an equal probability of $p/3$. Simultaneously, a measurement error channel of strength p causes an erroneous outcome for each measurement with a probability p. In the *entangling measurements* error model [30], each check measurement is accompanied by an error channel with probability p. In the case of an error, a non-identity weight-two Pauli is applied to the qubits of the measured check, and in half of the cases the measurement outcome is erroneous too.

Error detection properties of the hyperbolic Floquet codes are very similar to the honeycomb Floquet codes. For readability, this Section contains a summary of the essentials of our approach, and the full technical description is given in Appendix A.

Inferred values of the plaquette operators, called *plaquette syndromes* or simply *syndromes*, construct a $(2+1)D$ space-time lattice that we refer to as the *syndrome lattice*. In the absence of errors, consecutive inferences of each plaquette syndrome yield the same value. Conversely, an updated plaquette syndrome that differs from its previous value implies the presence of an error (or errors) affecting that syndrome. We call these events *detection events*, and they are used in identifying the occurred errors.

Each single-qubit Pauli error, depending on the type of Pauli operator and the most recent round of measurements, yields either two or four detection events. We call Pauli error with two detection events *simple Pauli errors* and those with four detection events *compound Pauli errors*. Two-qubit Pauli errors give rise to the detection events corresponding to their constituent single-qubit Paulis, and measurement errors are associated with four detection events. (see Fig. 8 and Fig. 9.)

To simplify our decoder, we leverage the fact that all errors can be expressed as equivalent combinations of simple Pauli errors resulting in the same detection events and impacting the logical operators identically. Specifically, our decoder operates under the assumption that only simple Pauli errors have occurred. It then identifies a set of such errors associated with the observed detection events. Given that each simple error is linked to two detection events, forming an error edge in the syndrome lattice, the decoder

[2] The colored adjacency matrices of the hyperbolic and honeycomb lattices used for Floquet codes in this paper, as well as Python scripts for evaluating these codes are available online at https://github.com/fahimniya/HyperbolicFloquetQECC.

performs this by pairing (with a path of possible error edges) each detection event in the syndrome lattice with another detection event. To achieve this, we use the Minimum-Weight Perfect Matching (MWPM) decoder from the PyMatching package [43].

Misidentifying more complex errors as combinations of simple ones can have a negative impact on the performance of the decoder. However, being able to utilize MWPM speeds up the decoding process significantly. In this paper, we do not optimize the decoder substantially and instead focus on establishing the existence of a threshold. Since our decoder is easily adaptable to honeycomb Floquet codes, we can also perform a direct comparison between hyperbolic and honeycomb Floquet codes.

4 Fault-Tolerant Memory Threshold

Logical Error Rates. With the codes, error model, and the decoder defined in Secs. 2 and 3, we can proceed to analyze how hyperbolic Floquet codes perform in storing logical qubits while being affected by the introduced error channels impacting the physical qubits.

For each hyperbolic Floquet code listed in Table 1, this analysis involves sampling instances of decoding scenarios. In each decoding scenario, depolarization and measurements error channels with strength p (as defined in Sec. 3) are applied during $3d$ rounds of check measurements, corresponding to d Floquet periods, where d is the code distance. Repetition of measurements is needed to preserve the logical memory for a long time in the presence of measurement and decoherence processes. The threshold for further rounds of syndrome extraction (up to times polynomial in d) should be identical [3]. At lower depth, the threshold may shift to higher values as it becomes closer to the code-capacity threshold.

We denote individual Pauli or measurement errors as E_i's. As explained in Sec. 3 and Appendix A, compound single-qubit Pauli, two-qubit Pauli, and measurement errors can be decomposed as multiple simple Pauli errors. Thus, each E_i is equivalent to a subset of all possible error edges in the syndrome lattice, based on its simple Pauli decomposition. This subset of error edges can be represented by a vector in the vector space of all possible error edges in the entire syndrome lattice, with each error edge corresponding to a simple Pauli error [Fig. 8(b) and (c)]. This vector space operates over a $\mathbb{Z}_2 = \{0, 1\}$ field, where addition is carried out modulo 2. The $\mathbb{Z}_2$ vector of each subset of error edges contains a 1 for the edges within that subset and a 0 for every other edge. Then, the total error in the syndrome lattice, E, comprised of individual errors is equal to:

$$E = \sum_i E_i, \tag{4}$$

where the summation is well-defined as a modulo-2 addition of the $\mathbb{Z}_2$ vectors representing individual errors (E_i's).

The vertices at the two ends of an error edge in the $(2+1)D$ syndrome lattice represent the detection events associated with that simple Pauli error. If that error occurs, the plaquette syndromes of these vertices change sign during the round corresponding to the vertex. Consequently, a vertex that is shared by an odd (even) number of error edges within a subset of all error edges will (will not) exhibit a detection event should this subset of errors occur. The subset of syndrome lattice vertices that are shared by an odd number of error edges within a set of error edges is termed the *endpoints* of that set. The endpoints of the error edges corresponding to total error, E, indicate the detection events linked to these errors. We denote these detection events as D.

The decoder's role is to find a set of errors, denoted as C, that would generate the same detection events, D, in the syndrome lattice. This set of errors is the proposed *correction* by the decoder. Our MWPM decoder carries this out by finding a set of error edges with the minimum number of edges that has D as their endpoints. The correction, C, is then the set of simple Pauli errors corresponding to these error edges. The combined effect of the occurred errors and the decoder's correction is linked to the error edges corresponding to $E + C$. Since both the error edges of E and C share D as their endpoints, the error edges corresponding to $E + C$ have no endpoints and form closed paths in the syndrome lattice. Given the periodicity of the syndrome lattice in its hyperbolic $2D$ plane, these $(2+1)D$ loops can be either contractible or noncontractible. For

[3] Appendix B shows some results where we substitute these $3d$ rounds with d to $4d$ rounds to show that this choice does not affect the results reported here.

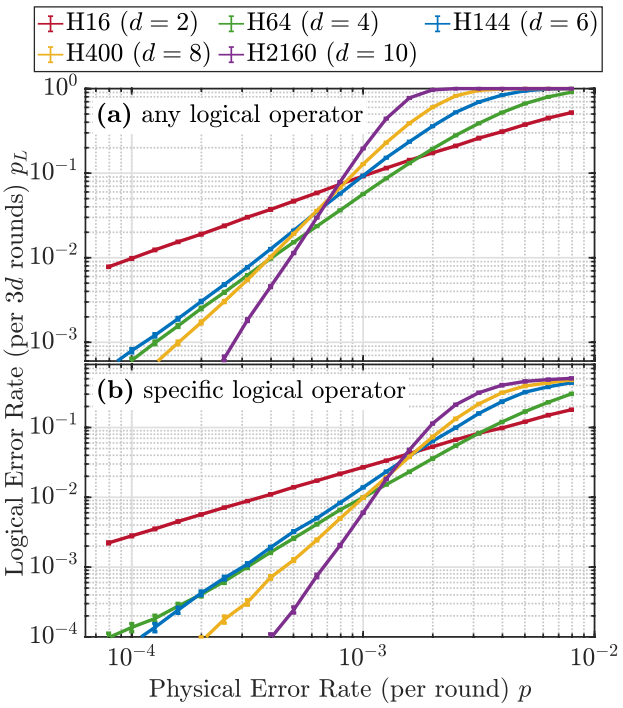


Figure 4: Logical error rates for the family of hyperbolic Floquet codes listed in Table 1 under the phenomenological error model. (a) The *any logical operator* error rate. An error in any of the $2k$ logical operators of each code is considered a logical error for this rate. (b) The *specific logical operator* error rate. For this rate, failure or success of the decoder is defined based on having an error or not impacting a specific logical operator within $2k$ logical operators of each code. Since hyperbolic Floquet codes are based on symmetric graphs, the choice of the specific logical operator does not matter for obtaining this plot.

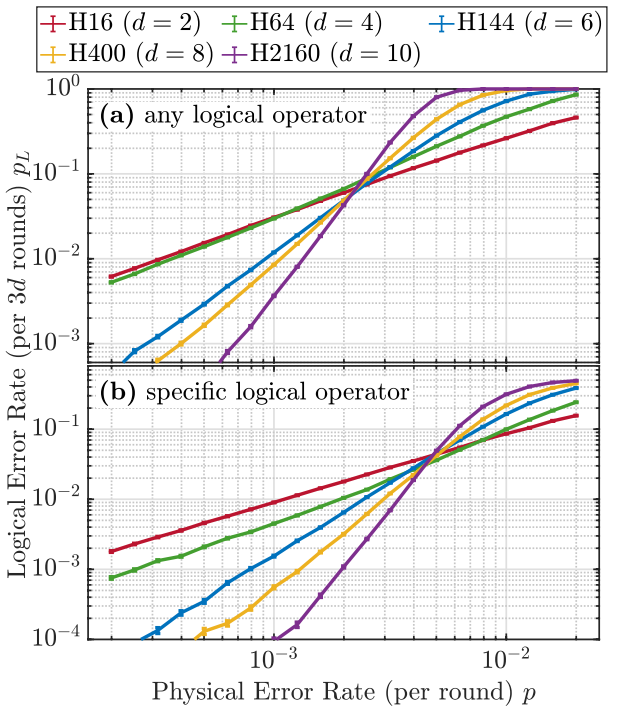


Figure 5: Logical error rates for the family of hyperbolic Floquet codes listed in Table 1 under the entangling measurements error model. (a) The *any logical operator* error rate. (b) The *specific logical operator* error rate. (See the caption of Fig. 4 for additional information about the two logical error rates.)

each noncontractible loop within the hyperbolic manifold, there exists at least one logical operator that anticommutes with the set of errors of that loop. If $E + C$ comprises an odd number of loops that anticommute with a specific logical operator, the decoding process has resulted in a logical error impacting that logical operator.

Figures 4 and 5 present the MWPM decoder's performance in correcting the physical errors for the hyperbolic Floquet codes listed in Table 1 [2], under the *phenomenological* and *entangling measurements* error models, respectively. The numerically calculated logical error rates have been determined by running the decoder on the syndrome lattice after applying the physical errors over $3d$ rounds of check measurements, with d the code distance[3]. Then we can examine whether each logical operator has been impacted by an error by counting the number of loops in $E+C$ that anticommute with it. In Fig. 4(a) and Fig. 5(a), an error impacting any of the $2k$ logical operators is considered a failure in a decoding scenario. Conversely, in Fig. 4(b) and Fig. 5(b), we focus on solely a specific logical operator and assess decoder's performance concerning it. Since these codes are based on symmetric graphs (as discussed in Sec. 2), the resulting logical error rate is independent of the specific equivalence class of the chosen logical operator. Hence, Fig. 4(b) and Fig. 5(b) would be the same, up to numerical variations, if any other specific logical operator were chosen.

At high physical error rates, both defined logical error rates saturate at a maximum value. This saturation value is 0.5 for the *specific logical operator error rate*. The reason behind this is that the combined effect of too many errors on a logical operator is equal to an unbiased coin toss: in half of scenarios, there may be an even number of noncontractible loops in $E + C$ that cross this logical operator once, hence canceling each other's effects and leaving the logical operator unaffected. For the *any logical operator error rate*, however, the probability that $E + C$ has no

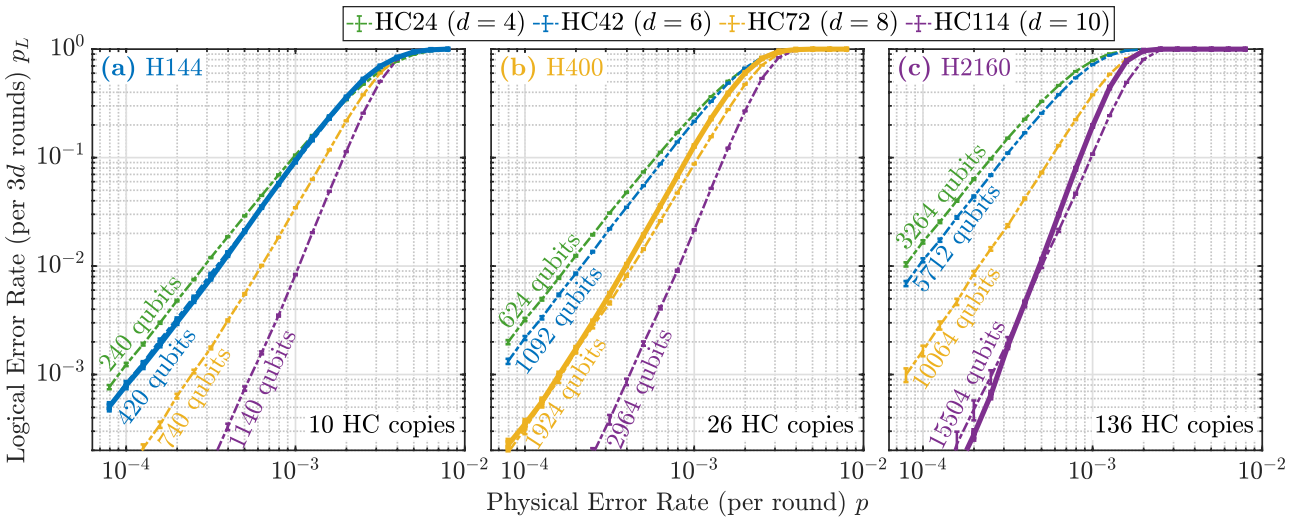


Figure 6: Comparison between hyperbolic and honeycomb Floquet codes under the phenomenological error model. Panels (a), (b), and (c) show the *any logical operator* error rate of H144, H400, and H2160 hyperbolic codes (solid lines) compared to 10 copies, 26 copies, and 136 copies of honeycomb codes (dashed lines), respectively. Honeycomb codes are denoted by HC followed by their number of physical qubits. Each honeycomb code stores two physical qubits. The number of copies of each honeycomb code is chosen such that each hyperbolic code with k logical qubits (see Table 1) is compared to ensembles of honeycomb codes that also have k total logical qubits. The total number of physical qubits needed for that many copies of honeycomb codes is noted alongside the respective dashed line.

effects on any of the $2k$ logical operators of a code is $1/2^{2k}$, which vanished as k increases. Therefore, this logical error rate saturates at ~ 1 for larger codes.

At lower values of the physical error rate, both *any logical operator error rate* and *specific logical operator error rate* [panels (a) and (b), respectively, in both figures] show a threshold behavior, as expected from a family of codes with increasing distance. Specifically, for sufficiently low physical error rates, $p < p_0$, the logical error rate decreases as the code's physical qubit count increases past $d = 4$. The threshold physical error rate, p_0, for both defined logical error rates is approximately $0.1\,\%$ per measurement round for the phenomenological error model (Fig. 4) and $0.2 - 0.5\,\%$ per measurement round for the entangling measurements error model (Fig. 5). However, it is noticeably smaller for the *any logical operator error rate* [Fig. 4(a) and Fig. 5(a)] compared to the *specific logical operator error rate* [Fig. 4(b) and Fig. 5(b)]. We denote these two values as p_0^{all} and p_0^{one}, respectively, and $p_0^{\text{all}} < p_0^{\text{one}}$. This subtle difference is a characteristic of finite encoding-rate codes. What links these two defined error rates is the number of logical operators, whereby a code with larger distance and a larger number of physical qubits has more logical operators that could experience a *specific logical operator* error, resulting in a *any logical operator* error.

The regime $p_0^{\text{all}} < p < p_0^{\text{one}}$ is an intermediate phase. For these error rates, the error rate for each logical operator is exponentially suppressed with the code distance [44]. On the other hand, there is an extensive number of logical operators that could potentially fail. Since the code distance is proportional to the logarithm of the number of physical qubits, and thus, of the number of logical operators, the possibilities for failure also grow exponentially with the code distance. As a result, even in the thermodynamic limit, the *any logical operator error rate* can remain nonzero for these intermediate values of the physical error rate, p. Only when $p < p_0^{\text{all}} < p_0^{\text{one}}$, will the *any logical operator error rate* also experience an exponential suppression with the code distance. This situation closely resembles the difference between the paramagnetic-ferromagnetic phase transitions in a random-bond Ising model (RBIM) on a hyperbolic lattice and in a dual-RBIM on the same lattice, as investigated in Ref. [45].

In addition to resulting in different values for the threshold physical error rate, error models also affect the behavior of each code. For example, H64 with a code distance of 4 (defined based on the minimum weight of logical operators dur-

ing Floquet rounds), behaves similarly to H144 ($d = 6$) under the phenomenological error model (Fig. 4), while it only slightly outperforms H16 ($d = 2$) under the entangling measurements error model (Fig. 5). This behavior is not unique to the hyperbolic Floquet codes and is observed for the honeycomb Floquet codes as well (see Appendix D). This shows that the true performance of a code under an error model depends on a code distance that is defined for that specific error model [4].

Comparison to Honeycomb Floquet Codes. The threshold physical error rates for hyperbolic codes are typically lower than those for related zero encoding-rate codes [28, 29]. This trend holds true for hyperbolic Floquet codes when compared to honeycomb Floquet codes as well. However, relying solely on the threshold value is not a useful method for comparing different codes. This value pertains to the thermodynamic limit and obscures the actual values of logical error rates, which are more practically relevant. Furthermore, below their threshold rates, hyperbolic codes are storing an extensive number of logical qubits, while the non-hyperbolic codes only preserve a constant number of them. Therefore, the threshold value alone falls short when comparing these codes performing the same task, namely, storing a given number of logical qubits. In response to this, we introduce a new figure of merit that aims at making this comparison [5].

In Fig. 6 and Fig. 7, we present our proposed comparison between hyperbolic Floquet codes and honeycomb Floquet codes, under the phenomenological and entangling measurements error models, respectively. The four honeycomb Floquet codes featured in these plots, denoted by HC followed by their respective number of physical qubits, are based on symmetric trivalent graphs in Ref. [40] with a *girth* of six, and are closed hexagonal lattices tessellating the surface of a torus [6]. Each panel in Fig. 6 and Fig. 7 illustrates the *any logical operator error rate* of a hyperbolic Floquet code (solid lines) compared to a number of copies of the honeycomb code with the same number of logical qubits (dashed lines). For instance, Fig. 6(a) and Fig. 7(a) compare the H144 code ($[[144, 20, 6]]$) with 10 copies of various honeycomb codes. Each of these honeycomb codes stores two logical qubits, and hence, 10 copies have a storage capacity for 20 logical qubits, equal to H144. The total number of physical qubits used in 10 copies of each honeycomb code is noted alongside its respective dashed line. The plotted logical error rate, p_L, for honeycomb codes is obtained by the reasonable assumption that the logical events in different copies are independent of each other.

As evident in Fig. 6 and Fig. 7, hyperbolic Floquet codes have performances on par with honeycomb codes with the same distance while using a significantly smaller number of physical qubits. For example, H400, with just 400 physical qubits, competes with honeycomb codes with up to 1924 qubits [Fig. 6(b) and Fig. 7(b)], and H2160, using 2160 physical qubits, compares to 15504-qubit honeycomb codes [Fig. 6(c) and Fig. 7(c)], providing respectively 5x and 7x reductions in the number of physical qubits. In general, as the logical error rate suppression is exponential in the code distance [44], it is expected that it would take honeycomb codes with equal or greater distances than a hyperbolic code to achieve a superior performance at low physical error rates. Thus, from a practical perspective, if their implementation is feasible within the architecture, hyperbolic Floquet codes represent a more sensible choice than honeycomb codes.

5 Discussion

In this work, we introduced a new family of dynamically generated codes called hyperbolic Floquet codes. These codes are useful for fault-tolerant implementation of multi-qubit quantum memories utilizing weight-two measurements. The hyperbolic nature of these codes enables us to find codes with a finite encoding rate and an increasing distance with the total number of physical qubits. We investigated the fault-tolerance of these codes under two different error models by testing an MWPM decoder on them and showing the existence of a threshold error rate, below which the logical error rates are suppressed. By comparing to honeycomb Floquet

[4] See Appendix C for more details in this regard.

[5] Independently introduced in Refs. [12, 46] as well.

[6] Read more about the honeycomb Floquet codes and their performance in Appendix D.

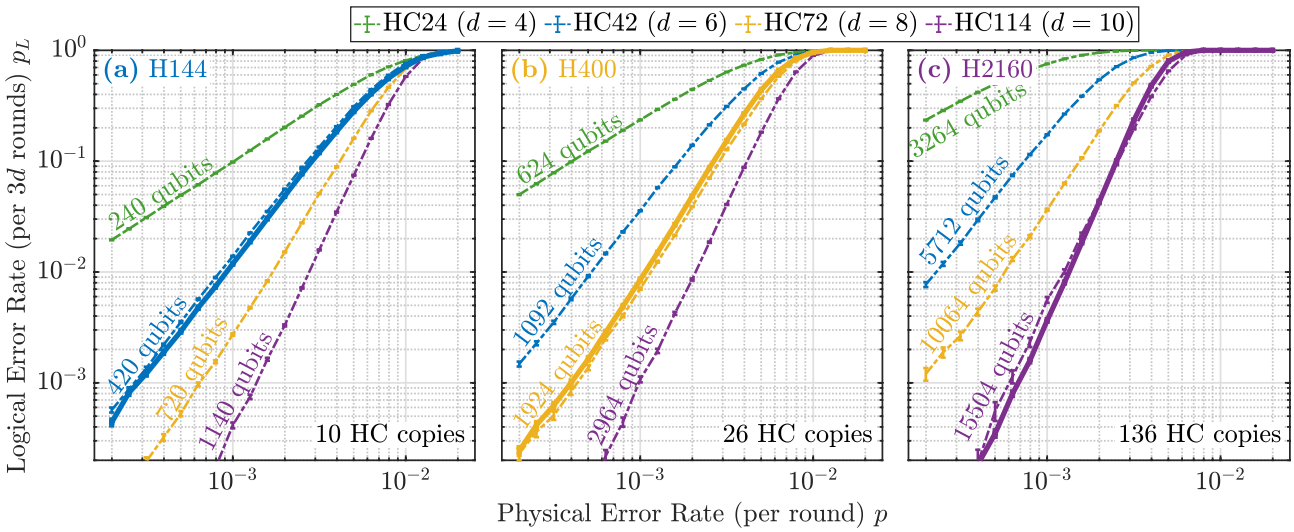


Figure 7: Comparison between hyperbolic and honeycomb Floquet codes under the entangling measurements error model. Panels (a), (b), and (c) show the *any logical operator* error rate of H144, H400, and H2160 hyperbolic codes (solid lines) compared to 10 copies, 26 copies, and 136 copies of honeycomb codes (dashed lines), respectively. Honeycomb codes are denoted by HC followed by their number of physical qubits. Each honeycomb code stores two physical qubits. The number of copies of each honeycomb code is chosen such that each hyperbolic code with k logical qubits (see Table 1) is compared to ensembles of honeycomb codes that also have k total logical qubits. The total number of physical qubits needed for that many copies of honeycomb codes is noted alongside the respective dashed line.

codes, we made the case that, given the same number of physical qubits, hyperbolic Floquet codes can achieve logical error suppression better than copies of honeycomb codes storing the same number of logical qubits. However, a noteworthy disadvantage for the hyperbolic codes is that the decoding task for a hyperbolic code-block of a given number of logical qubits can be more time-consuming than decoding copies of honeycomb code-blocks encoding the same number of logical qubits. Improving and optimizing decoders for hyperbolic codes, similar to other quantum LDPC codes, could reduce this barrier.

When encoding k logical qubits into n data qubits, we may also use c auxiliary qubits to measure the error syndrome. This means the code uses a total of $n + c$ physical qubits. Thus, the net encoding rate can be defined as $r = k/(n + c)$ [12]. For traditional surface code architectures, $c = n - 1$, whereas for the newer quantum LDPC codes, $c = n$ [12]. In our scheme, which takes inspiration from the properties of the Floquet code, we have $c = 0$ for platforms with native two-body measurements, leading to resource savings. Consequently, our net encoding rate is identical to the regular encoding rate, that is $k/n = 1/8 + 2/n$.

To further improve the logical error suppression of our approach, one can interpolate between hyperbolic and honeycomb Floquet codes by moving beyond regular hyperbolic tessellations. This has been previously implemented for hyperbolic surface codes, resulting in the so-called *semi-hyperbolic surface codes* [29]. The infusion of additional qubits within each regular polygon of a hyperbolic Floquet code will result in a code with the same number of logical qubits, but higher number of physical qubits. This new code, however, will have a greater distance than its parent. This procedure, sometimes referred to as *leap-frogging*, can be carried out repeatedly. Therefore, there exist families of codes with lower encoding rates than our hyperbolic Floquet codes, but a more favorable distance, although still scaling logarithmically with the number of physical qubits. Each family of these codes can also have higher threshold error rates than regular hyperbolic Floquet codes, approaching the threshold error rate of the honeycomb codes.

The imperfections in its components challenge the building of a quantum computer. For example, some qubits might possess manufacturing defects, rendering them "dead." These dead qubits are either entirely nonfunctional or have a noise level substantially higher than their counterparts. For the surface code, there are several

existing strategies [47, 48, 49, 50, 51, 52] to address these dead qubits. A recent method, as mentioned in Ref. [53], suggests recoupling the lattice applicable to any code that operates on a face three-colorable lattice. The recoupling can be viewed as a re-triangulation of the three-colorable lattice, ensuring dead qubits aren't utilized in the subsequent measurement schedule. Since our codes are also based on face three-colorable lattices, we can use the aforementioned strategy from Ref. [53] for handling dead qubits.

In this work, we have focused on the fault-tolerant memory threshold of hyperbolic Floquet codes. For an error correcting code family to be useful for quantum computation it is also necessary to implement state preparation, logical gates, and measurement. State preparation and measurement protocols for our hyperbolic Floquet codes can follow standard protocols from honeycomb Floquet codes [13, 30, 54]. Efficiently implementing logical gates with limited time overhead for finite-rate LDPC codes remains a fundamental challenge [20]. However, there has been recent progress adapting lattice surgery methods to LDPC codes that could be applied to our setting [29, 55, 56, 25].

Note added: While this manuscript was in preparation, Ref. [46] appeared discussing hyperbolic Floquet codes using a slightly different lattice construction method. A threshold was reported for zero-rate hyperbolic codes using the *leap-frogging* method. Owing to the difference in lattice constructions, they fail to capture some of the distance-maximizing codes presented here.

Acknowledgements

We thank I. Boettcher, H. Nguyen, Y.-Z. Chou, A. Lavasani, V. Albert, and M. Barkeshli for useful discussions. This research was supported in part by NSF QLCI (award No. OMA-2120757), the Air Force Office of Scientific Research, and the Simons Foundation. A.F., K.B., and A.V.G. were also supported in part by the DoE ASCR Accelerated Research in Quantum Computing program (award No. DE-SC0020312), DoE ASCR Quantum Testbed Pathfinder program (award No. DE-SC0019040), NSF PFCQC program (award No. 1818914), AFOSR (awards No. FA95502010108 and No. FA9550-19-1-0275), ARO MURI (award No. W911NF1610349), AFOSR MURI (award No. FA9550-20-1-0323), and DARPA SAVaNT ADVENT (award No. W911NF2120106). H.D. acknowledges support from DOE DE-AC02-05CH11231, AFOSR FA9550-22-1-0339, OMA-2120757 and the Simons Investigator in Physics Award. Support is also acknowledged from the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers, Quantum Systems Accelerator. The authors acknowledge the University of Maryland supercomputing resources (`http://hpcc.umd.edu`) made available for conducting the research reported in this paper.

References

[1] A. Y. Kitaev. "Quantum Error Correction with Imperfect Gates". In O. Hirota, A. S. Holevo, and C. M. Caves, editors, Quantum Communication, Computing, and Measurement. Pages 181–188. Springer US (1997).

[2] A. Y. Kitaev. "Fault-tolerant quantum computation by anyons". Annals of physics **303**, 2–30 (2003).

[3] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill. "Topological quantum memory". Journal of Mathematical Physics **43**, 4452–4505 (2002).

[4] M. Takita, A. D. Córcoles, E. Magesan, B. Abdo, M. Brink, A. Cross, J. M. Chow, and J. M. Gambetta. "Demonstration of Weight-Four Parity Measurements in the Surface Code Architecture". Physical Review Letters **117**, 210505 (2016).

[5] J. F. Marques, B. M. Varbanov, M. S. Moreira, H. Ali, N. Muthusubramanian, C. Zachariadis, F. Battistel, M. Beekman, N. Haider, W. Vlothuizen, A. Bruno, B. M. Terhal, and L. DiCarlo. "Logical-qubit operations in an error-detecting surface code". Nature Physics **18**, 80–86 (2022).

[6] S. Krinner, N. Lacroix, A. Remm, A. Di Paolo, E. Genois, C. Leroux, C. Hellings, S. Lazar, F. Swiadek, J. Herrmann, G. J. Norris, C. K. Andersen, M. Müller, A. Blais, C. Eichler, and A. Wallraff. "Realizing repeated quantum

error correction in a distance-three surface code". Nature **605**, 669–674 (2022).

[7] Y. Zhao, Y. Ye, H.-L. Huang, Y. Zhang, D. Wu, H. Guan, Q. Zhu, Z. Wei, T. He, S. Cao, F. Chen, T.-H. Chung, H. Deng, D. Fan, M. Gong, C. Guo, S. Guo, L. Han, N. Li, S. Li, Y. Li, F. Liang, J. Lin, H. Qian, H. Rong, H. Su, L. Sun, S. Wang, Y. Wu, Y. Xu, C. Ying, J. Yu, C. Zha, K. Zhang, Y.-H. Huo, C.-Y. Lu, C.-Z. Peng, X. Zhu, and J.-W. Pan. "Realization of an Error-Correcting Surface Code with Superconducting Qubits". Physical Review Letters **129**, 030501 (2022).

[8] R. Acharya, I. Aleiner, R. Allen, T. I. Andersen, M. Ansmann, F. Arute, K. Arya, A. Asfaw, J. Atalaya, R. Babbush, D. Bacon, J. C. Bardin, J. Basso, A. Bengtsson, S. Boixo, G. Bortoli, A. Bourassa, J. Bovaird, L. Brill, M. Broughton, B. B. Buckley, D. A. Buell, T. Burger, B. Burkett, N. Bushnell, Y. Chen, Z. Chen, B. Chiaro, J. Cogan, R. Collins, P. Conner, W. Courtney, A. L. Crook, B. Curtin, D. M. Debroy, A. Del Toro Barba, S. Demura, A. Dunsworth, D. Eppens, C. Erickson, L. Faoro, E. Farhi, R. Fatemi, L. Flores Burgos, E. Forati, A. G. Fowler, B. Foxen, W. Giang, C. Gidney, D. Gilboa, M. Giustina, A. Grajales Dau, J. A. Gross, S. Habegger, M. C. Hamilton, M. P. Harrigan, S. D. Harrington, O. Higgott, J. Hilton, M. Hoffmann, S. Hong, T. Huang, A. Huff, W. J. Huggins, L. B. Ioffe, S. V. Isakov, J. Iveland, E. Jeffrey, Z. Jiang, C. Jones, P. Juhas, D. Kafri, K. Kechedzhi, J. Kelly, T. Khattar, M. Khezri, M. Kieferová, S. Kim, A. Kitaev, P. V. Klimov, A. R. Klots, A. N. Korotkov, F. Kostritsa, J. M. Kreikebaum, D. Landhuis, P. Laptev, K.-M. Lau, L. Laws, J. Lee, K. Lee, B. J. Lester, A. Lill, W. Liu, A. Locharla, E. Lucero, F. D. Malone, J. Marshall, O. Martin, J. R. McClean, T. McCourt, M. McEwen, A. Megrant, B. Meurer Costa, X. Mi, K. C. Miao, M. Mohseni, S. Montazeri, A. Morvan, E. Mount, W. Mruczkiewicz, O. Naaman, M. Neeley, C. Neill, A. Nersisyan, H. Neven, M. Newman, J. H. Ng, A. Nguyen, M. Nguyen, M. Y. Niu, T. E. O'Brien, A. Opremcak, J. Platt, A. Petukhov, R. Potter, L. P. Pryadko, C. Quintana, P. Roushan, N. C. Rubin, N. Saei, D. Sank, K. Sankaragomathi, K. J. Satzinger, H. F. Schurkus, C. Schuster, M. J. Shearn, A. Shorter, V. Shvarts, J. Skruzny, V. Smelyanskiy, W. C. Smith, G. Sterling, D. Strain, M. Szalay, A. Torres, G. Vidal, B. Villalonga, C. Vollgraff Heidweiller, T. White, C. Xing, Z. J. Yao, P. Yeh, J. Yoo, G. Young, A. Zalcman, Y. Zhang, N. Zhu, and Google Quantum AI. "Suppressing quantum errors by scaling a surface code logical qubit". Nature **614**, 676–681 (2023).

[9] D. Bluvstein, H. Levine, G. Semeghini, T. T. Wang, S. Ebadi, M. Kalinowski, A. Keesling, N. Maskara, H. Pichler, M. Greiner, V. Vuletić, and M. D. Lukin. "A quantum processor based on coherent transport of entangled atom arrays". Nature **604**, 451–456 (2022).

[10] I. Cong, N. Maskara, M. C. Tran, H. Pichler, G. Semeghini, S. F. Yelin, S. Choi, and M. D. Lukin. "Enhancing detection of topological order by local error correction". Nature Communications **15**, 1527 (2024).

[11] G. Semeghini, H. Levine, A. Keesling, S. Ebadi, T. T. Wang, D. Bluvstein, R. Verresen, H. Pichler, M. Kalinowski, R. Samajdar, A. Omran, S. Sachdev, A. Vishwanath, M. Greiner, V. Vuletić, and M. D. Lukin. "Probing topological spin liquids on a programmable quantum simulator". Science **374**, 1242–1247 (2021).

[12] S. Bravyi, A. W. Cross, J. M. Gambetta, D. Maslov, P. Rall, and T. J. Yoder. "High-threshold and low-overhead fault-tolerant quantum memory". Nature **627**, 778–782 (2024).

[13] M. B. Hastings and J. Haah. "Dynamically Generated Logical Qubits". Quantum **5**, 564 (2021).

[14] R. Chao, M. E. Beverland, N. Delfosse, and J. Haah. "Optimization of the surface code design for Majorana-based qubits". Quantum **4**, 352 (2020).

[15] C. Gidney. "A Pair Measurement Surface Code on Pentagons". Quantum **7**, 1156 (2023).

[16] D. K. Tuckett, A. S. Darmawan, C. T. Chubb, S. Bravyi, S. D. Bartlett, and S. T. Flammia. "Tailoring Surface Codes for Highly Biased Noise". Physical Review X **9**, 041031 (2019).

[17] J. P. Bonilla Ataides, D. K. Tuckett, S. D. Bartlett, S. T. Flammia, and B. J. Brown. "The XZZX surface code". Nature Communications **12**, 2172 (2021).

[18] A. Dua, A. Kubica, L. Jiang, S. T. Flammia, and M. J. Gullans. "Clifford-Deformed Surface Codes". PRX Quantum **5**, 010347 (2024).

[19] Y. Wu, S. Kolkowitz, S. Puri, and J. D. Thompson. "Erasure conversion for fault-tolerant quantum computing in alkaline earth Rydberg atom arrays". Nature Communications **13**, 4657 (2022).

[20] D. Gottesman. "Fault-Tolerant Quantum Computation with Constant Overhead" (2014) arXiv:1310.2984.

[21] M. A. Tremblay, N. Delfosse, and M. E. Beverland. "Constant-Overhead Quantum Error Correction with Thin Planar Connectivity". Physical Review Letters **129**, 050504 (2022).

[22] N. P. Breuckmann and J. N. Eberhardt. "Quantum Low-Density Parity-Check Codes". PRX Quantum **2**, 040101 (2021).

[23] P. Panteleev and G. Kalachev. "Asymptotically good Quantum and locally testable classical LDPC codes". In Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing. Pages 375–388. STOC 2022. Association for Computing Machinery (2022).

[24] A. Leverrier and G. Zémor. "Quantum Tanner codes". In 2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS). Pages 872–883. 2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS) (2022).

[25] Q. Xu, J. P. Bonilla Ataides, C. A. Pattison, N. Raveendran, D. Bluvstein, J. Wurtz, B. Vasić, M. D. Lukin, L. Jiang, and H. Zhou. "Constant-overhead fault-tolerant quantum computation with reconfigurable atom arrays". Nature Physics **20**, 1084–1090 (2024).

[26] M. H. Freedman, D. A. Meyer, and F. Luo. "Z2-systolic freedom and quantum codes". In Mathematics of Quantum Computation. Chapman and Hall/CRC (2002).

[27] G. Zémor. "On Cayley Graphs, Surface Codes, and the Limits of Homological Coding for Quantum Error Correction". In Y. M. Chee, C. Li, S. Ling, H. Wang, and C. Xing, editors, Coding and Cryptology. Volume 5557, pages 259–273. Springer Berlin Heidelberg (2009).

[28] N. P. Breuckmann and B. M. Terhal. "Constructions and Noise Threshold of Hyperbolic Surface Codes". IEEE Transactions on Information Theory **62**, 3731–3744 (2016).

[29] N. P. Breuckmann, C. Vuillot, E. Campbell, A. Krishna, and B. M. Terhal. "Hyperbolic and semi-hyperbolic surface codes for quantum storage". Quantum Science and Technology **2**, 035007 (2017).

[30] C. Gidney, M. Newman, A. Fowler, and M. Broughton. "A Fault-Tolerant Honeycomb Memory". Quantum **5**, 605 (2021).

[31] C. Vuillot. "Planar Floquet Codes" (2021) arXiv:2110.05348.

[32] T. D. Ellison, J. Sullivan, and A. Dua. "Floquet codes with a twist" (2023) arXiv:2306.08027.

[33] M. Davydova, N. Tantivasadakarn, and S. Balasubramanian. "Floquet Codes without Parent Subsystem Codes". PRX Quantum **4**, 020341 (2023).

[34] M. S. Kesselring, J. C. Magdalena De La Fuente, F. Thomsen, J. Eisert, S. D. Bartlett, and B. J. Brown. "Anyon Condensation and the Color Code". PRX Quantum **5**, 010342 (2024).

[35] M. Davydova, N. Tantivasadakarn, S. Balasubramanian, and D. Aasen. "Quantum computation from dynamic automorphism codes". Quantum **8**, 1448 (2024).

[36] D. Aasen, Z. Wang, and M. B. Hastings. "Adiabatic paths of Hamiltonians, symmetries of topological order, and automorphism codes". Physical Review B **106**, 085122 (2022).

[37] J. Sullivan, R. Wen, and A. C. Potter. "Floquet codes and phases in twist-defect networks". Physical Review B **108**, 195134 (2023).

[38] V. A. Blatov, M. O'Keeffe, and D. M. Proserpio. "Vertex-, face-, point-, Schläfli-, and Delaney-symbols in nets, polyhedra and tilings: Recommended terminology". CrystEngComm **12**, 44–48 (2010).

[39] D. Aasen, J. Haah, Z. Li, and R. S. K. Mong. "Measurement Quantum Cellular Automata and Anomalies in Floquet Codes" (2023) arXiv:2304.01277.

[40] M. Conder and P. Dobcsányi. "Trivalent symmetric graphs on up to 768 vertices". Journal of Combinatorial Mathematics and Combinatorial Computing **40**, 41–64 (2002).

[41] I. Boettcher, A. V. Gorshkov, A. J. Kollár, J. Maciejko, S. Rayan, and R. Thomale. "Crystallography of hyperbolic lattices". Physical Review B **105**, 125118 (2022).

[42] S. D. Sarma, M. Freedman, and C. Nayak. "Majorana zero modes and topological quantum computation". npj Quantum Information **1**, 1–13 (2015).

[43] O. Higgott and C. Gidney. "Sparse Blossom: Correcting a million errors per core second with minimum-weight matching". Quantum **9**, 1600 (2025).

[44] S. Bravyi and A. Vargo. "Simulation of rare events in quantum error correction". Physical Review A **88**, 062308 (2013).

[45] B. Placke and N. P. Breuckmann. "Random-bond Ising model and its dual in hyperbolic spaces". Physical Review E **107**, 024125 (2023).

[46] O. Higgott and N. P. Breuckmann. "Constructions and Performance of Hyperbolic and Semi-Hyperbolic Floquet Codes". PRX Quantum **5**, 040327 (2024).

[47] T. M. Stace, S. D. Barrett, and A. C. Doherty. "Thresholds for Topological Codes in the Presence of Loss". Physical Review Letters **102**, 200501 (2009).

[48] T. M. Stace and S. D. Barrett. "Error correction and degeneracy in surface codes suffering loss". Physical Review A **81**, 022317 (2010).

[49] Y.-C. Tang and G.-X. Miao. "Robust surface code topology against sparse fabrication defects in a superconducting-qubit array". Physical Review A **93**, 032322 (2016).

[50] S. Nagayama, A. G. Fowler, D. Horsman, S. J. Devitt, and R. V. Meter. "Surface code error correction on a defective lattice". New Journal of Physics **19**, 023050 (2017).

[51] A. Strikis, S. C. Benjamin, and B. J. Brown. "Quantum Computing is Scalable on a Planar Array of Qubits with Fabrication Defects". Physical Review Applied **19**, 064081 (2023).

[52] A. Siegel, A. Strikis, T. Flatters, and S. Benjamin. "Adaptive surface code for quantum error correction in the presence of temporary or permanent defects". Quantum **7**, 1065 (2023).

[53] D. Aasen, J. Haah, P. Bonderson, Z. Wang, and M. Hastings. "Fault-tolerant Hastings-Haah codes in the presence of dead qubits" (2023) arXiv:2307.03715.

[54] J. C. Magdalena de la Fuente, J. Old, A. Townsend-Teague, M. Rispler, J. Eisert, and M. Müller. "XYZ Ruby Code: Making a Case for a Three-Colored Graphical Calculus for Quantum Error Correction in Spacetime". PRX Quantum **6**, 010360 (2025).

[55] D. Horsman, A. G. Fowler, S. Devitt, and R. V. Meter. "Surface code quantum computing by lattice surgery". New Journal of Physics **14**, 123011 (2012).

[56] L. Z. Cohen, I. H. Kim, S. D. Bartlett, and B. J. Brown. "Low-overhead fault-tolerant quantum computing using long-range connectivity". Science Advances **8**, eabn1717 (2022).

A Details of Error Detection and Correction

Sec. 3 contains a summary of the essentials of our decoding method. In this Appendix, we provide the full technical description of error detection and correction.

Syndrome Lattice. Floquet codes possess the capability to detect potential error occurrences and provide the necessary information for attempting to correct the effects of these errors on logical qubits. Error detection is executed by utilizing the inferred values of the plaquette operators, hereafter referred to as *plaquette syndromes* or simply *syndromes*. Each plaquette syndrome is inferred every third round. The inference rounds depend on plaquette's color; specifically, the red, green, and blue plaquette syndromes are inferred at rounds $r = 3n + 2$, $3n + 3$, and $3n + 1$, respectively, for every nonnegative integer n. (No plaquette syndrome is inferred at round $r = 0$.) By extending a line perpendicular to the hyperbolic plane for each plaquette and marking each syndrome update with a circle on this line, we construct a $(2 + 1)D$ lattice comprising updates of plaquette syndromes, as partially depicted in Fig. 8(a). We refer to this lattice as the *syndrome lattice*.

In the absence of errors, the plaquette operators become part of the ISG upon their initial inference (rounds $r = 2$, $r = 3$, and $r = 1$ for red, green, and blue plaquettes, respectively) and remain in the ISG, as they commute with all check operators. Consequently, any subsequent inference of a plaquette syndrome yields a value equal to the previously inferred one. This condition is denoted by empty circles in the $(2 + 1)D$ lattice of syndromes in Fig. 8(a). Conversely, an updated plaquette syndrome that differs from its previous value implies the presence of an error (or errors) affecting that syndrome. Following Gidney, we call these events *detection events*. They are indicated by filled circles in the $(2 + 1)D$ lattice of syndromes.

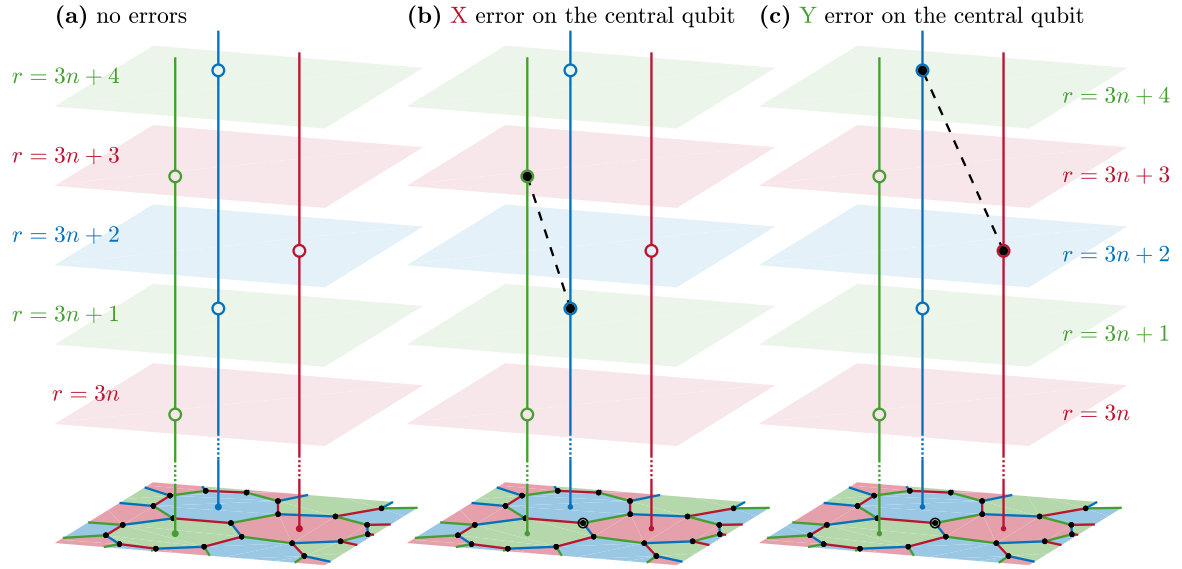


Figure 8: Detection events for Pauli errors. (a) $(2 + 1)D$ lattice of plaquette syndromes. Each plaquette syndrome is inferred every third measurement round, denoted by empty circles in the timelines of the plaquettes. (b) X and (c) Y Pauli errors between rounds $r = 3n$ and $r = 3n + 1$ lead to detection events marked on lines perpendicular to the hyperbolic plane. In the presence of errors, inference of a plaquette syndrome may yield a value different than its most recent inferred value. In this case, the inferred syndrome exhibits a detection event, denoted by a filled circle. For the Pauli errors happening between rounds $r = 3n$ and $r = 3n + 1$, X and Y Paulis are simple Pauli errors and are associated with two detection events, connected to each other by an error edge in the syndrome lattice. A Z Pauli, on the other hand, is a compound Pauli error and is equivalent to a combination of these X and Y Paulis encompassing all four detection events. The four detection events of a compound Pauli error form a hyper-edge in the syndrome lattice.

Error Syndromes. The lattice's edge three-coloring ensures that the three measurement checks involving a given physical qubit have distinct Pauli operators associated with that qubit. As a result, a Pauli error on a physical qubit anticommutes with two of those checks. For instance, the X Pauli error on the qubit marked with a circle in Fig. 8(b) anticommutes with the green (YY) and blue (ZZ) checks intersecting at that vertex. Assuming the error occurs between rounds $r = 3n$ and $r = 3n + 1$, the immediately following green and blue checks will be affected by the error. The green check's faulty outcome will be used in the inference of the adjacent blue plaquette syndrome resulting in a detection event at round $r = 3n + 1$. This is denoted by a filled circle in Fig. 8(b). At round $r = 3n + 2$, the adjacent red plaquette syndrome will be inferred using two faulty outcomes of both green and blue checks, and hence, will be agnostic to the error. No detection event for the red plaquette syndrome is marked with an empty circle in Fig. 8(b). The green plaquette syndrome's inference at round $r = 3n + 3$, however, will use the outcomes from the faulty blue check and the unaffected red check, exhibiting a detection event similar to the blue plaquette syndrome, indicated with a filled circle. The two detection events of the X Pauli error form an error edge in the syndrome lattice, represented by a dashed line in Fig. 8(b).

The analysis for a Y Pauli error on the same qubit follows similarly. The Y Pauli error on the qubit marked with a circle and its detection events are illustrated in Fig. 8(c). The Y Pauli anticommutes with the red and blue checks that intersect at the vertex, while leaving the green check unaffected. Assuming once again that the error occurs between rounds $r = 3n$ and $r = 3n + 1$, the green check's measurement at round $r = 3n + 1$ is agnostic to the error. However, the blue check's and the red check's measurements at rounds $r = 3n + 2$ and $r = 3n + 3$ do capture the error. During round $r = 3n + 2$, the red syndrome is updated using the outcomes of the unaffected green check and the faulty blue check, thereby triggering a detection event. In the next round $r = 3n + 3$, the green syndrome is updated using faulty outcomes from both the blue and red checks, leaving it unaffected. Lastly, at round $r = 3n + 4$, the outcomes from the faulty red check and the unaffected green check lead the blue syndrome to exhibit another detection event. These two detection events of the Y Pauli error form an edge in the syndrome lattice, as shown in Fig. 8(c). Notably, for these errors occurring between rounds $r = 3n$ and $r = 3n + 1$, the two detection events for the Y Pauli error happen with a one round delay as compared to the case of the X Pauli error.

A Z Pauli error that occurs between rounds $r = 3n$ and $r = 3n + 1$ can be regarded as a combination of X and Y Pauli errors, following from $Z = -iXY$. As a result, the associated detection events are also a combination of those corresponding to X and Y Pauli errors. Specifically, a Z Pauli error between rounds $r = 3n$ and $r = 3n + 1$ leads to the following four detection events: the blue syndrome at $r = 3n + 1$, the red syndrome at $r = 3n + 2$, the green syndrome at $r = 3n + 3$, and once again, the blue syndrome at $r = 3n + 4$. These four detection events create a hyper-edge (i.e., an "edge" that connects more than two vertices) in the syndrome lattice.

The decisive factor for a Pauli error to yield either two or four detection events is the most recent round of measurements. In the outlined example above, the Pauli errors take place between rounds $r = 3n$ (XX measurements) and $r = 3n + 1$ (YY measurements). In this scenario, X and Y Pauli errors generate two detection events, while the Z Pauli is associated with four. We call Pauli errors with two detection events (X and Y in this case) *simple Pauli errors* and those with four detection events (Z in this case) *compound Pauli errors*. During the interval between YY and ZZ measurements, Y and Z Pauli errors are simple, while X Pauli errors are compound. Similarly, between ZZ and XX measurements, Z and X Pauli errors are simple, and Y Pauli errors are compound.

In a measurement error, the outcome of a check's measurement is faulty. This faulty outcome is then utilized in inferring two syndromes, causing them to exhibit detection events. An example is illustrated in Fig. 9(a). In this example, the highlighted red check's outcome at round $r = 3n$ is erroneous, while all other measurements are accurate. This faulty red check leads to the green plaquette syndrome at round $r = 3n$ and the blue plaquette syndrome at round $r = 3n + 1$ exhibiting detection events. Given the assumption that all other measurements are accurate, the next measurement of the same red check at round $r = 3n + 3$ relates to the accurate value of the plaquette syndromes. Thus, the same green

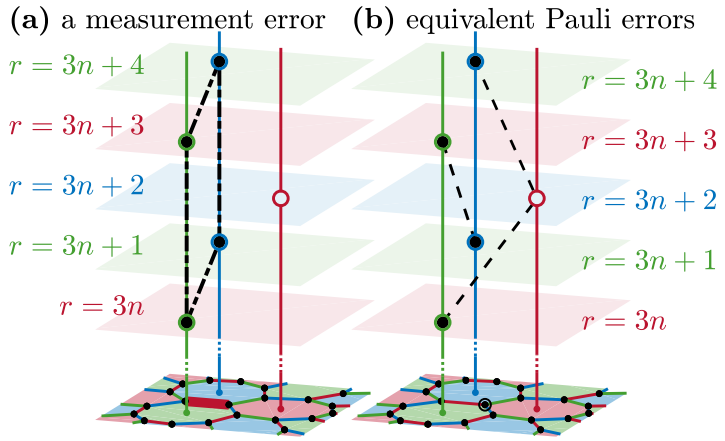


Figure 9: Detection events of a measurement error. (a) The faulty measurement outcome of the highlighted red check at round $r = 3n$ is used to infer an incorrect value for the adjacent green and blue plaquette syndromes at rounds $r = 3n$ and $r = 3n + 1$. Later correct measurement of the same check leads to correct inference of these syndromes three rounds later. In both cases, the inferred syndrome differs from the most recent inference and gives rise to a detection event. Hence, a measurement error has four detection events forming a hyper-edge in the syndrome lattice, shown with a dashed parallelogram. (b) A combination of three simple Pauli errors on one of the qubits of the faulty check, denoted with a circle, gives rise to the same detection events. Three simple Pauli errors in this case are: a Z error between rounds $r = 3n - 1$ and $r = 3n$, an X error between rounds $r = 3n$ and $r = 3n + 1$, and a Y error between rounds $r = 3n + 1$ and $r = 3n + 2$. These equivalent Pauli errors have the same effects on the logical operators as the measurement error.

and blue syndromes generate additional detection events three rounds later. A measurement error, therefore, is linked to four detection events, forming a hyper-edge in the syndrome lattice, depicted with a dashed parallelogram in Fig. 9(a).

Error Detection and Correction. The occurrence of each error may affect the logical operators of the code. For a Pauli error on a physical qubit, this happens when that qubit is on the noncontractible closed path of a logical operator featuring a distinct nonidentity Pauli on it compared to the error. In this scenario, the Pauli operator and the logical operator anticommute, leading the error to induce a sign flip in the logical operator. In the case of a measurement error, it impacts a logical operator if the measured check is along the closed path of that logical operator. Consequently, the faulty measurement outcome is used in updating the logical operator's expression, resulting in an incorrect sign. We argue that the detection events associated with each error provide sufficient information for attempting to correct the respective error. This attempt, however, is not always successful, and if the number of errors equals to or exceeds $d/2$, can lead to logical errors.

Consider the pair of detection events linked to a simple Pauli error. There are four different Pauli errors that could trigger these two detection events and, thus, are indistinguishable from one another. Without any loss of generality, assume the simple Pauli error is an X error. The two plaquette syndromes with detection events correspond to adjacent green and blue octagons [see Fig. 8(b)]. These two octagons identify the location of the physical qubit affected by the X error to their shared red edge, without distinguishing which of the two ends of that edge it is. We label these two qubits as q_1 and q_2. Moreover, the X error could have occurred either between the most recent XX and YY measurements or between the most recent ZZ and XX measurements. These possibilities for error's support qubit, q_1 or q_2, and its timing, between ZZ and XX measurements or between XX and YY measurements, give rise to the four indistinguishable errors with identical detection events.

Now consider a logical operator, L, that contains one of these two qubits (q_1, without loss of generality) in its closed path. Between the most recent ZZ and XX measurements, L's expression can either be type one (Z Paulis on red checks) or type two (X Paulis on blue checks). In each case, between the most recent XX and YY measurements, L's expression will be type two (Y Paulis on red

checks) or type one (X Paulis on green checks), respectively. The X error on q_1 affects L only if L's expression features Y or Z Paulis on q_1 at the time of the error. This holds true if L is a type one and then a type two logical operator, in addition to having the red edge of q_1's vertex along its closed path. In this case, L's expression during the two possible time intervals includes Y or Z Paulis on both q_1 and q_2. Therefore, all four indistinguishable simple Pauli errors impact logical operators in the same way, and their indistinguishability does not pose an obstruction in correcting their effects on logical operators.

A compound Pauli error is indistinguishable from the pair of simple Pauli errors that constitute its detection events. In the example illustrated in Fig. 8, a Z error on the qubit marked with a circle between rounds $r = 3n$ and $r = 3n + 1$ causes the same detection events as a combination of X and Y errors on that qubit occurring during the same time interval (or their indistinguishable counterparts). Let us label this qubit q_1 and consider a logical operator, L, with q_1 on its path. Between XX and YY measurements, L's expression is either type one (X Paulis on green checks) or type two (Y Paulis on red checks). The Z error affects L only if its expression features X or Y Paulis on q_1. In such a case, one of the X or Y errors on q_1 anticommutes with L, while the other one commutes with it. As as result, a compound Pauli error and its indistinguishable simple Pauli error pairs impact logical operators in the same way, and the indistinguishability again does not pose an obstruction to error correction.

The indistinguishability analysis also extends to measurement errors. A measurement error is indistinguishable from a combination of three simple Pauli errors. These equivalent Pauli errors for an erroneous measurement of a red check are shown in Fig. 9(b). The three Pauli errors consist of: first, a Z between $r = 3n - 1$ and $r = 3n$; second, an X between $r = 3n$ and $r = 3n + 1$; and third, a Y between $r = 3n + 1$ and $r = 3n + 2$. All these errors impact the physical qubit marked with a circle in Fig. 9(b). Equivalent Pauli errors for measurement errors of central green and blue checks can be obtained by $2\pi/3$ or $4\pi/3$ rotations in the hyperbolic plane, followed by one- or two-round shifts in time.

A logical operator, L, with the faulty red edge on its closed path will incorporate the erroneous outcome of its measurement in the update protocol. This will lead to an incorrect sign in its expression. If L's expression immediately before the measurement error at round $r = 3n$ is type one (Z Paulis on red checks), its subsequent two expressions will be type two (Y Paulis on red checks) and, then, type one (Y Paulis on blue checks). Among the three equivalent Pauli errors, the first and third one commute with L, while the second one anticommutes with it. Hence, their combined effect would be a sign flip in L's expression, exactly as in the case of the measurement error.

Analogously, if L's initial expression immediately before the faulty measurement's round is type two (X Paulis on blue checks), its subsequent two expressions will be X Paulis on green checks and Z Paulis on green checks. Depending on whether L's path traverses the blue or green check connected to the marked qubit, either the first or third Pauli error among the three equivalent errors will anticommute with L, while the other two will commute with it. Therefore, the combined effect of these equivalent Pauli errors is again the same as for the measurement error. Thus, the indistinguishability of measurement errors and combinations of simple Pauli errors does not pose an obstruction to error correction.

To simplify our decoder, we leverage the fact that all errors can be expressed as equivalent combinations of simple Pauli errors resulting in the same detection events and impacting the logical operators identically. Specifically, our decoder operates under the assumption that only simple Pauli errors have occurred. It then identifies a set of such errors associated with the observed detection events. Given that each simple error is linked to two detection events, forming an error edge in the syndrome lattice, the decoder performs this by pairing (with a path of possible error edges) each detection event in the syndrome lattice with another detection event. To achieve this, we use the Minimum-Weight Perfect Matching (MWPM) decoder from the PyMatching package [43].

Misidentifying more complex errors as combinations of simple ones can have a negative impact on the performance of the decoder. However, being able to utilize MWPM speeds up the decoding

process significantly. In this paper, we do not optimize the decoder substantially and instead focus on establishing the existence of a threshold. Since our decoder is easily adaptable to honeycomb Floquet codes, we can also perform a direct comparison between hyperbolic and honeycomb Floquet codes.

B Threshold Plots with Various Duration of the Applied Errors

In this Appendix, we numerically study the dependence of the threshold behaviour reported in Sec. 4 on the choice of $3d$ rounds of erroneous simulation for each code, where d is the code distance.

In Fig. 10 we present logical error rates of the hyperbolic Floquet codes under the entangling measurement error model for various duration of the applied errors. In all four panels the duration in which the errors are applied is proportional to the distance of the code so that the observed phase transition is a property of the three-dimensional space-time lattice of syndromes. To be more precise, this duration equals hd rounds of check measurements with $h = 1$-4 for panels (a)-(d). As seen here, the approximate value of the threshold error rate is around 0.25 % for all four panels and does not depend on the value of h. All plots in the main text are produced with $h = 3$.

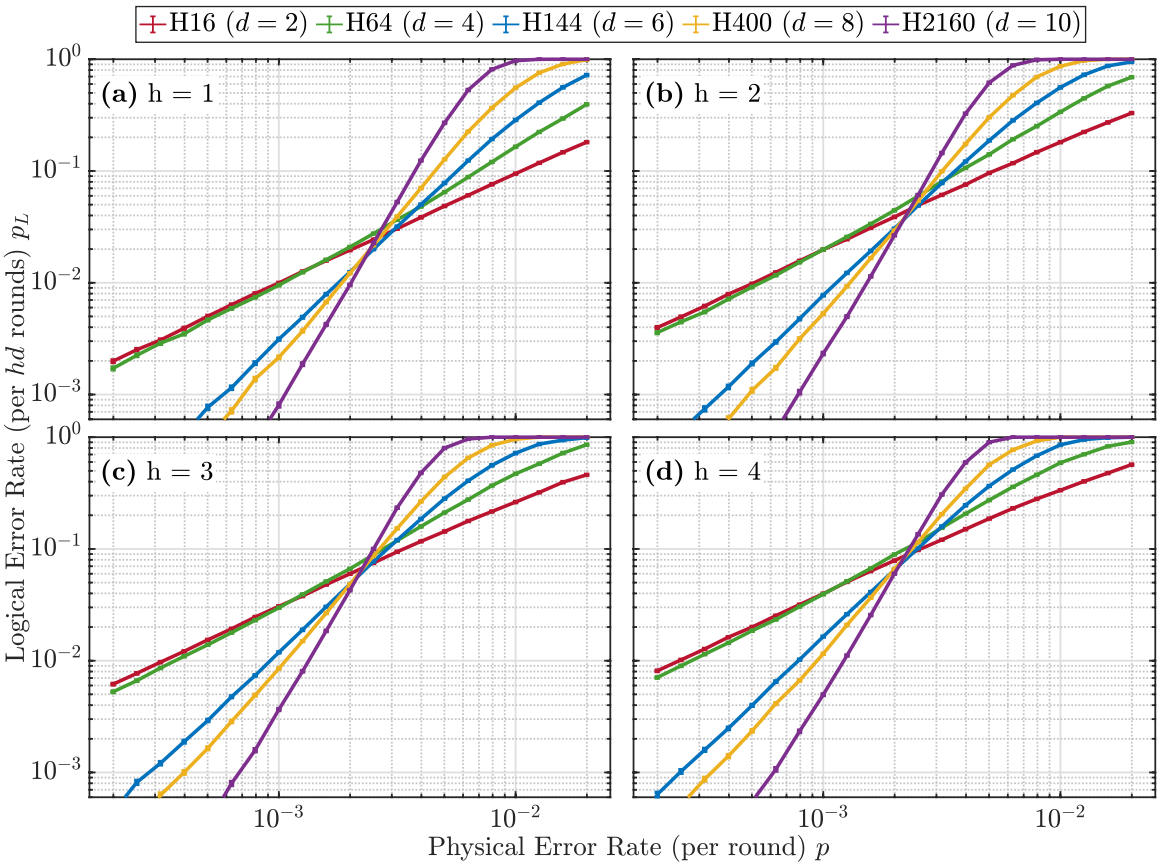


Figure 10: Logical error rates of the hyperbolic Floquet codes under the entangling measurements error model for various duration of the applied errors.

C Empirical Distances of Hyperbolic Floquet Codes

In this appendix, we investigate the performance of hyperbolic codes in suppressing logical errors. It is expected that at low physical error rates, p, the logical error rate of a code, p_L, is expected to decay exponentially with the code distance, d, [44]:

$$p_L = A(d)p^{-\alpha d}, \tag{5}$$

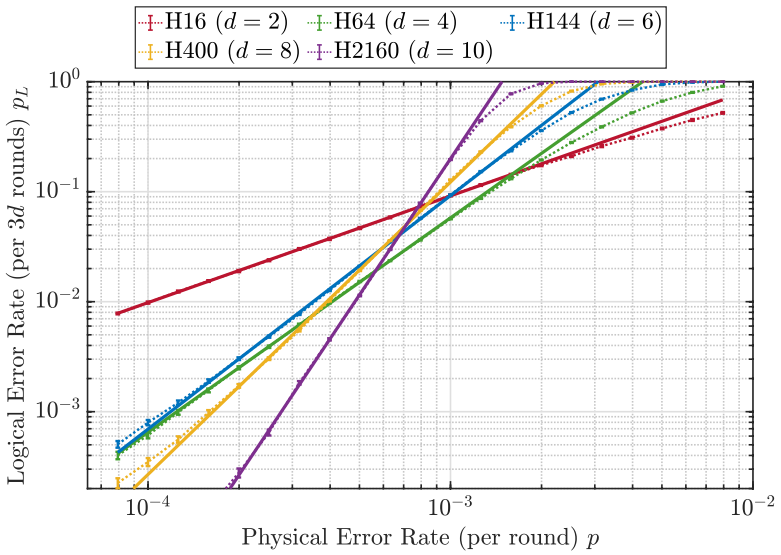


Figure 11: Any logical operator error rates for the family of hyperbolic Floquet codes under the phenomenological error model (dotted lines) [same as Fig. 4(a)] and linear fits to the values below the approximate error rate threshold of $p_0^{\text{all}} \simeq 8 \times 10^{-4}$ per round (solid lines).

where $A(d)$ is a coefficient dependent on the code distance and α is a constant.

The linear behavior of logical error rates in log-log plots under both error models (Figs. 11 and 12) provides numerical evidence for Eq. 5. In these figures, we also show linear fits for each line below the corresponding approximate error rate threshold. The extracted slopes from these fits should be proportional to what we call the *empirical distances* of the codes. Similar to the threshold rate, we expect empirical distances to depend on the code, the error model, and the decoder.

Figure 13 shows the obtained empirical distances of hyperbolic codes under the phenomenological (panel a) and entangling measurements (panel b) error models. These empirical distances are assumed to be proportional to the slope of the fitted lines in Figs. 11 and 12. The proportionality coefficient is chosen such that, in each of the panels in Fig. 13, the square sum of the length of the dotted lines is minimized. These dotted lines connect each code's data point to the solid line with a slope equal to

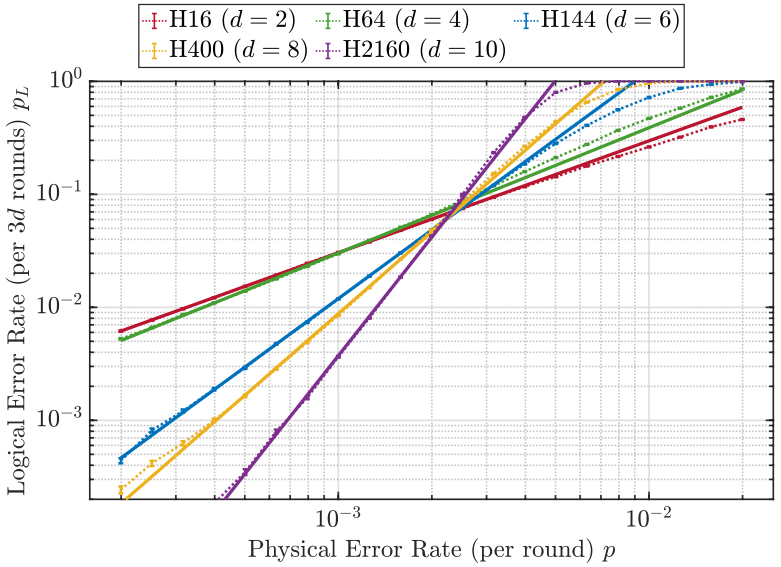


Figure 12: Any logical operator error rates for the family of hyperbolic Floquet codes under the entangling measurements error model (dotted lines) [same as Fig. 5(a)] and linear fits to the values below the approximate error rate threshold of $p_0^{\text{all}} \simeq 1.5 \times 10^{-3}$ per round (solid lines).

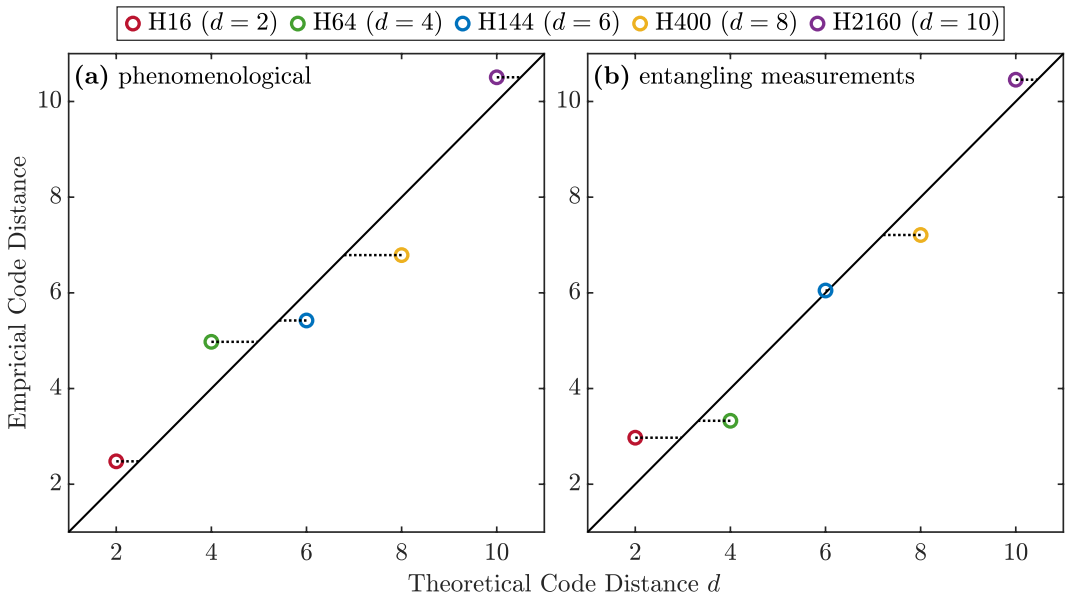


Figure 13: Empirical distances of hyperbolic codes under (a) phenomenological, and (b) entangling measurements error models. Theoretical code distances are defined as the lowest minimum weights of logical operators of a code (see Sec. 2). The empirical distance is proportional to the slope of a line fitted to the logical error rates of the codes below the approximate threshold value (see Figs. 11 and 12). If a code's data point is to the left (right) of the diagonal line with a slope equal to 1, then it is outperforming (underperforming) compared to the average of the code family. For example, H64 outperforms its theoretical code distance of $d = 4$ under the phenomenological error model but underperforms under the entangling measurements error model.

1. The closer a data point is to this line, the better the theoretical code distance predicts that code's logical performance.

Whenever a code's data-point falls to the left of the unit-slope solid line, the code is performing better in preserving logical qubits than predicted by its theoretical code distance. We anecdotally observe that while H16 and H2160 outperform their theoretical distances under both error models, and H144 and H400 underperform under both error models, the behavior of H64 changes with the error model. Specifically, H64 outperforms its theoretical code distance of $d = 4$ under the phenomenological error model, with an empirical code distance closer to that of H144 ($d = 6$), but underperforms under the entangling measurements error model, behaving similarly to H16 ($d = 2$).

D Honeycomb Floquet Codes from Symmetric Graphs

This Appendix includes some additional information about the honeycomb Floquet codes that are used as a baseline of comparison for the performance of hyperbolic Floquet codes in Sec. 4.

These honeycomb Floquet codes are based on symmetric trivalent graphs in Ref. [40] with a *girth* of six, which are closed hexagonal lattices tessellating the surface of a torus. This is unlike the usual method of constructing honeycomb codes which uses rectangular patches of a hexagonal lattice with periodic boundary conditions [30]. Turns out using symmetric graphs is beneficial is terms of number of physical qubits for a given code distance. For example, codes with code distances of 8, 12, 16, and 20 based on the construction in Ref. [30] have 96, 216, 384, and 600 physical qubits, respectively. While codes with the same distance based on symmetric graphs need 72, 162, 288, and 450 physical qubits, respectively. This corresponds to a 25 % save on the resources. Using symmetric graphs for non-Floquet hexagonal codes can also lead to reducing the number of physical qubits for a given distance and make the realization of higher-distance codes in the near future more feasible.

In Fig. 14 and Fig. 15, we present the performance of honeycomb Floquet codes under both of our error models and decoded by the MWPM decoder described in Sec. 3 and Appendix A. Honeycomb codes show threshold error rates around 0.45 % and 1.5 % under the phenomenological and entangling

measurements error models, respectively. Similar to the hyperbolic codes, the value of the threshold is higher for the entangling measurements error model.

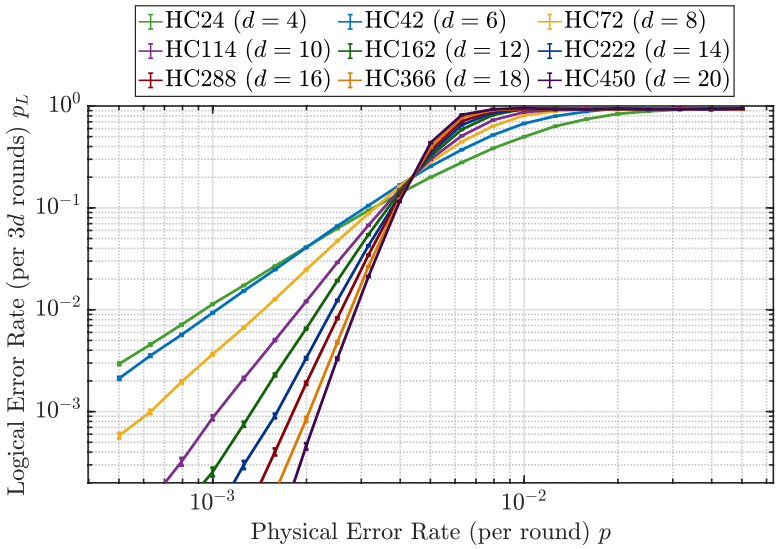


Figure 14: *Any logical operator* error rates for the family of honeycomb Floquet codes with the phenomenological error model.

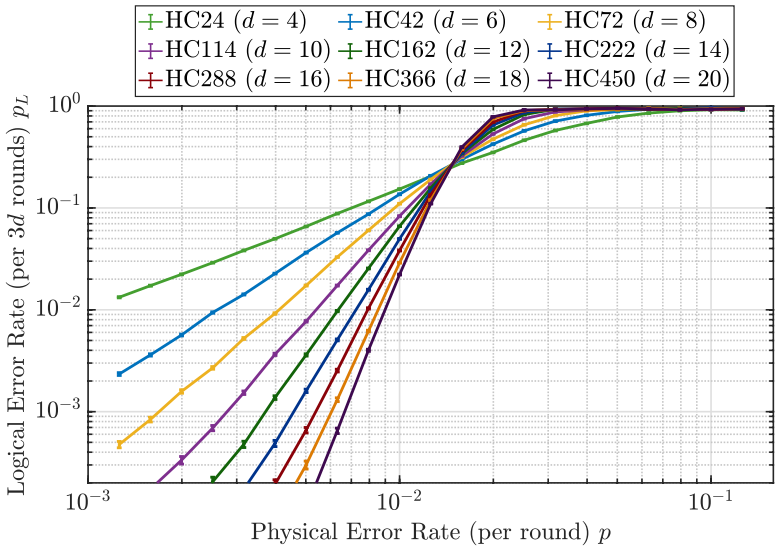


Figure 15: *Any logical operator* error rates for the family of honeycomb Floquet codes with the entangling measurements error model.