Comparative study of quantum error correction strategies for the heavy-hexagonal lattice

C. Benito,[1, *] E. López,[1] B. Peropadre,[2] and A. Bermudez[1, †]

[1]*Instituto de Física Teórica UAM-CSIC, Universidad Autónoma de Madrid, Cantoblanco, 28049, Madrid, Spain*
[2]*IBM Quantum, IBM Research, Cambridge, MA 02142, USA*

Topological quantum error correction is a milestone in the scaling roadmap of quantum computers, which targets circuits with trillions of gates that would allow running quantum algorithms for real-world problems. The square-lattice surface code has become the workhorse to address this challenge, as it poses milder requirements on current devices both in terms of required error rates and small local connectivities. In some platforms, however, the connectivities are kept even lower in order to minimise gate errors at the hardware level, which limits the error correcting codes that can be directly implemented on them. In this work, we make a comparative study of possible strategies to overcome this limitation for the heavy-hexagonal lattice, the architecture of current IBM superconducting quantum computers. We explore two complementary strategies: the search for an efficient embedding of the surface code into the heavy-hexagonal lattice, as well as the use of codes whose connectivity requirements are naturally tailored to this architecture, such as subsystem-type and Floquet codes. Using noise models of increased complexity, we assess the performance of these strategies for IBM devices in terms of their error thresholds and qubit footprints. An optimized SWAP-based embedding of the surface code is found to be the most promising strategy towards a near-term demonstration of quantum error correction advantage.

* cesar.benito@estudiante.uam.es
† Currently on sabbatical at Department of Physics, University of Oxford, Clarendon Laboratory, Parks Road, Oxford OX1 3PU, United Kingdom

I. INTRODUCTION TO QUANTUM ERROR CORRECTION (QEC)

Quantum computers hold promise for solving a variety of real-world problems with a clear-cut speedup with respect to their classical counterparts. Within the standard quantum circuit model [1], these problems are addressed through specific quantum algorithms that get decomposed into a sequence of primitive building blocks: product-state preparation, a sequence of unitary single- and two-qubit gates, and single-qubit projective measurements and resets. It is by carefully estimating the algorithmic resources in terms of the qubit number and basic operations required, together with possible additional costs of classical mid- or post-circuit processing, that one can make rigorous statements about the speedup and quantum advantage of a specific quantum algorithm [2, 3].

It is important to note that none of these primitive operations are perfect in practice, as qubits are never completely isolated from the environment and, additionally, the control techniques are subject to both systematic and stochastic errors [4]. Even if error rates p in leading quantum-computing platforms have currently reached values as low as $p \sim \mathcal{O}(10^{-3})$, the unavoidable accumulation of errors in quantum circuits composed of many of these operations still limits the available circuit depths. A rough estimate shows that such noisy circuits can allow for $\mathcal{O}(1/p) \approx 10^3$ operations before the quantum information gets totally corrupted. Moreover, errors can spread and proliferate if the circuits use nested two-qubit gates, leading to more severe limitations on the maximum number of noisy operations. As a consequence, and with the exception of certain tailored problems [5, 6], current quantum computers have not yet been able to demonstrate a practical quantum advantage in quantum algorithms with relevant real-world applications [2, 3]. Actually, even if the ex-

perimental error rates were to be lowered to the $p \approx 10^{-6}$ level, which is extremely challenging from a technological perspective, none of these quantum algorithms would still be at reach, as they typically require on the order of 10^7-10^{12} operations [3]. Hence, for the progress of quantum computing, it is essential to develop strategies to cope with errors.

Quantum error mitigation (QEM) [7–10] and quantum error suppression (QES) [11–14] are one two families of such strategies, a suite of low-overhead methods that correct for measurement bias or actively suppress idle/gate noise, respectively. QEM and QES techniques, and their combination, are allowing to improve the computational power of current noisy intermediate-scale quantum (NISQ) devices [15], and will likely be crucial in the demonstration of quantum advantage in various other more relevant problems, entering into the quantum utility era as evidenced by recent QEM-enhanced simulations of many-body systems [16]. In the long run, however, one will have to develop techniques that prevent the error propagation and spreading in larger devices and deeper circuits, limiting their overall accumulation to a desired target level. This would ultimately allow running reliable quantum algorithms with trillions of gates for real-world applications. A strategy that has the potential to achieve this goal is that of quantum error correction (QEC) [17–19], which works by encoding the information redundantly into logical qubits composed of multiple physical qubits. For that, specific QEC codes are used, which allow to actively detect and correct errors during a computation without distorting the encoded logical information. The redundant encoding, the error detection and correction, as well as the processing of the encoded information, involve a large overhead of operations that are themselves faulty. The threshold theorem ensures that, if the physical error rate of the primitive operations lies below a certain value p_{th} [20, 21], an arbitrarily-small logical error rate p_L can be achieved by increasing the redundancy of the QEC code to a certain required level. The value of the error threshold p_{th}, below which encoding increases the protection against noise, is code dependent and thus the central figure of merit of a QEC code. We are witnessing a fascinating time in which the quantum computer prototypes [22–24] are scaling up to the required sizes to go beyond NISQ circuits that operate on bare qubits, protecting and processing the information redundantly at the level of logical qubits. Still, the available number of qubits at present is, and will remain in the near and medium term, a limited resource. Hence, an important metric gauging the viability of a code, on which we will focus in this work, is the number of physical qubits $N(p, p_L)$ required to reach a target logical error p_L given a physical one $p < p_{th}$. This qubit number will be from now on referred to as the QEC footprint.

We note that the QEC footprint of a given strategy will also depend on the specific platform where it is to be implemented. Indeed, the physical mechanisms underlying the primitive operations required for QEC, as well as their main sources of noise, are strongly dependent on the experimental setup. This brings us to a related point, the microscopic noise will lead to an effective error model with much more structure than a single error rate p. For instance, high-fidelity two-qubit gates with characteristic error rates $p_{2q} \sim \mathcal{O}(10^{-3}$-
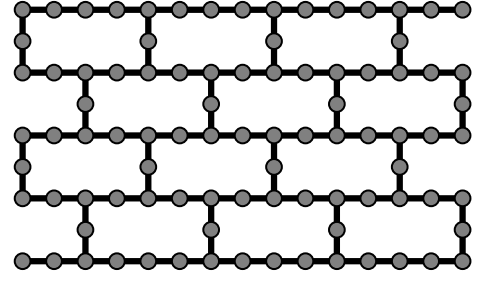

FIG. 1. **Heavy-hexagonal quantum computers:** Physical qubits are located at vertices and edges of a hexagonal lattice, here represented as grey circles in a brick-wall tiling of the plane. The edges (black lines) represent the available connectivity via two-qubit gates, corresponding to a nearest-neighbor graph with connectivities $z = 2$ ($z = 3$) for qubits at the edges (vertices) of the heavy-hexagonal lattice. Depending on the QEC strategy, different subsets of these qubits shall be used as data, syndrome, flag or bridge qubits.

10^{-4}) have already been achieved in the main platforms [25–30], while single-qubit error rates are typically much more accurate $p_{1q} \ll p_{2q}$. To provide more realistic predictions of the QEC footprint, one must thus consider a multi-parameter noise model with various error rates and, moreover, also consider the different effects that the errors may have for each of the primitive operations. Using realistic platform-dependent error models, together with advanced QEC decoders, is thus important for an accurate assessment of QEC strategies.

Finally, another important platform-dependent property relevant for QEC that mostly motivates our work is the connectivity of two-qubit gates. Trapped-ion [31–36] and Rydberg-atom [37–41] setups allow for shuttling physical qubits which, when combined with laser addressing techniques [42–46], can lead to programmable arbitrary connectivities. Instead, and in spite of promising ideas [47, 48], all current implementations of high-fidelity gates in superconducting quantum computers use static qubits arranged in a two-dimensional grid, and coupled via coplanar waveguides or tunable couplers that are typically restricted to nearest neighbors. This limits the qubit connectivities, from $z = 2$ in linear qubit arrays [25, 49, 50] to a combination of $z \in \{2, 3, 4\}$ for different planar grids [5, 51–55]. These reduced connectivities have motivated the choice of QEC strategies to the so-called topological codes [19], such as the Kitaev surface code [56–61], as the workhorse to fight against the errors in superconducting quantum computers. In topological codes, qubits are arranged in a planar lattice, and error correction only requires the measurement of local low-weight parity-checks known as stabilizers [62]. These codes display a much larger error threshold approaching $p_{th} \approx 1\%$ [63–65] with respect to that of concatenated QEC codes [20, 66, 67], making them very promising.

Superconducting devices based on either flux-tunable transmon qubits [54] or flux-tunable couplers [55] allow for controlled-phase entangling gates [1, 68], and have been used for implementations of the surface code with a $z = 4$ connectivity [54, 55, 69–71]. On the other hand, IBM's prototypes have been mostly based on fixed-frequency transmon qubits which, instead of using flux pulses, are subject to mi-

crowave drivings with certain target frequencies. These induce a cross-resonance controlled-not gate (CNOT [1]) when a pair of qubits is resonant with the drive [72, 73]. The high-fidelity regime $p_{2q} \sim \mathcal{O}(10^{-3})$ has been achieved in [73], which has also been used to demonstrate key operations in smaller prototype QEC codes [74–78]. However, spurious resonances with levels of neighboring qubits, or with higher-energy levels beyond the computational subspace, can give rise to crosstalk and leakage errors. In order to reduce them, current IBM devices use a small number of fixed qubit frequencies, limiting the practical connectivity to $z \in \{2,3\}$, and leading to a so-called heavy-hexagonal lattice [79] with superconducting qubits being placed at both edges and vertices (see Fig. 1). We note that recent devices based on the so-called Heron chip, such as the ibm_torino discussed below, have moved from cross-resonance gates to controlled-phase ones with flux-tunable couplers, although they maintain the $z \in \{2,3\}$ connectivity of the heavy-hexagonal lattice.

From the perspective of QEC, the ==heavy hexagonal architecture restricts the codes that can be directly embedded using the native low qubit connectivities.== We will analyze various possible strategies to circumvent this limitation. They can be organized into those that make use of *(i) SWAP gates* to adapt the surface-code stabilizer-measurement circuits, which in the standard implementation require $z = 4$, to the heavy-hexagon connectivity. In a related approach, the bridge qubits mediating the coupling between data and syndrome qubits, can be used as *(ii) flag qubits* [80, 81], measuring them and including the readout information in the QEC decoder. In the context of the heavy-hexagonal lattice, this approach has been exploited for instance in the recent works [82, 83]. Alternatively, one may consider *(iii) subsystem(-type) codes* in which (some of) the stabilisers are reduced to products of lower-weight operators requiring a reduced connectivity. This is the case of the heavy-hexagon code discussed in [79], which also leverages the use of flag qubits to map a Bacon-Shor-type subsystem code [84–86] to the heavy-hexagonal lattice. Finally, another group of strategies makes use of time-dynamic techniques such as *(iv) Floquet codes* on the honeycomb lattice [87–90]. This codes only require the reduced connectivity of the heavy-hexagon layout, reducing the syndrome extraction to a time-periodic measurement of weight-2 parity-checks.

Our work constitutes a thorough comparison of all these different strategies for QEC with heavy-hexagon quantum computers. We assess the prospects of the different QEC codes by presenting both the corresponding error thresholds, and QEC footprints both for the near and longer terms under a standard single-parameter noise model. In order to gauge the QEC capabilities for current IBM devices, we upgrade the noise model to a more realistic one, which consists of multi-parameter error channels extracted from actual characterization data of the IBM devices. In the following subsection, we summarise our main findings.

Code	Threshold
Surface code	
Square lattice (this work)	0.67(1)%
Square lattice (previous work [64]*)	0.60%
Heavy-hexagonal lattice, using flags (this work)	0.30(1)%
Heavy-hexagonal lattice using SWAPs (this work)	0.30(1)%
Floquet Honeycomb code	
Hexagonal lattice Z_L (this work)	0.357(8)%
Heavy-hexagonal lattice X_L (this work)	0.168(7)%
Heavy-hexagonal lattice Z_L (this work)	0.191(3)%
Heavy-hexagonal lattice (previous work [90]**)	0.2%-0.3%
Heavy-hexagon code	
Heavy-hexagonal lattice Z_L (this work)	0.27(2)%
Heavy-hexagonal lattice Z_L (previous work [79]***)	0.45%

*: Using a simplified matching graph for the decoding.
**: Using a different code aspect ratio.
***: Using smaller measurement/ reset errors by a 2/3 factor and a different matching graph.

TABLE I. **Calculated QEC thresholds:** We consider various QEC codes under a circuit-level noise model with a single error rate p, the SCL noise model of Eq. (6), and use a minimum-weight perfect matching decoder. For error thresholds previously reported in the literature, we comment on the slight differences regarding the error model, decoder, or a different arrangement of the qubits.

A. Organization and main results

We start by introducing the different QEC codes in Sec. II. Recently, it has been shown that the surface code can be adapted from $z = 4$ to $z = 3$ connectivity while keeping a comparable qubit footprint [83]. A standard technique to further lower the connectivity and cope with the heavy-hexagonal requirements, is to use some physical qubits as bridges in SWAP operations mediating the data-syndrome couplings. This solution may be naively discarded, as SWAP gates imply a threefold overhead when expressed in terms of CNOT gates and, more importantly, would typically be nested between neighboring bridge qubits leading to an uncontrolled spread of errors. We have found however that the SWAP-based embedding of the surface code into the heavy-hexagonal grid can be simplified to a large amount. Two different versions of syndrome extraction circuits are considered, where in one of them the bridge qubits are prepared in a reference state and subsequently used as flags. Remarkably, both SWAP- and flag-based embeddings lead to a QEC performance that is superior to other QEC strategies which were either especially-designed for this architecture, or can be directly implemented with the native connectivity: the flag-based heavy-hexagon code [79] and the Floquet honeycomb code [87–90], respectively. Some of technical aspects of the latter are discussed in App. A.

To gauge the performance of the codes, we perform large-scale Pauli-frame simulations of the noisy QEC circuits, and a minimum-weight perfect matching decoding under noise models of increasing complexity, which are all introduced in Sec. III. For the standard circuit-level noise model with a single error rate p for all operations, we obtain the error thresholds reported in Tab. I. For both surface code embed-

dings we find the very competitive value $p_{\text{th}} = 0.30\%$. This is larger than the Floquet honeycomb code thresholds, which are different for the two logical operators $p_{\text{th}}(Z_L) = 0.19\%$ and $p_{\text{th}}(X_L) = 0.17\%$ as explained in App. A. The difference becomes more drastic for the heavy-hexagon subsystem-type code, where we find $p_{\text{th}}(Z_L) = 0.26\%$, but there is no threshold for X_L, i.e. $p_{\text{th}}(X_L) = 0\%$. In Tab. I, we also quote the value of those thresholds that have been previously calculated in the literature, highlighting differences in the decoding, noise models, or qubit layouts that may cause slight deviations from those predictions. Further comparisons between codes on different qubit layouts can also be found in App. C.

We provide a more thorough assessment of the different codes in Sec. IV by reporting on their QEC footprints $N(p, p_L)$. This metric can be efficiently calculated even in the regime of small error rates using the new tools discussed in App. B. In particular, we report on the scaling qubit footprints that would be required to enter in the regime of *(i) QEC advantage*, which we define as a tenfold increase of the logical error rate with respect to current errors in the high-fidelity regime $p = 0.1\%$, or *(ii) QEC teraquop operations*, where circuits with trillions of transversal gates would be feasible. For the SWAP- and flag-based surface code embeddings, we find that the former requires a footprint of $N(0.1\%, 0.01\%) \approx 600$ qubits, whereas the later is $N(0.1\%, 10^{-10}\%) \approx 8000$ qubits under the standard circuit level noise model. Turning to a more realistic noise model, which assigns different weights to each of the primitive operations, we find that the SWAP-based surface code is superior when measurement noise dominates. When the error budget leans toward 2-qubit errors, the SWAP-based strategy gives way however to the flag-based approach, or even to the Floquet-based approach in the limit in which the entangling gates have a much larger error than the measurements and decoherence of idle qubits. In this way, one can predict which strategy should be preferred depending on foreseeable improvements on the different primitive operations.

Let us finally comment on our assessment of QEC strategies for current IBM devices, which require a more detailed study that includes hardware-specific details. Using experimentally-calibrated data for the `ibm_brisbane`, `ibm_sheerbroke` and `ibm_torino` devices, we estimate that current error rates must be reduced by a factor $\chi \sim 0.25$-0.45 in order to lie below the threshold of the best QEC strategy for IBM quantum computers: the SWAP-based embedding of the surface code. By considering a ten-fold improvement, i.e. $\chi = 0.1$, we find that the associated footprint for QEC advantage would be $N(\chi p, 0.01\%) \in \{1000, 300, 250\}$ for these three devices, respectively. The most promising one is `ibm_torino`, which operates the new Heron chip [91]. Even if this ten-fold improvement requires important technological advances, we believe that a demonstration of QEC-advantage with IBM devices, along the lines discussed in this work, is not a long-term target but lies in the intermediate or even the near term.
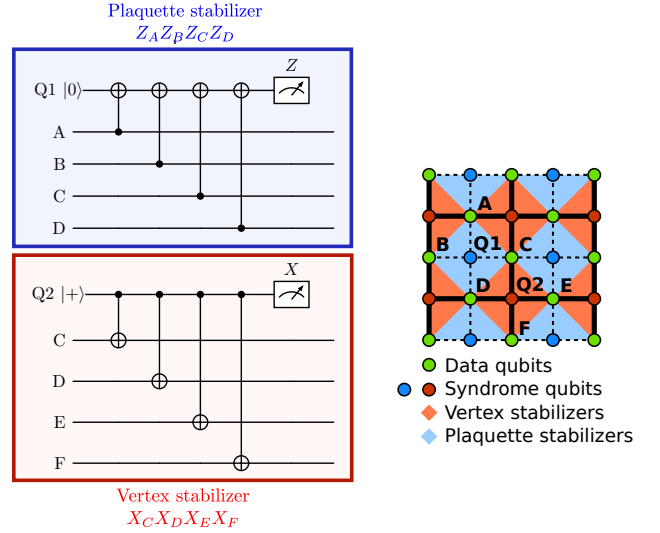

FIG. 2. **Surface code on a square lattice:** . On the right panel, the data qubits (green circles) are distributed along the edges of a square lattice (black lines), and the stabilizers correspond to the products of Z, X Pauli operators of the qubits belonging to the plaquette (blue) and vertex (red) areas, respectively. These stabilisers are measured simultaneously by mapping the information to bare ancillary qubits (red and blue circles) via syndrome-data couplings depicted with dashed lines. These couplings are implemented according to the circuits with 4 consecutive CNOTs between data and syndrome qubits, following the specific order depicted on the left panels, requiring 6 time steps to extract the error syndrome. Note that the plaquettes and vertices at the boundaries only involve 3 qubits.

II. QEC CODES ON THE HEAVY-HEXAGONAL LATTICE

We consider two different strategies to implement QEC in the heavy-hexagonal lattice. We first address the efficient embedding of the surface code in this low-connectivity architecture. We then move to QEC codes that by construction only require the connectivity of the heavy-hexagonal lattice: the heavy-hexagon code [79] and the Floquet code [87–90].

A. The surface code embeddings

The surface code is a local stabilizer code defined on a square lattice with qubits located on its edges [56, 58, 92]. Its stabilizer group S is generated by vertex and plaquette operators: weight-4 products of Pauli-X and Z operators respectively, which are reduced to weight-3 operators in the boundaries, all of which are depicted in Fig. 2. This code is a representative of topological QEC [19], and has been widely studied due to the relative simplicity of the stabilizer readout, and its overall good performance, having a threshold approaching $p_{\text{th}} \approx 1\%$ [63–65]. One of the simplifications of this code is that the standard syndrome extraction circuits, which is shown in Fig. 2, require a single ancilla qubit per stabiliser and a local connectivity of $z = 4$. This simplifies the implementation of rounds of QEC as compared to concatenated
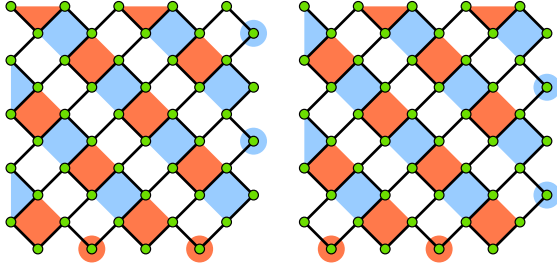
FIG. 3. **Surface-code embedding in a hexagonal lattice:** The green vertices and black edges define a hexagonal architecture, which is represented as a 45°-rotated brick-wall lattice. With this orientation, the qubits residing at the edges can be readily identified with those of the surface code (see Fig. 2), provided one can measure the corresponding plaquette (blue) and vertex (red) stabilizers. Depending on the boundary, stabilizers are formed by either 1 or 3 qubits. Dividing stabilizers in two halves corresponding to two sets of alternating diagonals (left an right panels) allows to measure them in parallel, while preserving the hexagonal $z = 3$ connectivity.

QEC codes [20, 21], which require a non-local connectivity and more resource-intensive syndrome-extraction procedures to achieve fault tolerance [93–95], leading to much smaller error thresholds [66, 67] and restricting the architectures on which they can be directly implemented.

Although the $z = 4$ connectivity is a great improvement with respect to concatenated QEC, there are specific platforms where the available connectivity is lower, and one must think about indirect strategies to embed the surface code. Recently, McEwen *et al.* [83] have proposed a simple and efficient method to embed the surface-code stabilizers in a hexagonal grid, measuring the stabilisers without additional ancillary qubits. As opposed to the standard approach in the surface-code readout (see Fig. 2), stabilizer measurements in the hexagonal lattice are split into two sub-rounds (see Fig. 3), in each of which all of the qubits are used as data qubits, and subsequent (un)folding and projective measurement steps are applied to them. The specific syndrome extraction circuit shown in Fig. 4 requires 5 time steps, differing from the 6 time steps required by the standard square-lattice readout circuits in Fig. 2. Following the indexing of the data qubits of Fig. 4, the CNOTs in the first two steps map the information of the weight-4 stabilizers to a single data qubit

$$Z_A Z_B Z_C Z_D \rightarrow Z_B Z_C \rightarrow Z_C \ ,$$
$$X_C X_D X_E X_F \rightarrow X_D X_E \rightarrow X_E \ . \tag{1}$$

which can be understood as two consecutive steps in which the stabilizer information is folded onto a single qubit. Then, this qubit is projectively measured in a third step, and the CNOTs in the final pair of steps revert (unfold) to the original situation. This method reduces the required qubit number by half, as it dispenses with the ancilla syndrome qubits. As a combination of these simplifications, the QEC footprint is lowered with respect to that of the square-lattice surface code [83]. As will be shown below in more detail, we have calculated the corresponding error syndrome, finding a value $p_{\text{th}} \approx 0.78\%$ that is higher than the square-lattice one $p_{\text{th}} \approx 0.69\%$ under
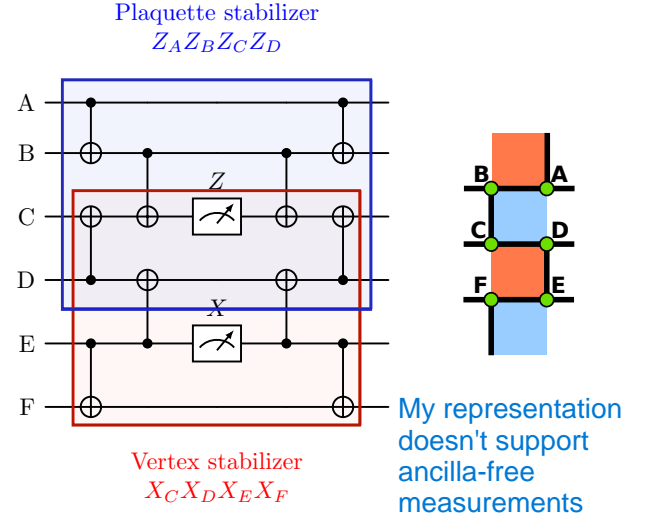

FIG. 4. **Ancilla-free readout of surface-code stabilizers on the hexagonal lattice:** On the right panel, we depict a plaquette-vertex pair along one of the diagonals, with data qubits labelled by $A, \cdots F$. On the left panel, we display the readout circuits for the plaquette (blue square) and vertex (red square), which require 5 steps, and can be done in parallel. Note that the first and last CNOTs are common for adjacent stabilizers (i.e. qubits C and D are shared between two simultaneously measured stabilizers), allowing the measurement of all stabilizers in a diagonal simultaneously.

the same error model (see Tab. I and App. C). Remarkably, even if only a sparser qubit connectivity $z = 4 \mapsto 3$ is available, this method shows that one can achieve unexpected improvements in surface-code embeddings.

Let us now consider an even lower reduction of the qubit connectivity $z = 3 \mapsto \{2, 3\}$, which corresponds to the heavy-hexagonal lattice with qubits on both the vertices and edges. Therefore, we need to adapt the above QEC strategy by coping with the additional qubits at every edge. In particular, the circuit of Fig. 4 requires a CNOT between qubit pairs A-B, C-D and E-F, each of which must be mediated by an additional bridge ancilla qubit that lies in between when considering the heavy-hexagon architecture. This can be achieved by inserting SWAP gates [1], implemented as three consecutive CNOTs. In the context of the heavy-hexagon architecture, we only require vertex-edge SWAP gates between neighboring qubits followed by a CNOT, which can be readily simplified into
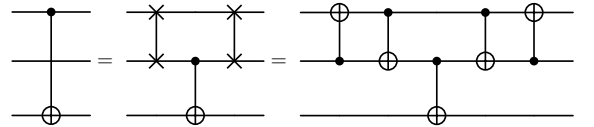
Note that, since the intermediate bridge qubit is not used to encode information, this SWAP circuit does not lead to the proliferation of errors in the data block, which would no longer be the case if one had to swap over further distances involving several data qubits.

In spite of these simplifications, the SWAP procedure still involves a 4 time step overhead, thus leading to an increased
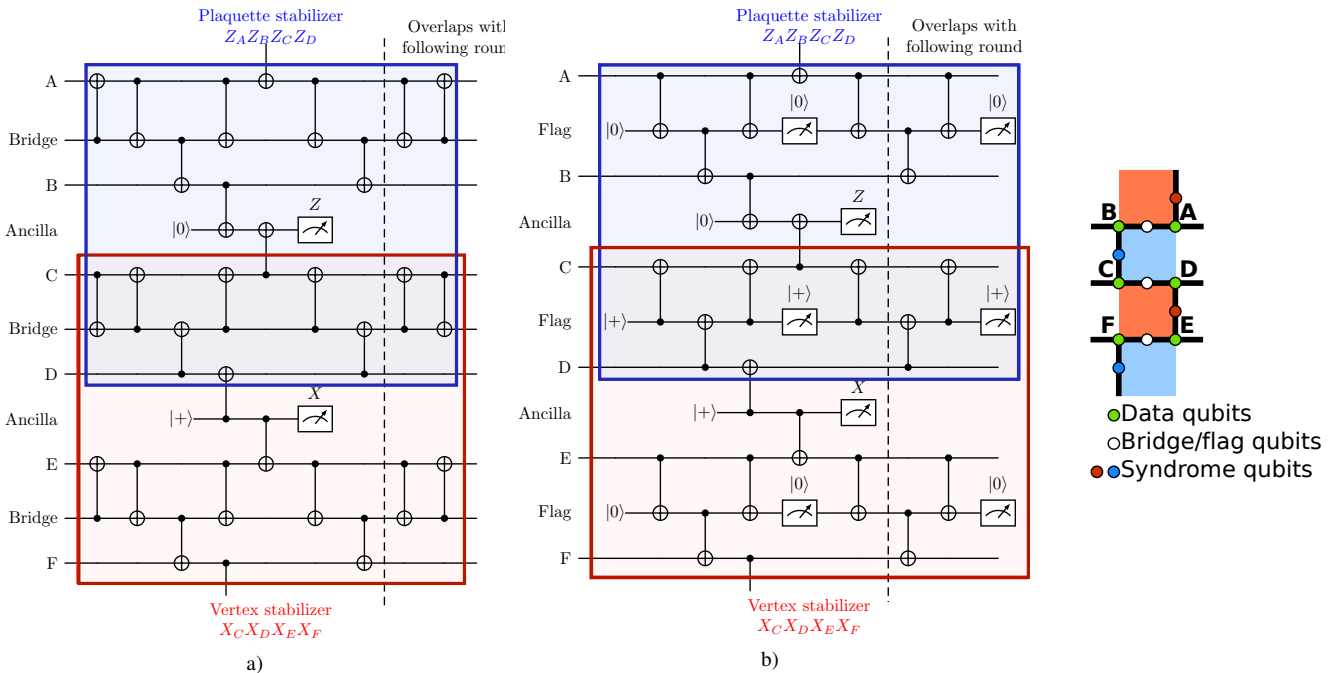
FIG. 5. **Swap- and flag-based readout of surface-code stabilizers on the heavy-hexagonal lattice:** On the rightmost panel, we depict the distribution of data qubits (green circles), and the remaining ancillary qubits for a neighboring plaquette-vertex of the heavy-hexagonal embedding. The ancilla qubits are divided into syndrome qubits (red and blue circles), and flag/bridge qubits (white circles). The circuits on the leftmost and central panels describe the required operations to measure the corresponding plaquette (blue rectangle) and vertex (red rectangle) stabilisers, for (a) the SWAP- and (b) flag-based approaches. The last time steps of one round can be executed simultaneously with the first time steps of the following, such that (a) requires 7 time steps per round, whereas (b) requires 6 time steps.

accumulation of errors. We note that, since the state of the intermediate bridge qubits does not need to be preserved, they can be initialized at the beginning of the SWAP-based transport to either $|0\rangle$ or $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$. We can also measure them after the SWAP and, in absence of noise, they should be preserved in the same Z or X eigenstate. In this case, since the bridge qubit is in a known initial and final state, one can dispense with the outer CNOTs, such that the SWAP operation is simplified to a single CNOT gate
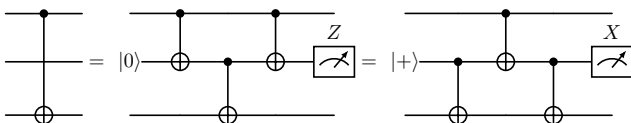
This not only reduces the overhead in the number of basic operations, but also allows for improving the error decoding and the subsequent error correction by considering the information from the bridge-qubit measurement. In a certain sense, the bridge qubit becomes a flag qubit that can be useful to identify further errors in the circuit, as will be discussed below. This idea underlies several recent adaptations of various QEC codes to reduced connectivities [79, 82, 83]. As noted for the pure SWAP-based approach, using flag qubits also has some drawbacks: we are introducing additional measurement and reset operations. These will take some time, during which the qubits that remain idle decohere and ac-

cumulate errors from their coupling to the environment. In this situation, if CNOT gates are less noisy than the flag measurements, depending on their overall contribution to the QEC code, it might be more convenient to adopt the previous SWAP approach.

Following these SWAP- and flag-based approaches, we now present two simplified embeddings of the surface code into the heavy-hexagonal lattice. We show the corresponding circuits in Fig. 5, which adapt the previous ancilla-free stabilizer measurements of the hexagonal lattice to the connectivity of the heavy-hexagon architecture. In particular, the parity-check measurement between qubits B-C and D-E after the first folding step of Fig. 4 have been converted back to the traditional fault-tolerant measurement circuits that use an ancilla qubit. We note that an adaptation of the toric code from the hexagonal to the heavy-hexagon lattice has also been discussed in [83], which also exploits flag and ancilla qubits for the measurements. Our proposed flag-based circuit in Fig. 5b provides a more efficient scheduling of the operations, saving a time step with respect to [83]. In turn, it consists of a total of 6 time steps and, thus, only adds a single additional time step with respect to the hexagonal-grid embedding of the surface code. Additionally, we have also taken into account the two-type of boundaries with either 1 or 3 qubits (see Fig. 3) in order to run the circuit in a planar lattice, considering heavy-hexagonal layouts that target the surface code rather than the toric code of [83]. Further benchmarking of different surface

code adaptations for the heavy-hexagons grid can be found in App. C.

Regarding the SWAP-based approach, it is interesting to see that the aforementioned CNOT-simplifications of the SWAP gates in this specific application lead to the circuit in Fig. 5a, which has a very similar complexity with respect to the flag-based approach, consisting only of 7 time steps. We thus remark that naively discarding SWAP-based methods for topological QEC in low-connectivity devices, guided by the $1 \mapsto 3$ SWAP-to-CNOT translation can be misleading. In fact, as advanced in Table I, both the flag- and SWAP-based approach display the same threshold for the single-rate noise model $p_{th} \approx 0.3\%$. We emphasise that this threshold is very competitive, as it is only roughly halved with respect to that of the square-lattice surface-code with $z = 4$ connectivity. Below, we shall provide a more detailed account of this threshold and the QEC footprint, comparing the various surface code embeddings to other QEC strategies, demonstrating the benefits of the present constructions.

B. Subsystem and Floquet codes

Let us now discuss a different strategy. Instead of adapting the stabilizer readout to embed the surface code on the heavy-hexagonal lattice, one may consider instead QEC approaches beyond stabilizer codes. The general idea behind the techniques we consider in this section is to reconstruct high-weight stabilizers from the measurement outcomes of lower-weight operators. This reduced weight, in turn, relaxes the connectivity requirements, an eases the integration into the heavy-hexagon lattice.

1. The heavy-hexagon code

The heavy-hexagon code [79] is a QEC code specifically designed for the heavy-hexagonal lattice (see Fig. 6), which falls in the category of subsystem codes [96, 97]. Subsystem codes are defined by starting from a $[n,k,d]$ parent stabilizer code with n data qubits, k logical qubits and a distance d. This construction extends the stabilizer group S into a larger gauge group G by including $k - l$ of the logical operators from the parent code, together with the stabilisers, in the generator set

$$G = \langle g_1, g_2, \ldots, g_{n-k}, \bar{X}_{l+1}, \bar{Z}_{l+1}, \ldots, \bar{X}_k, \bar{Z}_k \rangle . \quad (2)$$

The stabilizer group emerges now as the center of the gauge group, namely, the subgroup that commutes with all elements of G. Measuring the generators of the gauge group allows to determine all the stabilizers, and together with the remaining logical qubits, this defines a $[n,l,d]$ subsystem code. The advantage of subsystem codes comes from the fact that a different set of generators can be chosen for the gauge group, which allows us to reconstruct high-weight stabilizers from the product of lower-weight operators. Therefore, measurement circuits can be made simpler, less noisy and more convenient for architectures with a reduced qubit connectivity.
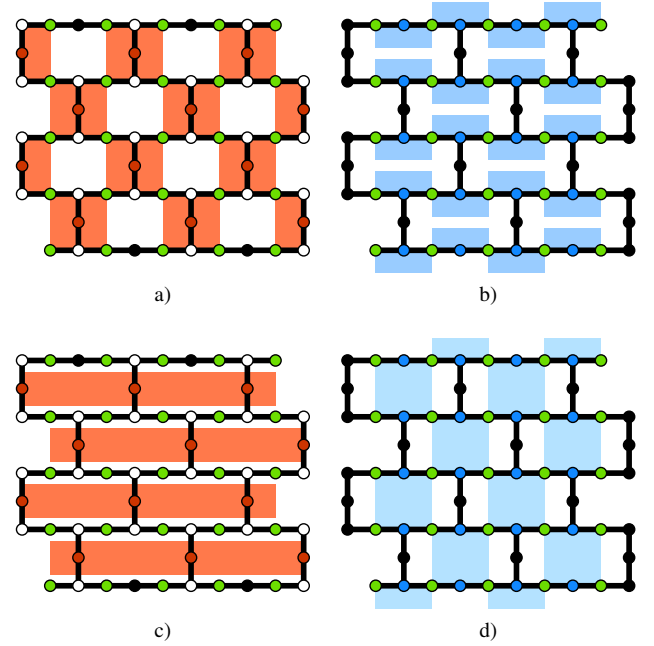

FIG. 6. **Heavy-hexagon code gauge and stabilizer operators:** Brick-wall representation of the heavy-hexagonal lattice using the same color convention as in previous figures. Additionally, some of the ancilla qubits are only used when measuring X (or Z) operators, and are thus depicted with black circles when not used. (a) *X*-type gauge operators (weight-4 operators in the bulk and weight-2 operators at the boundaries). (b) *Z*-type gauge operators (weight-2 parity checks). (c) *X*-type strip stabilizers, built by multiplying all X operators in the same row. (d) *Z*-type plaquette stabilizers, built from a single parity check at the boundary and two parity checks in the bulk.
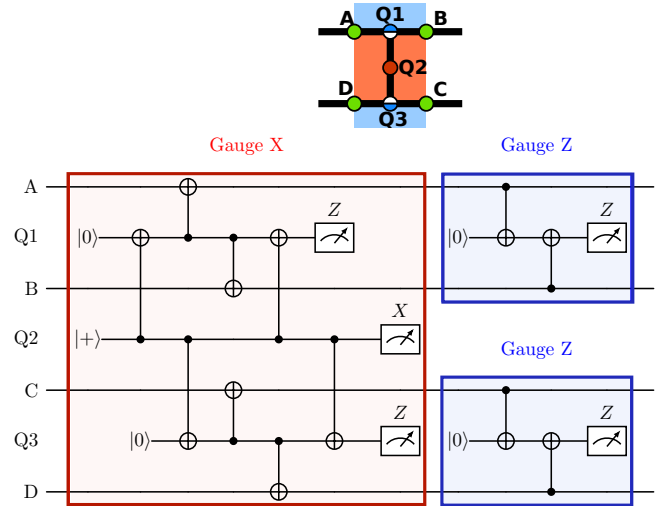
FIG. 7. **Gauge operator readout for the heavy-hexagon code:** In the upper panel, qubits A-D correspond to data qubits from the QEC code, and qubits Q1-Q3 are ancilla qubits used during gauge operator measurement, acting as flag or syndrome qubits. In the lower panel, we present the readout circuits for the *X*-type (red rectangle) and *Z*-type (blue rectangle) gauge operators, which can be measured sequentially in a total of 11 time steps.

The heavy-hexagon code has a $[d^2, \frac{1}{2}(d-1)^2+1, d]$ parent stabilizer code. Contrary to the previous embeddings of the surface code, the data qubits lie now on the horizontal edges of the heavy hexagonal lattice as depicted by the green circles of Fig. 6. The gauge group generators can be chosen to be purely of type X or Z. The Z-type generators are formed by products of simple weight-two parity checks (see Fig. 6b), whereas the X-type generators are weight-four in the bulk and by weight-two in the boundaries (see Fig. 6a). The stabilizer generators are accordingly of Z or X type. The former coincide with the Z-type gauge generators at the boundary, while they are built from products of two parity checks in the bulk (see Fig. 6d). The latter are given by the product of all X-type gauge generators in a horizontal row (see Fig. 6c). The heavy-hexagon code encodes a single logical qubit, with logical operations $\bar{X}$ and $\bar{Z}$ being the product of the corresponding Pauli operators along a row and a column respectively.

The ancillary qubits of the heavy hexagon code are also shown in Fig. 6. Part of them are used to to encode the gauge operator measurement outcomes, while others are used as flag qubits. We remark that since not all gauge operators commute among themselves, they cannot be measured simultaneously. The specific measurement circuit [79] is shown in Fig. 7. X-type gauge operators are measured first, and Z-type operators go afterwards, with a total of 11 time steps per round. The heavy-hexagon code provides an interesting solution with low logical errors for the heavy-hexagonal lattice for small code distances [79]. As advanced in Table I, we find a threshold for bit-flip errors of $p_{\text{th}} \approx 0.26\%$, which would be competitive with respect to the previous surface-code heavy-hexagonal embeddings. However, this subsystem code has an important drawback: the weight of its X-type stabilizers increases with the code distance, which prevents it from having a threshold when protecting against phase-flip errors. As occurs for the Bacon-Shor codes [84–86], under a realistic physical error model, there will be a maximum distance for which the corresponding logical error rate is minimized, but further increasing it causes the overall error to become larger. Consequently, it is likely not suitable for implementations of quantum algorithms unless the microscopic noise is highly biased towards bit-flip errors. Note however that there have been recent proposals to solve this issue for Bacon-Shor codes [98, 99].

2. *Floquet honeycomb code*

Similarly to subsystem codes, Floquet codes exploit a set of low-weight non-commuting operators to extract the value of higher-weight stabilizers. The crucial difference with respect to subsystem codes is that part of the stabilizer group, as well as the logical operators, change periodically in time. This interesting idea was recently introduced by Hastings and Haah in [88] for an hexagonal (honeycomb) lattice.

The honeycomb lattice is three-colorable: each hexagonal cell can be assigned an integer 0 (red), 1 (green) and 2 (blue), such that adjacent cells always have different colors. Edges are also given a color, which is inherited from the two cells they join, as depicted in Fig. 8. In the Floquet code, qubits
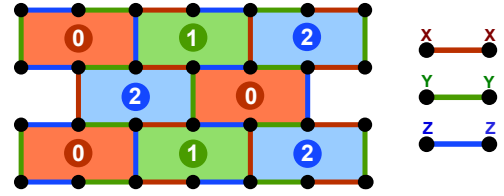

FIG. 8. **Floquet honeycomb code**. Qubits are located at the vertices of a brick-wall representation of the hexagonal lattice. The hexagonal cells are 3-colorable, being each color associated to a specific Pauli operator. Each edge is assigned a parity check operator, which shares the color of the cells it joins, and fixes the definition of the weight-2 parity checks.
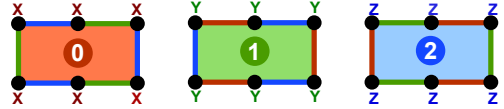

FIG. 9. **Stabilizers of the Floquet honeycomb code:** The product of all parity checks surrounding each cell defines a weight-6 plaquette stabilizer, which changes periodically in time.

are located at vertices of the lattice. The basic operators are weight-2 parity checks acting on all qubit pairs: one parity check per edge. Edge colors are identified with products of different Pauli operators: type 0, 1 and 2 are associated with XX, YY and ZZ respectively [89] (see Fig. 8) . The parity checks are measured sequentially according to their color following three sub-rounds.

Plaquette operators are defined as the product of the six Pauli operators of the corresponding color at the vertices of a cell (see Fig. 9). These operators coincide with the product of the parity checks around the cell, and thus their value can be reconstructed from the weight-2 parity check outcomes, which is similar in spirit to the previous subsystem codes. We also note that the plaquette operators commute among themselves, as well as with the parity checks. The stabilizer group at each round is thus generated by the plaquette operators, together with the parity checks measured at that round, which trivially commute with each other. As a consequence, part of the stabilizer group changes periodically with time. Additionally, also the logical operators have a temporal evolution. A more detailed description of the logical operators of the Floquet honeycomb code can be found in Appendix A.

The Floquet honeycomb code can show an improved error threshold in architectures for which the weight-2 parity-check can be measured directly instead of being decomposed into primitive operations. In particular, the reduced depth of the parity-check measurement circuit leads to an error threshold $p_{\text{th}} \approx 2\%$ [90], which provides a clear improvement with respect to that of the standard surface code $p_{\text{th}} \approx 0.6\%$ under the single error rate noise model [64]. In fact, it also surpasses the thresholds $p_{\text{th}} \approx 0.3\%$ - 0.7% of surface-code variants with native two-body measurements [100, 101]. This Floquet-based improvement relies on the assumption that the two-qubit parity measurement is performed in a single time step with the same error rate p as the single-qubit projec-

tive measurement. In the context of superconducting qubits, the single-qubit measurements are typically performed by a dispersive-coupling of the qubits to individual resonators, leading to a signal that is amplified and recorded for single-shot qubit readout [24]. Joint measurements can be achieved by coupling qubits to the same resonator [102], which can be exploited towards a direct parity-check readout [103, 104]. In spite of the remarkable progress, these collective measurements are still slower and noisier than the single-qubit ones, and it is thus not realistic to incorporate them in a circuit using the same readout error and readout time as in single-qubit measurements. Moreover, the effect of these measurement errors will be very different, including for instance an additional dephasing within the individual even and odd parity sectors. This error modelling would need to be considered in detail, and incorporated in Floquet-code QEC simulations to derive realistic estimates of the improvement of the threshold in actual superconducting devices.

For current superconducting-qubit computers with only single-qubit readout, Floquet codes are a priori less attractive under the following argument. Floquet codes can be expected to offer an advantage when splitting an operator into several lower-weight measurements is preferable over performing circuits with several CNOTs for the higher-weight stabiliser readout. However, this is not the case in measurement-error dominated architectures such as most of the current superconducting devices. On the other hand, one cannot directly rule them out for architectures with reduced connectivities such as IBM's heavy-hexagon devices, where the Floquet honeycomb code is naturally suited since the qubits at the edges can be used as ancillas for the parity-check measurements. The relative advantage or disadvantage with respect to other strategies will depend on the specific error model. As advanced in Table I, for a single-rate error model, we find a threshold of $p_{\rm th} \approx 0.19\%$, which is lower than for the alternative QEC strategies. In the section below, we will also present a detailed comparison of the near- and long-term QEC footprints, also considering a realistic noise model for IBM devices with variable noise weights to assess how improvements in different primitive operations can change the preferred QEC strategy.

C. Fault tolerance and code distance

Before delving into the numerical assessment of the performance of the different codes, let us discuss the role of fault tolerance in all these approaches, which is a desirable property of the stabilizer readout circuits. A circuit is said to be fault tolerant if it prevents low-weight errors from spreading into higher weight errors, which could potentially cause a logical error in a distance-d code in a situation in which less than d errors actually occurred [105]. Thus, a non fault-tolerant (FT) circuit design can reduce the "effective" distance of the QEC code, degrading its error correcting capabilities.

A way to achieve fault tolerance this is to build the circuit ensuring that a single error never spreads to more than one data qubit. An example of FT measurement is given by the parity-check readout of the Floquet honeycomb code using ancilla qubits. The circuit circuit employs CNOT gates, which spread X errors from the control to the target qubit, and Z errors in the reverse direction. The Z-type parity-check circuit with a single ancilla qubit
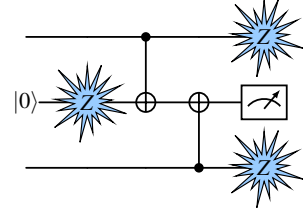
ensures that no individual error is propagated to two data qubits. Indeed, the only error that is propagated to two data qubits is harmless, as it corresponds to the measured operator, which is a Z-type weight-2 parity check.

An example of a non-FT readout is the ancilla-free circuit for the parity-check readout in the Floquet honeycomb code. which follows a similar philosophy to the (un)folding embedding scheme of the surface code to the heavy-hexagonal lattice presented above. In particular, the Floquet honeycomb code could be run directly in a hexagonal lattice with the following parity-checks readout circuit
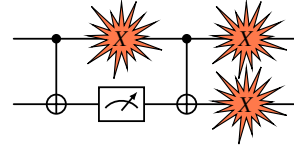

Switching to this ancilla-free scheme changes the error propagation. Note that single errors that propagate into two data qubits are dangerous due to their potential to reduce the minimum number of errors that constitute a logical error. This can be easily understood for a 3-qubit repetition code under bit-flip errors [19], which requires measuring the parities Z_1Z_2, Z_2Z_3. If a bit-flip error as shown in the circuit above occurs during the Z_1Z_2 parity check measurement, it shall spread into a weight-2 error X_1X_2 that is no longer captured by the Z_1Z_2 parity check, but rather by Z_2Z_3. Hence, after the decoding, the correction would be X_3, which altogether leads to a logical operation $X_L = X_1X_2X_3$. In this situation, a single error has sufficed to cause a logical error, so that the effective code distance is reduced from $d = 3 \mapsto 2$, and one can no longer correct single bit-flip errors. The situation is different for the previous ancilla-based parity-check readout, which is directly FT as the circuit does not spread dangerous 2-qubit errors.

For the Floquet honeycomb code, this non-FT parity check is known to reduce the effective code distance by half [90, 106]. One could expect that, since we are interested in the heavy-hexagonal lattice, which has a larger number of ancillary qubits at our disposal, switching to measurement with ancillas to preserve the full distance of the code would improve the QEC performance and reduce the required overheads. However, as shown in Appendix C, we observe the opposite effect. While restricting error propagation to a single data qubit is sufficient to ensure fault tolerance, sometimes it is too restrictive and can have an associated overhead in the

number of extra ancilla qubits and extra noisy gates that leads to an overall decrease in the QEC performance.

The clearest example where using circuits where single errors on data qubits only propagate to single errors on data qubits results in unnecessary complexity is that of the surface code. Here, error propagation to more than one data qubit is not always associated to a reduction in the effective code distance. In particular, a single error after the second CNOT gate of a weight-4 stabilizer readout propagates into two data qubits as follows
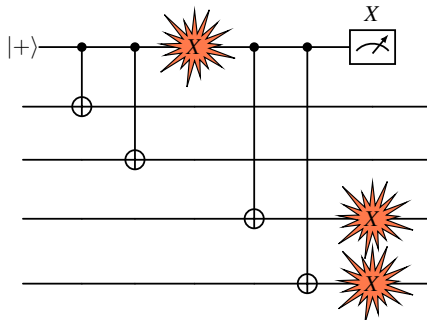
In spite of this error spreading, the syndrome extraction can be fully FT, preserving the full code distance, by a judicious scheduling of the CNOTs [60]. This is so because the spreading of the resulting two-qubit errors can be arranged along a certain direction of the full surface code that ensures that still d errors are required to generate a logical operator, thus preserving code distance (see Fig. 10). This is achieved by ordering the operations of stabilizer measurements, such that North, West, East and South data qubits are entangled to the ancilla qubit in that order, forcing the potentially-dangerous two-qubit errors to align always along the same diagonal direction. This has to be kept in mind when defining the boundaries for the code. This also influences the code performance, since the shape of the boundary affects the maximum code size that fits in a given processor, as we will see in Sec. V.

III. LARGE-SCALE CLIFFORD-CIRCUIT SIMULATIONS AND REALISTIC NOISE MODELS

The final goal of QEC is to reduce the failure rate of a quantum algorithm. This is parametrized by the logical error rate p_L, which determines the failure probability using a given QEC code under a noisy situation. Assessing the performance of a QEC code entails the evaluation of the logical error rate under different conditions. The process involves the noisy simulation of a quantum circuit where QEC rounds are inserted to extract the error syndrome. Then, syndrome information is fed to a decoder that applies error corrections, reducing error rates with respect to an encoded computation. In the following sections we describe the simulator, the noise model and the decoder, essential ingredients that lead to the calculation of p_L.

From the logical error rate we can construct important metrics describing the code efficiency. Thus, in order to provide more accurate predictions of the performance of QEC codes,
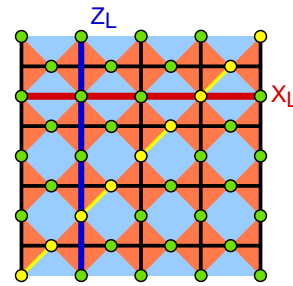

FIG. 10. **Single-ancilla FT readout in the surface code:** In its original formulation [57, 58], data quits are arranged on the edges of a square lattice, in this case corresponding to a $d = 5$ surface code with the logical Z_L, X_L operators depicted by the solid-blue and the dashed-red lines, respectively. Measuring the vertex stabilizers $S_v = \prod_{i \in v} X_i$ in a sequence of north, west, east south can lead to the spread of weight-2 phase errors depicted in yellow. In particular, for this scheduling, 5 weight-2 errors (in yellow) are required to flip any of the $d = 5$ surface code logical operators, ensuring that the full distance is preserved and the circuit is thus FT.

one must consider a multi-parameter noise model $p \mapsto \boldsymbol{p}$ with different error rates and, typically, also different effects of the errors on the qubits for each of the primitive operations. Using a realistic platform-dependent error model is therefore important for an accurate prediction of the QEC resources, which will also be discussed below considering current IBM quantum devices.

A. Pauli-frame simulations

In order to assess the performance of each of the QEC codes, one must simulate the corresponding noisy quantum circuits with a very large number of qubits, which cannot be performed by full wavefunction/density matrix circuit simulations on existing classical hardware. Fortunately, the QEC protocols discussed so far only require gates that belong to the Clifford group, the normalizer of the N-qubit Pauli group G_N, as all the circuits that have been presented only include CNOTs and projective single-qubit Pauli measurements.

Since Clifford gates map Pauli operators to Pauli operators, a noise model with Pauli errors can be propagated through the circuit very efficiently [107, 108] by updating their effects each time a gate is applied. If the simulation is restricted to Clifford gates and Pauli errors, the influence of noise is entirely characterized by tracking whether each qubit suffered X and Z errors, since Y errors are equal to the sequential combination of both X and Z errors, while two X or Z errors are equivalent to no error at all. This is the key idea of Stim's Pauli frame simulator [109], which allows for an efficient and fast simulation of noisy stabilizer circuits.

This is really convenient when analysing different Pauli-type noise models. After each Clifford gate or Pauli basis measurement, Pauli noise propagates according to the rules:

- Single-qubit Pauli gates do not affect error propagation.

- The Hadamard gate exchanges X and Z errors.

- The phase gate creates an additional Z error for each X error (i.e. it converts X errors to Y errors).

- The CNOT gate propagates X errors from control to target, and Z errors from target to control.

- Z measurements report the wrong result when an X error happened, and vice-versa.

Thus, noisy circuit simulations become really efficient as they only require $\mathcal{O}(N_G)$ operations for a circuit with N_G gates. A full stabilizer simulation requires $\mathcal{O}(N_G N_q)$ operations, with an additional overhead in the number of qubits: there are N_q stabilizers that must be updated after each operation. This reduces the performance for large registers as required for QEC.

B. Decoding matching strategy

In quantum systems, it is not possible to obtain a direct and univocal determination of the error that has actually happened. Instead, one has to measure a set of operators, whose outcomes constitute the error syndrome. From a given syndrome, it is the decoder task to apply an error recovery operation. There may be several errors that are compatible with the error syndrome, so the decoder should take into account their likelihood when choosing a specific recovery operation. In this work, we use a minimum-weight perfect matching algorithm, which efficiently decodes the error syndrome by finding the most-likely error, and recovering from it. In particular, we use PyMatching [110], for which Stim has a built-in integration.

The fundamental structure for matching algorithms is the so-called 'matching graph', which consists of a set of nodes and edges that join them. The decoder's task is to find the shortest path that visits a given subset of those nodes. Nodes on the matching graph correspond to 'detectors', a product of measurement outcomes which is deterministic in the absence of noise [109]. Detectors generalize the idea of stabilizer and flag measurements, since both produce deterministic results in the error-free case. A given error channel can either flip or leave unchanged a detector, regardless of whether it contains flag or syndrome information, which allows the decoder to treat them indistinctly.

Each detector has an associated 'detecting region' [83], which covers all possible error locations which can flip the value of the detector within the spacetime span of a circuit. In Fig. 11, we present a simple example of the detecting region of a circuit used in the syndrome extraction of the repetition code. We note that the analysis of detecting regions is interesting beyond the decoding, as it allows to find modifications of QEC codes that lead to new codes with improved performances [83, 111].

Coming back to the matching graph, we note that its edges correspond to individual errors. Two nodes are connected by an edge if their detectors are triggered by a given error mechanism. Edges can be weighted according to error probability of the related noise channel, such that the decoder returns the
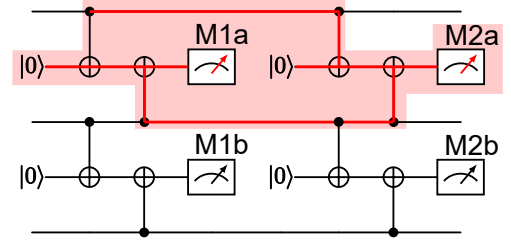

FIG. 11. **A detecting region for the repetition code:** In two subsequent measurements of the parity-check operators of a repetition code, we show all possible X error locations (red) that can flip the detector formed from the multiplication of the $M1a$ and $M2a$ measurement results. By considering subsequent measurements, one only needs to consider the relative changes of the stabilizer values, reducing the size of the detecting region with respect to the whole circuit, and simplifying the whole error syndrome extraction for larger codes.
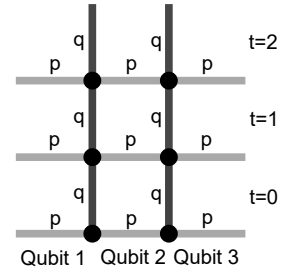

FIG. 12. **Matching graph for 3 rounds of the repetition code:** Nodes correspond to parity check measurements. Horizontal edges are identified with bit flip errors (with probability p) on data qubits, which trigger two adjacent detectors. Vertical edges represent measurement errors (with probability q), which flip the same detector twice in subsequent readout rounds.

minimum weight chain that visits exactly the nodes for which the corresponding detector has been triggered. A simple example of a matching graph is represented in Fig. 12, corresponding again to the repetition code. Stim and PyMatching generate automatically these matching graphs from the detecting regions for the more-involved QEC codes considered in this work.

C. Noise models: standard, variable-weight and Pauli-twirled biased errors at the circuit level

As advanced at the beginning of this section, accurate estimates of the QEC footprints require a realistic modelling of the microscopic noise that afflicts the circuits. For the efficiency reasons mentioned in the simulation section, we use a noise model that is based on Pauli error channels, while providing a close approximation to the microscopic errors of IBM's devices. Deviations from ideal operations will be thus represented by the action of Pauli channels

$$\mathscr{E}_{\mathrm{P}}(\rho) = \sum_{\alpha=0}^{D^2-1} p_\alpha E_\alpha \rho E_\alpha : \ \sum_\alpha p_\alpha = 1, \qquad (3)$$

where we have introduced

$$E_\alpha \in \{I, X, Y, Z\}^{\otimes^N} : \quad E_0 = I \otimes I \otimes \cdots \otimes I, \qquad (4)$$

and $D = 2^N$ for a channel acting on N qubits. For instance, a single qubit can be subject to X, Y or Z errors with probabilities p_x, p_y and p_z, such that $p_0 = 1 - p$ with $p = p_x + p_y + p_z$ is the probability that no error occurs. We further restrict our choice by considering only certain Pauli noise channels with specific rates for each of the basic operations. In particular, we make use of the following channels:

- *Bit- and phase-flip channels* correspond to single-qubit Pauli channels with either a bit-flip X ($p_x = p$, $p_y = p_z = 0$) or a phase-flip Z error ($p_z = p$, $p_x = p_y = 0$).

- *Biased dephasing channel* corresponds to a single-qubit Pauli channel having the same probability for X and Y errors, while Z errors are more likely [112, 113]. This model introduces an additional bias parameter $\eta > 0$, such that the Pauli error rates read

$$p_x = p_y = \frac{p}{2} \frac{1}{1+\eta}, \qquad p_z = p \frac{\eta}{1+\eta}. \qquad (5)$$

- *Single-qubit depolarizing channel* corresponds to a Pauli channel where all X, Y and Z errors are equiprobable with $p_x = p_y = p_z = p/3$. It can be seen as the $\eta = 1/2$ limit of the above biased channel.

- *Two-qubit depolarizing channel* corresponds to a Pauli channel by uniformly choosing between Pauli errors in the set $\{I, X, Y, Z\}^{\otimes 2} \setminus I \otimes I$. Each error has probability $p/15$, adding up to a total probability $p = 1 - p_0$.

We model noise at the circuit level by appending the previous Pauli error channels after each ideal quantum operation appearing in the circuit. For the numerical simulations, we use the following faulty gate set:

(i) Single-qubit gates: we consider that a qubit can be prepared, measured and controlled in any basis. In this way, all single qubit gates used in QEC can be absorbed into the rest of gates. Thus, we do not need to consider additional single qubit errors. This is justified when single qubit gates are much more accurate than the entangling and measurement operations, as motivated by current hardware.

(ii) Measurement operations: we allow for measurements in any basis (X, Y or Z). Before X measurements, we insert a phase-flip channel with probability p_m, and before Y or Z measurements, we insert a bit-flip error with the same probability p_m. This causes an error in the measurement outcome, as well as a flipped qubit after the measurement.

(iii) Reset operations: we allow for resetting a qubit in any basis. After X resets ($|+\rangle$), we consider a phase-flip channel with probability p_r, which causes the state $|-\rangle$ being prepared instead. After Y or Z resets, we insert a bit-flip channel such that an orthogonal state to the desired one is prepared.

(iv) Entangling gates: we consider CNOT gates followed by a two-qubit depolarizing channel with probability p_{2q}.

(v) Idling operations: qubits that remain idle during the application of gates/measurements/resets on other qubits are still subject to environmental noise. This is modelled by a depolarizing error channel with probability p_{id} inserted after an identity operation. The idling error probability depends on the duration of the time step, which is equal to the time of the longest operation taking place during the idle period. Therefore, there is a clear distinction between $p_{id,2q}$ for time steps involving CNOT gates and $p_{id,m}$ for measurement time steps. In IBM's transmon qubits, readout time is typically longer than CNOT time, which implies that when a CNOT is applied to some qubits while others are being measured, the former will also suffer an idle error that takes into account difference between gate times $\Delta p_{id} \approx p_{id,m} - p_{id,2q}$.

Using this noisy gate set, we have studied the QEC performance under three different noise models:

(i) Standard circuit-level (SCL) noise model: Also referred to sometimes as a uniform depolarising model, it has a single error rate for all operations

$$p_m = p_r = p_{2q} = p_{id} = p. \qquad (6)$$

This is an error model commonly used in the literature [65], as it allows for a quick comparison of different QEC codes.

(ii) Variable-weight circuit-level (VCL) noise model: Instead of considering that all error sources have equal rates, we take into account different weights for each of them. This allows us to explore the optimal noise regime for each QEC code. If we assign an error rate p to idle errors during the time it takes to apply a CNOT gate, the rest of errors can be expressed in relation to it, introducing relative α-weights

$$\begin{aligned} p_{id,2q} &= p, & p_{2q} &= \alpha_{2q} p, \\ p_{id,m} &= \alpha_{id,m} p, & p_m &= p_r = \alpha_m p. \end{aligned} \qquad (7)$$

Sweeping over the α-values allows us to assess how future changes in the error budget can affect the QEC footprints.

(iii) Pauli-twirled biased circuit-level (PBCL) noise model: As discussed in the introduction, the main motivation for developing codes for a heavy-hexagon lattice is that current IBM devices use this architecture to minimise frequency collisions, and thus improve the two-qubit cross-resonance CNOT fidelities. The PBCL noise model gives a more accurate description of noise in IBM devices. Using the Pauli twirling approximation [114–116], one can find the closest Pauli error channel that incorporates the effects of amplitude and phase-damping errors associated to decoherence of idle qubits. Given the decay times T_1 and T_2, twirling produces a biased dephasing (5) idling errors [116] with

$$p_{id}^X = p_{id}^Y = \frac{\tau}{4T_1}, \qquad p_{id}^Z = \frac{\tau}{2}\left(\frac{1}{T_2} - \frac{1}{2T_1}\right), \qquad (8)$$

or alternatively , using the parametrization of Eq. (5), as

$$\eta_{id} = \frac{T_1}{T_2} - \frac{1}{2}, \qquad p_{id} = \frac{\tau}{2}\left(\frac{1}{2T_1} + \frac{1}{T_2}\right). \qquad (9)$$

Here, τ corresponds to the CNOT execution time when calculating $p_{id,2q}$, or to the readout time for $p_{id,m}$.

We have extracted a set of parameters $\{\bar{p}_{2q}, \bar{p}_m, \bar{p}_r, T_1, T_2, \tau_{2q}, \tau_m\}$ from calibration data of several IBM superconducting chips as a reference for our noise model. Different sets are listed in Table II, together with the corresponding quantum device and the calibration date. The reference biased dephasing errors are characterized by Eq. (9) with

$$\bar{p}_{\mathrm{id},2q} = p_{\mathrm{id}}(\tau_{2q}, T_1, T_2) , \qquad \bar{p}_{\mathrm{id},m} = p_{\mathrm{id}}(\tau_m, T_1, T_2) . \quad (10)$$

We assume that ratios between the different errors remain invariant in the noise model, such that all channels have an error rate that is a fraction χ of the original, $p = \chi \bar{p}$. This is useful to study performance of QEC codes in less noisy processors, but preserving the same noise structure. We use the calibration data to compute the relative α-weights of the errors

$$\alpha_{\mathrm{id},m} = \frac{\bar{p}_{\mathrm{id},m}}{\bar{p}_{\mathrm{id},2q}} , \quad \alpha_{2q} = \frac{\bar{p}_{2q}}{\bar{p}_{\mathrm{id},2q}} , \quad \alpha_m = \frac{\bar{p}_m}{\bar{p}_{\mathrm{id},2q}} , \quad (11)$$

obtaining a similar model to the VCL noise model (7) with an additional bias parameter for idling errors: the PBCL noise model.

IV. ASSESSMENT OF HEAVY-HEXAGON QEC STRATEGIES

The increase in redundancy of the previous QEC codes clearly has an associated overhead in the number of physical qubits, which can be used to gauge the performance of different QEC strategies. As advanced in the introduction, one can estimate the number of physical qubits $N(p, p_L)$ that is required to reach a target logical error rate p_L when the device is characterised by a physical error rate p, which constitutes the QEC footprint.

The QEC footprint depends on the choice of the specific code and decoder, and can be used as a fair comparison of the QEC power of different strategies, both in the near and in the longer term. For instance, assuming that current error rates are $p = 10^{-3}$, $N(p, p_L)$ for $p_L = 10^{-4}$ is a reasonable resource estimate for near-term QEC-advantage footprint, allowing for a tenfold increase in circuit depth. On the other hand, $N(p, p_L)$ for $p_L = 10^{-12}$ is a resource estimate of the so-called teraquop footprint which, in the longer term, may allow to run practical quantum algorithms with one-in-a trillion faulty gates, paving the way for large-scale quantum computers. Evaluating these footprints requires the use of efficient large-scale simulation techniques discussed below.

Let us remark that $N(p, p_L)$ will also depend on the specific experimental platform where QEC is to be implemented including the particular sources of noise. In order to provide more accurate predictions of the QEC-advantage and the teraquop footprints, one must consider a multi-parameter noise model $p \mapsto \boldsymbol{p}$ with different error rates and, typically, also different effects of the errors on the qubits for each of the primitive operations, leading to $N(\boldsymbol{p}, p_L)$. Using a realistic platform-dependent error model is thus important for an accurate prediction of the QEC resources, which will also be discussed in the next section for current IBM quantum devices.
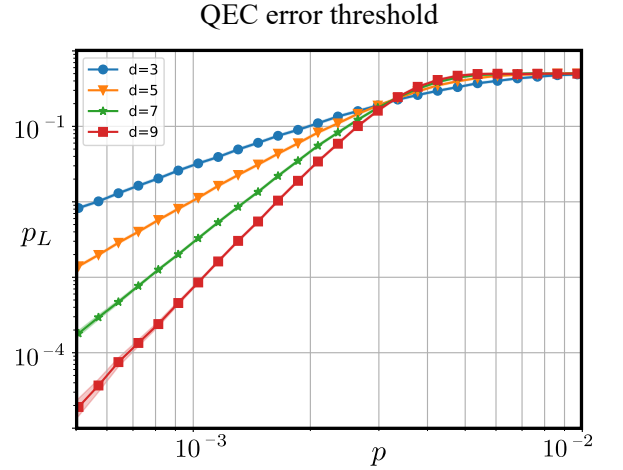

FIG. 13. **Error threshold for the SWAP-based heavy-hexagon embedding of the surface code:** Logical error rate p_L as a function of the single physical error rate p of the SCL noise model, using different sizes of the surface code adapted to the heavy-hexagonal lattice using SWAPs. The filled circles represent the numerical data, while the solid lines are a simple guide joining those points. Different code distances cross around $p = 3 \cdot 10^{-3}$, what determines the error threshold. Using $N_s = 10^7$, the Monte Carlo sampling error only becomes appreciable for low logical errors $p_L < 10^{-4}$.

A. Error thresholds and QEC footprints

When simulating noisy QEC rounds for a specific code with an increasing distance d, one finds that there is a crossing at a certain physical error rate p, which converges towards a single threshold value p_{th} for sufficiently-large code distances. Provided that the physical error rates are kept below threshold, the logical error rate p_L can be arbitrarily reduced by enlarging the size of the code, as stated by the threshold theorem for concatenated QEC [20, 21]. In Fig. 13, we depict this numerical calculation of the error threshold using the SCL noise model (6) for the SWAP-based embedding of the surface code into the heavy-hexagonal lattice, obtaining a crossing at $p_{\mathrm{th}} \approx 0.30\%$. In order to extract p_L, we performed Monte Carlo simulations of the noisy circuit using the aforementioned Pauli-frame formalism, dividing the logical error count N_{err} by the total number of circuit simulations N_s. In Table I of Sec. I A, we listed the thresholds for the remaining QEC codes introduced in the previous section that we have obtained using these methods.

As discussed in the introduction, a useful metric beyond these thresholds is the QEC footprint $N(p, p_L)$ when implementing a logical qubit. Below the threshold $p < p_{\mathrm{th}}$, the logical error rate decreases exponentially fast as one increases the code size and QEC can be advantageous. It is in this regime where it makes sense to quantify the QEC footprint $N(p, p_L)$ for a given target p_L. There are two independent factors that influence $N(p, p_L)$ for a given code: *(i)* the number of qubits $N(d)$ that is required to implement a distance d, and thus correct up to $(d-1)/2$, and *(ii)* the minimum dis-

tance $d(p, p_L)$ required to achieve a target logical error rate p_L under the physical error rate p. Therefore, the number of qubits can be calculated as $N(p, p_L) = N(d(p, p_L))$. The first factor $N(d)$ is entirely determined by the qubit layout. For example, a distance-d toric code with static qubits and local $z = 4$ connectivity uses $2d^2$ data qubits and $2d^2$ ancilla qubits for a total $N(d) = 4d^2$ qubit overhead. On the other hand, the embedding of the surface code into the heavy-hexagonal grid with $z \in \{2, 3\}$ presented in Sec. II A requires $N(d) = 5d^2$ qubits. The second factor $d(p, p_L)$ is related to the correcting properties of the code.

We note that, as one lowers the target logical error rate, the required system sizes increase and soon reach sizes that are prohibitively large for a full numerical simulation. Additionally, Monte Carlo simulations can be resource expensive: keeping the sampling errors small requires a sufficiently large number of faulty runs N_{err}, which for low p calls in turn for a very large number of shots N_s. In Appendix B we present a method to overcome these difficulties. Following Eq. (B2) [89, 117], we derive a fitting equation describing the dependence of the required minimal distance $d(p, p_L)$ with the physical and logical error rates

$$d(p, p_L) = \frac{\log p_L - a_0 \log p - b_0}{a_1 \log p + b_1} . \qquad (12)$$

In this formula, a_0, a_1, b_0 and b_1 are parameters obtained from linear fits, simulating different code distances and error rates. The above formula allows to straightforwardly estimate the QEC footprint $N(p, p_L)$. In the subsection below, we will see how the result of this prediction of the QEC footprint agree extremely well with the numerical results, justifying the validity of this approach, and allowing us to extrapolate efficiently to regimes with very small physical errors without a prohibitive Monte Carlo shot overhead.

Let us also note that for a multi-parameter noise model, such as the VCL (7) and the PBCL (8)-(10) noise models, one must consider that $p \mapsto \boldsymbol{p}$ in all of the above discussion. In general, it is no longer possible to find a single error threshold, as there are multiple sources of noise with different error rates that can affect the QEC in a convoluted and correlated way. One can nonetheless express all the error parameters in terms of a single error rate p by introducing α parameters as in Eqs. (7). Assuming that these α-values are kept constant as one modifies p, one can proceed analogously and get estimates for the QEC footprint $N(\boldsymbol{p}, p_L)$ assuming that the above fitting parameters will depend on the specific α values.

B. QEC footprint under standard circuit-level noise

We present now our results for the QEC-advantage and teraquop footprints of the different QEC codes designed for a low-connectivity heavy-hexagonal lattice. We start by considering the simple SCL noise model (6), which assigns a single error rate to all primitive operations. In Fig. 14, we see how the result of our prediction of the QEC footprint $N(p, p_L)$ (solid lines) agree extremely well with the solid circles, justifying the validity of our approach. Although not shown in this
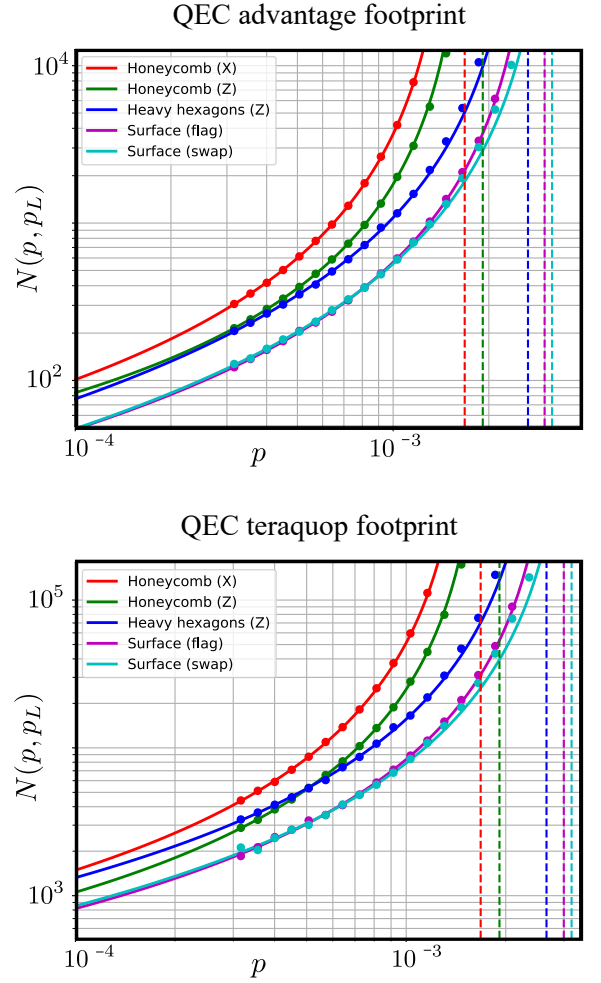

FIG. 14. **QEC footprint for the different codes:** Number of physical qubits required to preserve a logical qubit for d QEC rounds with a logical error rate lower than 10^{-4} (upper panel), and 10^{-12} (lower panel). We consider a noise at the circuit level according to the SCL noise model, and use a matching decoder.

figure, we can efficiently extrapolate to regimes with much smaller physical errors without the overhead of an increased Monte Carlo sampling. In this figure, we also display the error thresholds p_{th} of Table I as dashed vertical lines with the corresponding code color.

The upper panel of Fig. 14 shows the QEC-advantage footprints for a target logical error $p_L = 10^{-4}$, and thus corresponds to estimates of the requirements to achieve a tenfold-improvement in near-term QEC. We observe that the Floquet honeycomb code has the largest footprint. It is important to notice that vertical and horizontal operators in the honeycomb code have different structures, as shown in Fig. 20. As a result, the honeycomb code has different effective code distances for the X_L and Z_L operators, inducing different footprints. See Appendix A for details. The next code in terms of QEC footprint is the heavy-hexagon code, which roughly has a two and four-fold reduction for $p = 10^{-3}$ with respect to the honey-

comb Z and X, respectively. However, one should keep in mind that the Floquet honeycomb code is also capable to arbitrarily reduce errors that affect the X logical operator, while the heavy-hexagon code only has a threshold for Z_L. Hence, for even higher error suppressions, the Floquet honeycomb code will be a better alternative than the heavy-hexagon code, as it is capable of arbitrarily reducing all logical errors.

For the SCL noise model, our heavy-hexagon embeddings of the surface code, either the one that uses flag qubits or SWAP operations, display the best overall performance. Comparing both surface code variants, we see that using flags or SWAPs makes almost no difference under this noise model. Usually, SWAP gates degrade performance favoring the usage of flag qubits, but due to the CNOT gate cancellations and the circuit simplification in our heavy-hexagon QEC context, both methods become comparable and lead to similar QEC-advantage footprints. Which option is more efficient will depend on finer properties of microscopic noise of the device. In particular, for $p = 10^{-3}$, the QEC-advantage footprint for these two QEC strategies is $N(p, p_L) \approx 600$ physical qubits per logical qubit for a tenfold increase in circuit depth, outperforming the other strategies that require thousands of qubits.

In the lower panel of Fig. 14, we depict the teraquop footprints targeting a logical error rate of $p_L = 10^{-12}$. The trend is very similar, in the sense that the Floquet honeycomb and the heavy-hexagon codes have a worse performance. In this case, the Floquet honeycomb code actually requires a lower QEC footprint than the heavy-hexagon code when the physical error rate is well below the threshold. In any case, both the flag- and SWAP-based methods offer the best performance. At current error rates of $p = 10^{-3}$, one would need roughly $N(p, p_L) \approx 8000$ qubits to lower the logical error rates to the 10^{-12} level. If the physical error rates are reduced to $p = 10^{-4}$, $N(p, p_L) \approx 800$ can suffice to achieve that error suppression. This shows the typical interplay, requiring that hardware efforts are not only focused on scaling up the number of qubits, but should also improve on the physical error rates of the primitive operations, such that one lies as further apart from the threshold as possible.

C. QEC performance under variable noise weights

In the previous subsection, we have seen that both the SWAP- and flag-based approaches show a similar QEC performance, showing a clear advantage with respect to other strategies. This comparison was performed for the SCL noise model, and may change when considering a more flexible modelling that weighs the errors of the various primitive operations differently. In order to explore this possibility, we consider the VCL noise model and estimate how the thresholds change as a function of α_{2q}, α_m and $\alpha_{id,m}$, the quotients between the weights of the operations defined in Eq. (7).

In Fig. 15a, we show a contour plot of the error threshold as function of the variable weights of CNOT and measurement errors, α_{2q}, α_m. The quotient between idle errors has been fixed to $\alpha_{id,m} = 1$, namely $p_{id,2q} = p_{id,m} = p_{id}$. The idea behind these variations is that both software and hardware developments may improve differently the various primitive operations. In order to describe the thresholds, we have to chose one of the primitive operations as a reference. The threshold for subfigure (a) is defined with respect to idle errors, such that encoding is advantageous for $p_{id} = p < p_{th}^{(a)}$. This definition helps us to visualize the resilience of the different codes to CNOT and measurement errors. In particular, we observe that the SWAP-based embedding of the surface code is fairly resilient to measurement errors, while its performance gets degraded by CNOT errors. The flag-based strategy also shows a clear, although not as marked protection against measurement errors. On the contrary, the Floquet honeycomb code displays a similar performance under both measurement and CNOT errors. The SWAP- and flag-based embedding of the surface code into the heavy-hexagonal lattice have in general the higher error thresholds. However, when CNOT errors dominate within the VCL noise model (7), the Floquet honeycomb code can show a better performance than the surface-code variants, which is a consequence of the lower circuit depth of the parity-check circuits required by this code. In the figure, we have shadowed the parameter region where the performance of a given code is surpassed by the other.

We present a complementary plot of the same results in Fig. 15b, considering in this case an alternative approach to characterize the threshold in a multi-parameter noise model. We now take as a reference the dominant error, such that the code is effective if all errors are kept below threshold $p_x < p_{th}^{(b)}$, with $x \in \{2q, m, id\}$. The thresholds from both figures are related by the following equation

$$p_{th}^{(b)} = \max \left\{ \alpha_{2q}, \alpha_m \right\} p_{th}^{(a)} . \tag{13}$$

Using this approach, it becomes easier to assess how a reduction in the idle errors relative to the measurement and CNOT gate errors would affect the QEC performance. We note that this can be achieved by applying advanced dynamical-decoupling sequences [118] to the idle qubits all along the QEC algorithm, which can be applied to every physical gate in the circuit to protect the bare qubits [119] or else act at the level of the FT building blocks of the circuit to protect the encoded syndrome subspaces [120]. Dynamical decoupling is one of the main techniques of the QES approach mentioned in the introduction. Keeping a constant ratio p_m/p_{2q} defines straight diagonal lines in the logarithmic plot 15b, along which the idle errors increase from the left to the upper boundary. We observe that the influence of equally balanced idle errors is rather mild. In particular, it is minimal for the surface code embeddings in the measurement-error-dominated regime, which is the usual regime in most current experimental platforms. This is to be expected because there are few locations in their syndrome extraction circuits where qubits are idle (see Fig. 5).

Up to this point we have considered an identical idling error for CNOT and measurement gates, i.e. $\alpha_{id,m} = 1$. However, this assumption is not realistic for most current devices (see e.g. Table II for IBM devices), which typically have measurements and resets that take a much longer time than CNOT gates, such that the corresponding idling errors can
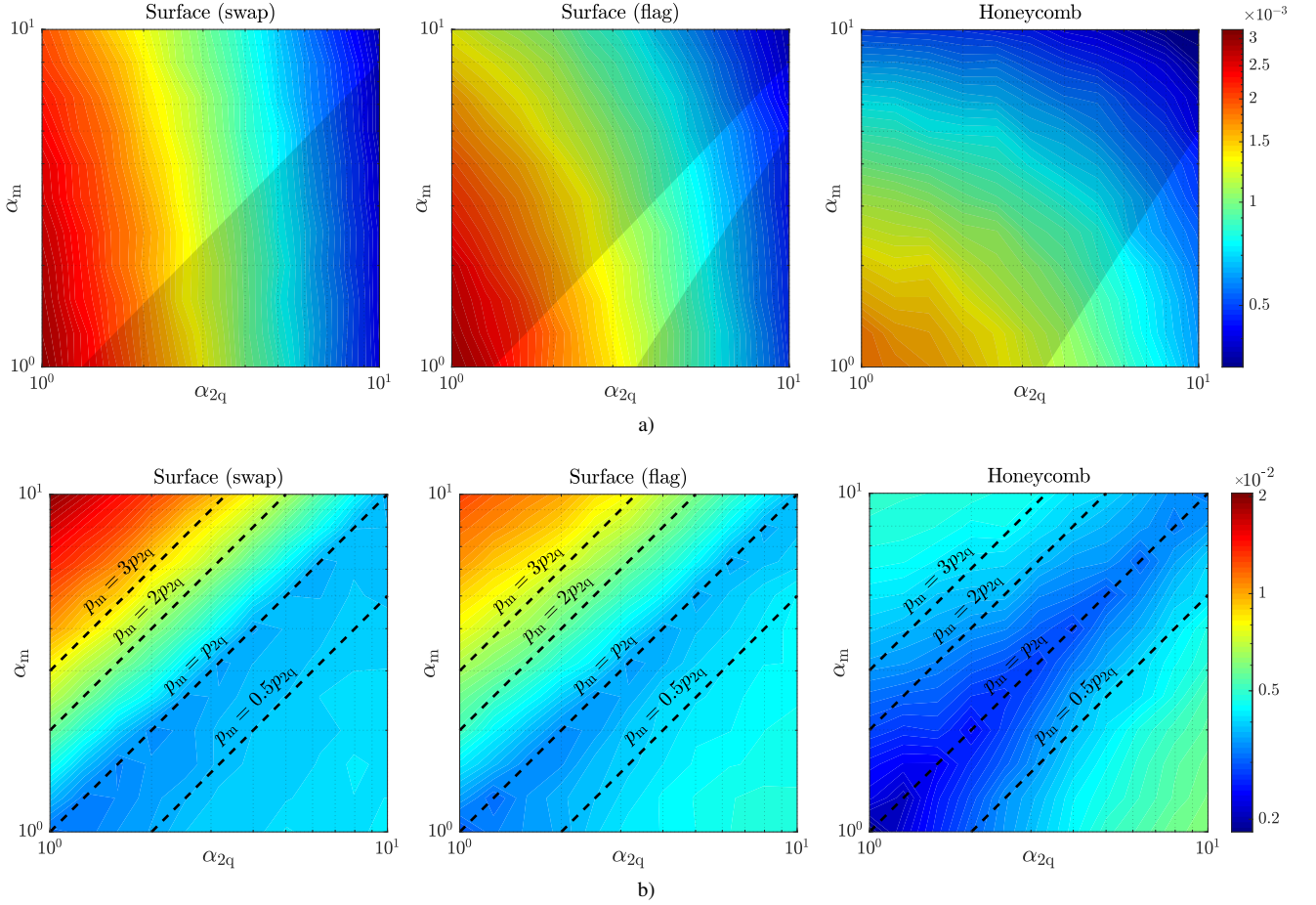
FIG. 15. **QEC thresholds for variable CNOT and measurement error weights:** Error thresholds when varying weights of CNOT and measurement errors with respect to idling errors, with $\alpha_{\text{id,m}} = 1$, in logarithmic scale. (a) Threshold with respect to idle errors. (b) Threshold with respect to the heaviest error type. (left panels) SWAP-based surface code on the heavy-hexagonal lattice, (middle panels) flag-based surface code on the heavy-hexagonal lattice, and (right panels) Floquet honeycomb code. The Floquet honeycomb code is balanced, acting similarly against measurement and CNOT errors, while the surface codes are more robust against measurement errors, primarily the SWAP-gate adaptation. The lighter regions within each contour plot in (a) delimit the areas where the corresponding code is superior to the other two. Diagonal lines in (b) correspond to variable idling errors with constant CNOT and measurement errors. It can be seen that idling errors have little influence in code performance.

Computer	$\bar{p}_{2q}$	$\bar{p}_m = \bar{p}_r$	T_1	T_2	τ_{2q}	τ_m	$\bar{p}_{\text{id,2q}}$	$\bar{p}_{\text{id,m}}$
ibm_sherbroke (2023-11-13)	$8 \cdot 10^{-3}$	$1.1 \cdot 10^{-2}$	$270\,\mu\text{s}$	$185\,\mu\text{s}$	$533\,\text{ns}$	$1244\,\text{ns}$	$1.9 \cdot 10^{-3}$	$4.5 \cdot 10^{-3}$
ibm_brisbane (2023-12-04)	$7.9 \cdot 10^{-3}$	$1.3 \cdot 10^{-2}$	$227\,\mu\text{s}$	$144\,\mu\text{s}$	$660\,\text{ns}$	$4000\,\text{ns}$	$3.0 \cdot 10^{-3}$	$1.8 \cdot 10^{-2}$
ibm_torino (2023-12-04)	$3.6 \cdot 10^{-3}$	$1.89 \cdot 10^{-2}$	$177\,\mu\text{s}$	$142\,\mu\text{s}$	$124\,\text{ns}$	$1560\,\text{ns}$	$6.1 \cdot 10^{-4}$	$7.7 \cdot 10^{-3}$

TABLE II. Noise calibration data for current IBM quantum computers used to build our realistic PBCL noise model of noise at the circuit level. Note that we are assuming that single-qubit gates have a negligible error, so that the error rate of CNOT gates can be directly obtained from calibration data of either cross-resonance or conditional-phase gates. $\bar{p}_{\text{id,2q}}$ and $\bar{p}_{\text{id,m}}$ are calculated using Eq. 10.

become comparable to that of p_{2q} and p_m. In our simulations, the main effect of longer readout times is that an additional idle error of magnitude $\Delta p_{\text{id}} = p_{\text{id,m}} - p_{\text{id,2q}}$ would be inserted in qubits where a CNOT is being applied while another qubit is being measured or undergoes a reset. For the surface code embeddings, this affects many CNOTs that are executed simultaneously with an ancilla qubit measurement or reset. Therefore, Δp_{id} may have a large impact that must be

carefully modelled going beyond the previous balanced idle errors. We have thus explored the $\alpha_{\text{id,m}} > 1$ regime. Guided by the calibration data of Table II, we set $\alpha_{2q} = 5$ and sweep over $\alpha_m \in [\alpha_{2q}, 10\alpha_{2q}] = [5, 50]$ and $\alpha_{\text{id,m}} \in [1, 25]$ in Fig. 16. We focus on the SWAP-based embedding of the surface code since it offers the best performance in this range of parameters. We observe that reducing the larger idle errors during a measurement or reset can lead to important improvements
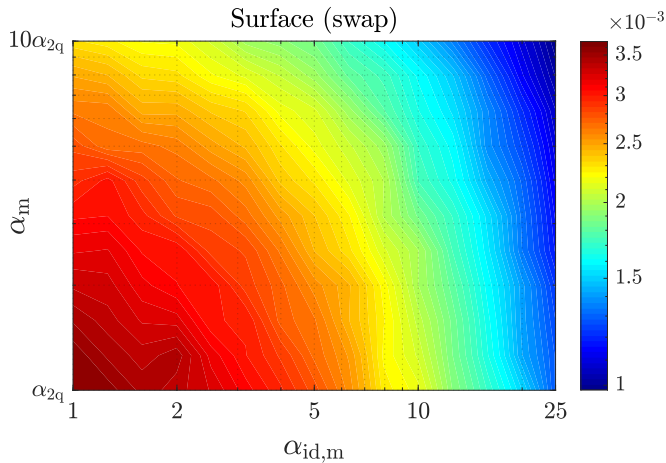
FIG. 16. **Surface code threshold for variable measurement error weights:** error threshold with respect to CNOT errors, using variable measurement idle $\alpha_{\text{id,m}}$ and readout errors α_{m}, with $\alpha_{2q} = 5$. A reduction in measurement idle errors can increase code performance.

on the code threshold. Thus, code performance would benefit from QES techniques during these long idle periods.

V. ASSESSMENT OF QEC FOR IBM DEVICES

In order to have a more accurate estimate of the QEC footprint for IBM devices, we now focus on the aforementioned PBCL noise model (8)-(10). In particular, we use the real calibration data from three IBM computers: `ibm_sherbroke` and `ibm_brisbane`, with 127 qubits and running on the Eagle chip, and `ibm_torino`, with 133 qubits and operating the new chip Heron. The specific calibration data we consider are listed in table II. We recall that the CNOT gate is not native to these devices. Since we are assuming that single-qubit gates have a negligible error, we directly read $\bar{p}_{2q}$ from the error of their native entangling gates.

We now calculate the QEC footprints for the three QEC codes that had a better performance with the simple noise model and a threshold for both bit- and phase-flip errors. As the current system sizes are still far away to consider the ter-aquop regime, we focus on estimating the requirements to show QEC-advantage in the near term. We thus set the target logical error to $p_L = 10^{-4}$, and display the QEC footprint $N(\boldsymbol{p}, p_L)$. For the current multi-parameter error rates, we find that none of the devices can achieve the desired regime of QEC-advantage in spite of an arbitrary scaling of the number of qubits. However, using our approach, we can predict what specific hardware improvements would be required in order to reach QEC-advantage. We introduce a $\chi < 1$ improvement of the error rate $p = \chi \bar{p}$, and re-scale all of the individual parameters with the current α-values of Eq. (11) extracted from IBM-device calibration data. For instance, a $\chi = 0.1$ implies a tenfold reduction of all error rates.

In Fig. 17, we present the QEC-advantage footprint as a function of this noise ratio χ. Contrary to the oversimpli-

fied SCL noise model, we find that all three devices show differences between SWAP- and flag-based embeddings of the surface code into the heavy-hexagonal lattice. For the `ibm_sherbroke`-based noise model, measurement and CNOT errors are almost in pair, so that the three QEC codes have a similar performance. For the `ibm_brisbane` computer, on the other hand, idle errors during a measurement $p_{\text{m,idle}}$ are more important. This degrades the performance of the surface code with flags, and it becomes comparable to the Floquet honeycomb code. Finally, for the `ibm_torino` computer, we find that it has significantly lower CNOT errors, which implies that the additional CNOTs required for the SWAP-based surface code do not introduce too much noise, so that this code significantly outperforms the other two. For all three IBM devices, the surface code with SWAPs discussed in Sec. II A has the best performance.

We emphasise again however that the current error rates on any of the three IBM devices are still above the threshold for this code. All noise sources must be reduced to a $\chi \sim 0.25$-0.45 of their current values to be below threshold (i.e. from a two- to a four-fold improvement). By considering a ten-fold improvement $\chi = 0.1$, we find that the SWAP-based surface code requires an QEC footprint of $N(\chi \boldsymbol{p}, p_L) \in \{1000, 300, 250\}$ qubits to show QEC advantage in the `ibm_brisbane`, `ibm_sherbroke`, and `ibm_torino` devices, respectively. Under this improvement, the devices would be dominated by a $p \sim 0.1\%$ measurement/reset error rate, and the QEC routines could attain a target $p_L = 0.01\%$, showing a tenfold improvement and thus QEC advantage of a logical qubit memory versus an un-encoded one. Even if these improvements will require important technological advances, these results show that a demonstration of QEC-advantage with the proposed SWAP-based surface-code embedding is not a long-term target, but at reach of near and intermediate advances in IBM hardware.

Let us now consider an additional argument that may be important when choosing a QEC code for the heavy-hexagonal lattice in near-term devices in which qubit number is still an important limitation. Even if the surface-code embeddings have a superior performance, the Floquet honeycomb code can leverage the existing resources in a more optimal manner. Note that surface-code patches have to be cut with a 45° angle with respect to heavy-hexagonal "bricks", which is required to preserve fault-tolerance in the stabiliser readout circuits, as described in Sec. II C. Therefore, a "rotated" surface-code patch would waste a big number of qubits, reducing in this way the maximum distance that can be implemented with a given "un-rotated" heavy-hexagonal device. On the other hand, the Floquet honeycomb code can be defined with un-rotated boundaries, making a more efficient use of qubit resources. Therefore, it should be borne in mind that the surface code can have an additional overhead when implementing it on un-rotated heavy-hexagonal lattices, virtually doubling the number of physical qubits due to spare qubits outside code's boundaries, as illustrated in Fig. 18. If we take into account those spare qubits in the QEC footprint, the honeycomb code would have a better performance with respect to any of the surface-code embeddings for the `ibm_sherbroke`
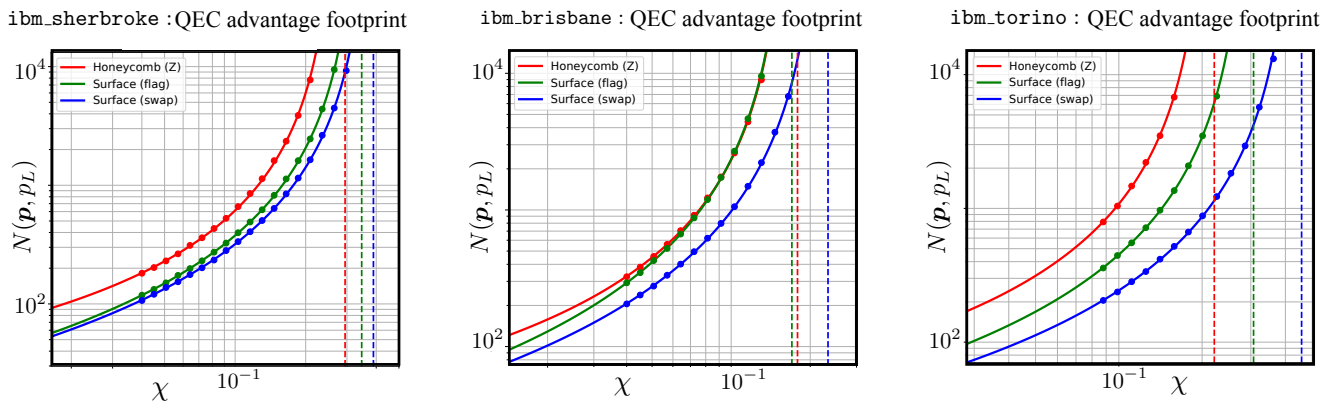
FIG. 17. **QEC-advantage footprint for IBM devices:** Number of physical qubits required to preserve a logical qubit for d QEC rounds with a logical error rate lower than 10^{-4}, considering three different IBM-devices with their corresponding PBCL noise models (8)-(10). We plot the QEC footprint as a function of the ratio χ between improved error and the current calibrated error rate. The surface code with SWAPs has the best behaviour, which nevertheless is not enough to show QEC advantage without a reduction of existing noise.
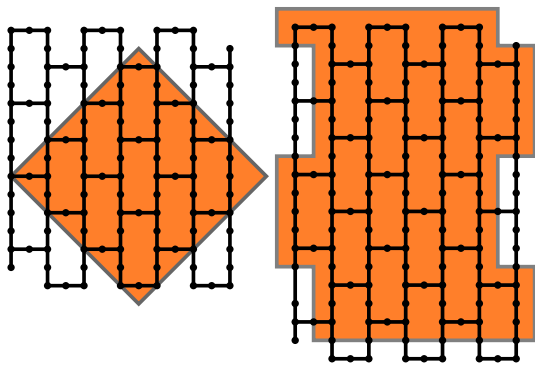

FIG. 18. **QEC-code patches in the IBM layouts:** (left panel) The surface-code patch wastes qubits due to the 45º rotation of boundaries required to maintain fault tolerance with single ancilla qubits. (right panel) The Floquet honeycomb code patch fits better in the IBM architecture, and makes a nearly-optimal use of the available resources.

and `ibm_brisbane` calibration data in table II. In any case, these spare qubits could be used for other purposes, such as additional neighboring ancilla and logical qubits for lattice-surgery logical operations.

VI. CONCLUSIONS AND OUTLOOK

In this work, we have performed a thorough comparison of various QEC strategies for devices with reduced connectivities, focusing in particular on topological QEC codes for the heavy-hexagonal lattice. We have presented optimised SWAP-based and flag-based embeddings of the surface code in this lattice, and compared to other strategies including subsystem-type and Floquet codes. We have found that, overall, the SWAP-based techniques offer the best QEC performance, which has been characterised by computing the error threshold and QEC footprints for various models of noise of increasing sophistication. Considering calibration data of current IBM devices, we have predicted the improvements of the microscopic error rates that would be required work below the QEC threshold which, depending on the particular device would involve a two- to four-fold improvement on the current error rates. Going beyond these numbers and considering a ten-fold improvement, we have found that scaling the heavy-hexagonal `ibm_torino` device to 250 qubits could allow for a demonstration of QEC advantage, i.e. a ten-fold reduction of the logical error rate with respect to the current measurement-dominated error. This target seems at reach of IBM developments in the intermediate or even in the near term.

We believe that further studies that incorporate improvements by QES, and ideally also QEM techniques, will be important to demonstrate this QEC advantage, and move forward towards more complex quantum algorithms with logical encoded qubits. In this respect, it should also be mentioned that important reductions in the footprint with high-threshold QEC codes that deal with several logical qubits can be achieved in architectures that combine reduced local connectivities with specific longer-range couplings [48]. The possibility of embedding these schemes into lower-connectivity architectures with ideas related to those presented in this work, exploring the resulting pattern of the long-range couplings, would also be interesting. Finally, we note that further theoretical and experimental developments for the characterization of the effective circuit-level error models beyond the Pauli-twirled approximation will also be important to assess the progress of QEC. In particular, characterizing the QEC gadgets and logical blocks as a whole, and modelling more efficiently the presence of coherent errors and time-correlations in the noise are interesting lines for future study.

ACKNOWLEDGMENTS

Appendix A: Floquet honeycomb code logical operators

In this Appendix, we describe some technical details about the time-periodic changes of the logical operators in the Floquet honeycomb code [87, 88]. To obtain the value of a plaquette stabilizer of this code (see Fig. 9), one multiplies the measurement outcomes of all parity checks surrounding the cell, collecting the values of all hexagonal cells to obtain the error syndrome. It should be noted that parity checks from different rounds do not commute with each other, so measuring them in sequence causes some stabilizers to change in time.

After a measurement round of a given color, all weight-2 parity checks of the same color become stabilizers themselves, which adds to the plaquette operators that are always stabilizers. Since there is an edge stabilizer associated to every qubit pair, each pair can be seen as a single "effective" qubit (a two qubit system where a parity check is measured requires only one additional stabilizer to be described). These "effective" qubits are associated to the edges of an hexagonal super-lattice. In this super-lattice, plaquette stabilizers from the original lattice form vertex and plaquette operators, which define a toric code over the hexagonal super-lattice (see Fig. 19). One can see that this super-lattice is shifted at each round of the period-3 scheme of the Floquet honeycomb code, causing the system to transition between different "effective" toric codes. This effect also induces a temporal evolution of the logical operators, as they are derived from the "effective" toric code at each round, as depicted in Fig. 20.

Due to the time-dynamics of the logical operators, the full code distance cannot be reached [90], with reduced effective distances for the X_L and Z_L operators in the heavy-hexagons lattice. Due to the different structure of vertical and horizontal operators, this results in an uneven reduction to $d_{\text{eff}}^X \sim \frac{2}{3}d$ and $d_{\text{eff}}^Z = d - 1$, inducing different qubit footprints. Conversely, for the ancilla-free circuit in the hexagonal lattice, this effect has limited importance, since the mechanism described in Sec. II C becomes dominant. The non-FT nature of parity check measurements halves the effective code distance for both operators, i.e. $d_{\text{eff}}^X = d_{\text{eff}}^Z = d/2$, so this effect is more important.

Appendix B: Minimum code distance for a target error rate

In this Appendix, we provide further details on the calculation of the QEC footprints. When benchmarking a QEC code, the standard metric is the code error threshold, which determines the upper bound of the physical error rate for which logical errors can be arbitrarily reduced by enlarging the code. This is an absolute figure since physical error rates must be kept below the threshold for QEC to be beneficial. For a noise model with a single error rate, whenever it is kept below threshold $p < p_{\text{th}}$, the logical error p_L is reduced when code distance d is increased. In the case of a multi-parameter noise model where one re-scales all error rates in units of a single one p by introducing the α parameters, such as those in Eq. (7), one can proceed with a similar analysis.

The effectiveness of increasing the code distance can be parametrized using the so-called Λ model [117], such that

$$p_L = C(p)/(\Lambda(p))^{(d+1)/2}, \tag{B1}$$

where $C(p)$ an $\Lambda(p)$ are functions of the physical error rate, and also the α parameters although we do not write this explicitly. Accordingly, the logical error rate becomes a function of the code distance and the physical error rate, and taking the logarithm, one finds a linear dependence with the code distance that can be exploited for fitting purposes

$$\log p_L(d,p) = \log C(p) - \frac{d+1}{2}\log\Lambda(p). \tag{B2}$$

This method can be used to extrapolate the logical error rate for very large distances without the corresponding numerical overhead of simulating prohibitively-large circuits [89]. As discussed in the main text, this fitting also allows to estimate the distance required to meet a target logical error rate p_L for a given physical error rate p, which is then used to predict the QEC footprint $N(p,p_L)$.

We now discuss a novel empirical relation (12) that can be used for an efficient extrapolation of the QEC footprint to the low-error-rate regime, where Monte Carlo simulations require a huge number of shots. Away from error threshold, when looking at logical versus physical error rate plots (e.g. Fig. 13), one observes a linear dependence between logarithms of physical and logical errors

$$\log p_L(d,p) = a(d)\log p + b(d), \tag{B3}$$

where we have introduced two functions of the code distance $a(d), b(d)$. Also, since $\log p_L$ depends linearly on d as shown in (B2), a and b must satisfy

$$a(d) = a_0 + a_1 d, \qquad b(d) = b_0 + b_1 d. \tag{B4}$$

We fit p_L vs p curves to obtain a and b values for different code distances, and then we fit those $a(d)$ and $b(d)$ to obtain a_0, a_1, b_0 and b_1, which are distance independent. Thus, we have a simple expression that links the logical error rate, the physical error rate and the code

$$\log p_L(d,p) = (a_0 + a_1 d)\log p + (b_0 + b_1 d) \tag{B5}$$

Solving for d, we can obtain the required code distance to achieve a target logical error probability under a given physical error rate $d(p,p_L)$, namely

$$d(p,p_L) = \frac{\log p_L - a_0 \log p - b_0}{a_1 \log p + b_1}, \tag{B6}$$
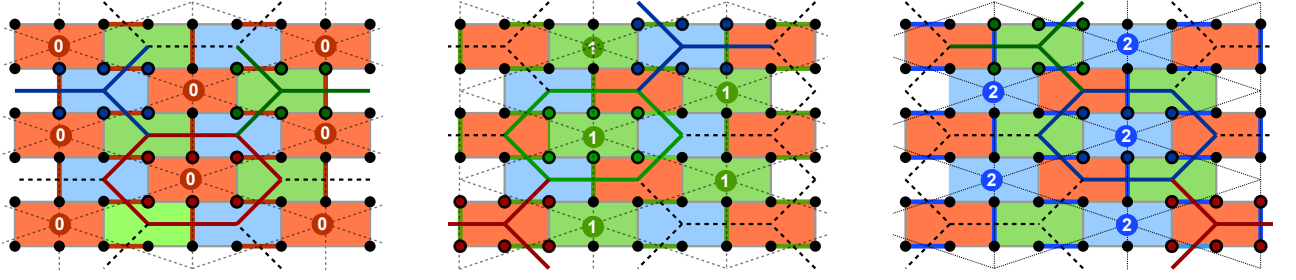
which has been used in Eq. (12) of the main text.

FIG. 19. **Floquet honeycomb code period-3 transitions:** After measuring all weight-2 parity checks of a given color (shown with thick colored lines: 0 (red), 1 (green), and 2 (blue), each qubit pair involved in the corresponding check is equivalent to a single effective qubit. These effective qubits form a toric code state over a hexagonal super-lattice (with its corresponding dual triangular super-lattice with sites at the corresponding plaquette centers, shown in dashed grey). Plaquettes from the Floquet honeycomb code correspond to plaquettes and vertices of the super-lattice toric code. Plaquettes with the same color as the measured edges correspond to plaquettes of the super-lattice, while plaquettes from the other two colors correspond to vertices (or alternatively, to plaquettes of the dual super-lattice).
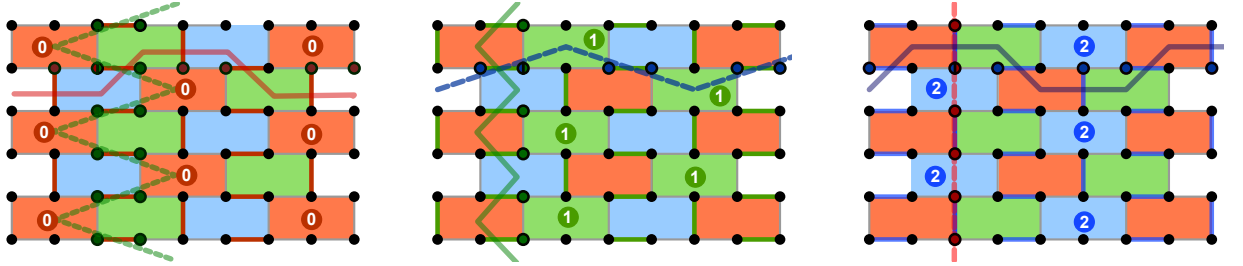

FIG. 20. **Periodic time evolution of the logical operators of the Floquet honeycomb code:** Logical operators for the Floquet honeycomb code in the bulk, inherited from the toric code state at each specific round of the period-3 periodic steps. Operators from two consecutive rounds differ by parity-check operators from the latter round, alternating between primal and dual chains over the super-lattices.

Appendix C: QEC codes on different qubit layouts

The starting point for both the surface code and the Floquet code adaptations to the heavy hexagonal lattice, where circuits designed for a hexagonal lattice [83, 88]. In this Appendix, we estimate the cost of these adaptations by comparing their QEC footprint $N(p, p_L)$ with that of the hexagonal lattice variants, which is calculated using the same approach as discussed in the main text. We also include a comparison of the QEC footprint for different surface code adaptations to the heavy-hexagonal lattice.

In Fig. 21, we show that the differences in the QEC footprint is quite dependent on the qubit layout. We note that, for the same code size, the ancilla-free variant of the Floquet honeycomb code (running on a hexagonal grid) has a lower effective code distance than the variant with ancilla qubits (on a heavy-hexagonal grid), due to the non-FT nature of the syndrome readout of the former (see our general discussion in Sec. II C). Therefore, one expects that an ancilla-based readout should lead to an increase in the effective code distance, improving in this way the QEC performance. This argument, however, does not take into consideration that the additional ancilla qubits needed for the heavy-hexagonal construction will also contribute to the QEC footprint. As we advanced in Sec. II C, the interplay between these two effects must be studied on a case-by-case basis. In fact, we find that the ancilla-free variant is still performing better and leading to smaller

QEC footprints.

With respect to the surface code, we observe similar footprints for the square and hexagonal lattices, with a slightly better threshold for the hexagonal version. For the honeycomb code, the increased qubit overhead associated to the heavy-hexagons lattice was partially compensated by the change to a FT circuit, but there is no such effect for the surface code, since all variants are fully FT. Therefore, its performance drop is bigger when switching to the heavy-hexagons lattice, with threshold being reduced from a competitive $0.7\% - 0.8\%$ (for the hexagonal lattice) to 0.3%.

In Fig. 22 we represent different surface code adaptations to the heavy-hex lattice, motivating the use of our constructions. Namely, we plot the surface code adaptation from [82], which requires 6 flag qubits per stabilizer, the toric code (without boundaries) from [83] requiring 2 flag qubits, our packed version (with boundaries) of the latter, also using 2 flag qubits, and our SWAP variant. The 6-flag circuit requires many operations, thus having a big QEC footprint. The other three alternatives share the same philosophy, originating from the surface code in the hexagonal grid, and show a lower overhead. For the SCL noise model, our two variants provide the best overall results.
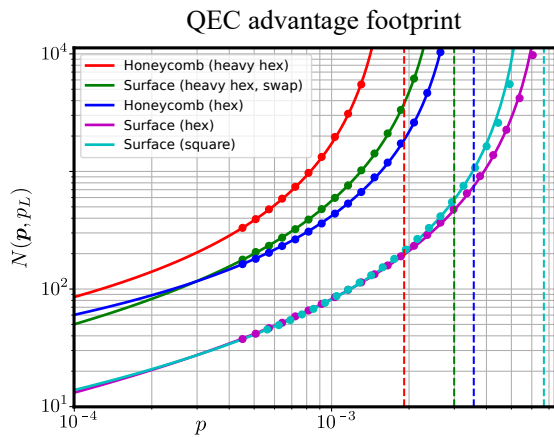
FIG. 21. **Thresholds and QEC footprints for different codes and qubit layouts**: Number of physical qubits $N(p, p_L)$ required to implement a logical qubit with error rate $p_L = 10^{-4}$ for the honeycomb and surface codes on the hexagonal and heavy-hexagonal architectures, using the SCL noise model (6). The dashed vertical lines stand for the corresponding error thresholds. We note that if one goes too close to error threshold, the fitting approximation in Eq. (B3) may fail, as occurs for the standard surface code.
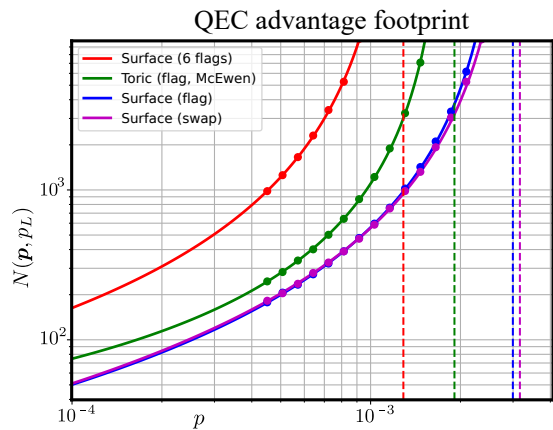

FIG. 22. **QEC-advantage footprint for the surface code in the heavy-hexagonal lattice**: Comparison between our SWAP and flag variants, and codes in [83] and [82], using the SCL (6) noise model.

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2000).

[2] A. Montanaro, Quantum algorithms: an overview, npj Quantum Information **2**, 15023 (2016).

[3] A. M. Dalzell, S. McArdle, M. Berta, P. Bienias, C.-F. Chen, A. Gilyén, C. T. Hann, M. J. Kastoryano, E. T. Khabiboulline, A. Kubica, G. Salton, S. Wang, and F. G. S. L. Brandão, Quantum algorithms: A survey of applications and end-to-end complexities (2023), arXiv:2310.03011 [quant-ph].

[4] W. G. Unruh, Maintaining coherence in quantum computers, Phys. Rev. A **51**, 992 (1995).

[5] F. Arute *et al.*, Quantum supremacy using a programmable superconducting processor, Nature **574**, 505 (2019).

[6] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu, P. Hu, X.-Y. Yang, W.-J. Zhang, H. Li, Y. Li, X. Jiang, L. Gan, G. Yang, L. You, Z. Wang, L. Li, N.-L. Liu, C.-Y. Lu, and J.-W. Pan, Quantum computational advantage using photons, Science **370**, 1460 (2020).

[7] K. Temme, S. Bravyi, and J. M. Gambetta, Error mitigation for short-depth quantum circuits, Phys. Rev. Lett. **119**, 180509 (2017).

[8] P. D. Nation, H. Kang, N. Sundaresan, and J. M. Gambetta, Scalable mitigation of measurement errors on quantum computers, PRX Quantum **2**, 040326 (2021).

[9] E. van den Berg, Z. K. Minev, A. Kandala, and K. Temme, Probabilistic error cancellation with sparse pauli–lindblad models on noisy quantum processors, Nature Physics **19**, 1116 (2023).

[10] Z. Cai, R. Babbush, S. C. Benjamin, S. Endo, W. J. Huggins, Y. Li, J. R. McClean, and T. E. O'Brien, Quantum error mitigation, Rev. Mod. Phys. **95**, 045005 (2023).

[11] L. Viola, E. Knill, and S. Lloyd, Dynamical decoupling of open quantum systems, Phys. Rev. Lett. **82**, 2417 (1999).

[12] K. Khodjasteh and D. A. Lidar, Fault-tolerant quantum dynamical decoupling, Phys. Rev. Lett. **95**, 180501 (2005).

[13] K. Khodjasteh and L. Viola, Dynamically error-corrected gates for universal quantum computation, Phys. Rev. Lett. **102**, 080501 (2009).

[14] D. A. Lidar, Review of decoherence-free subspaces, noiseless subsystems, and dynamical decoupling, in *Quantum Information and Computation for Chemistry* (John Wiley & Sons, Ltd, 2014) pp. 295–354.

[15] J. Preskill, Quantum Computing in the NISQ era and beyond, Quantum **2**, 79 (2018).

[16] Y. Kim, A. Eddins, S. Anand, K. X. Wei, E. van den Berg, S. Rosenblatt, H. Nayfeh, Y. Wu, M. Zaletel, K. Temme, and A. Kandala, Evidence for the utility of quantum computing before fault tolerance, Nature **618**, 500 (2023).

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

[18] A. M. Steane, Error correcting codes in quantum theory, Phys. Rev. Lett. **77**, 793 (1996).

[19] B. M. Terhal, Quantum error correction for quantum memories, Rev. Mod. Phys. **87**, 307 (2015).

[20] D. Aharonov and M. Ben-Or, Fault-tolerant quantum computation with constant error, SIAM J. Comput. **38**, 1207 (1998).

[21] E. Knill, R. Laflamme, and W. H. Zurek, Resilient quantum computation: error models and thresholds, Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences **454**, 365 (1998).

[22] C. D. Bruzewicz, J. Chiaverini, R. McConnell, and J. M. Sage, Trapped-ion quantum computing: Progress and challenges, Applied Physics Reviews **6**, 021314 (2019).

[23] M. Morgado and S. Whitlock, Quantum simulation and computing with Rydberg-interacting qubits, AVS Quantum Sci-

ence **3**, 023501 (2021).

[24] G. Wendin, Quantum information processing with superconducting circuits: a review, Reports on Progress in Physics **80**, 106001 (2017).

[25] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White, J. Mutus, A. G. Fowler, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, C. Neill, P. O'Malley, P. Roushan, A. Vainsencher, J. Wenner, A. N. Korotkov, A. N. Cleland, and J. M. Martinis, Superconducting quantum circuits at the surface code threshold for fault tolerance, Nature **508**, 500 (2014).

[26] C. J. Ballance, T. P. Harty, N. M. Linke, M. A. Sepiol, and D. M. Lucas, High-fidelity quantum logic gates using trapped-ion hyperfine qubits, Phys. Rev. Lett. **117**, 060504 (2016).

[27] S. J. Evered, D. Bluvstein, M. Kalinowski, S. Ebadi, T. Manovitz, H. Zhou, S. H. Li, A. A. Geim, T. T. Wang, N. Maskara, H. Levine, G. Semeghini, M. Greiner, V. Vuletić, and M. D. Lukin, High-fidelity parallel entangling gates on a neutral-atom quantum computer, Nature **622**, 268 (2023).

[28] C. R. Clark, H. N. Tinkey, B. C. Sawyer, A. M. Meier, K. A. Burkhardt, C. M. Seck, C. M. Shappert, N. D. Guise, C. E. Volin, S. D. Fallek, H. T. Hayden, W. G. Rellergert, and K. R. Brown, High-fidelity bell-state preparation with $^{40}ca^+$ optical qubits, Phys. Rev. Lett. **127**, 130505 (2021).

[29] L. Ding, M. Hays, Y. Sung, B. Kannan, J. An, A. Di Paolo, A. H. Karamlou, T. M. Hazard, K. Azar, D. K. Kim, B. M. Niedzielski, A. Melville, M. E. Schwartz, J. L. Yoder, T. P. Orlando, S. Gustavsson, J. A. Grover, K. Serniak, and W. D. Oliver, High-fidelity, frequency-flexible two-qubit fluxonium gates with a transmon coupler, Phys. Rev. X **13**, 031035 (2023).

[30] H. Zhang, C. Ding, D. K. Weiss, Z. Huang, Y. Ma, C. Guinn, S. Sussman, S. P. Chitta, D. Chen, A. A. Houck, J. Koch, and D. I. Schuster, Tunable inductive coupler for high fidelity gates between fluxonium qubits (2023), arXiv:2309.05720 [quant-ph].

[31] D. Kielpinski, C. Monroe, and D. J. Wineland, Architecture for a large-scale ion-trap quantum computer, Nature **417**, 709 (2002).

[32] V. Kaushal, B. Lekitsch, A. Stahl, J. Hilder, D. Pijn, C. Schmiegelow, A. Bermudez, M. Müller, F. Schmidt-Kaler, and U. Poschinger, Shuttling-based trapped-ion quantum information processing, AVS Quantum Science **2**, 014101 (2020).

[33] J. P. Home, D. Hanneke, J. D. Jost, J. M. Amini, D. Leibfried, and D. J. Wineland, Complete methods set for scalable ion trap quantum information processing, Science **325**, 1227 (2009).

[34] J. M. Pino *et al.*, Demonstration of the trapped-ion quantum CCD computer architecture, Nature **592**, 209 (2021).

[35] C. Ryan-Anderson, J. G. Bohnet, K. Lee, D. Gresh, A. Hankin, J. P. Gaebler, D. Francois, A. Chernoguzov, D. Lucchetti, N. C. Brown, T. M. Gatterman, S. K. Halit, K. Gilmore, J. A. Gerber, B. Neyenhuis, D. Hayes, and R. P. Stutz, Realization of real-time fault-tolerant quantum error correction, Phys. Rev. X **11**, 041058 (2021).

[36] J. Hilder, D. Pijn, O. Onishchenko, A. Stahl, M. Orth, B. Lekitsch, A. Rodriguez-Blanco, M. Müller, F. Schmidt-Kaler, and U. G. Poschinger, Fault-tolerant parity readout on a shuttling-based trapped-ion quantum computer, Phys. Rev. X **12**, 011032 (2022).

[37] J. Beugnon, C. Tuchendler, H. Marion, A. Gaëtan, Y. Miroshnychenko, Y. R. P. Sortais, A. M. Lance, M. P. A. Jones, G. Messin, A. Browaeys, and P. Grangier, Two-dimensional transport and transfer of a single atomic qubit in optical tweez-

ers, Nature Physics **3**, 696 (2007).

[38] D. Barredo, S. de Léséleuc, V. Lienhard, T. Lahaye, and A. Browaeys, An atom-by-atom assembler of defect-free arbitrary two-dimensional atomic arrays, Science **354**, 1021 (2016).

[39] M. Endres, H. Bernien, A. Keesling, H. Levine, E. R. Anschuetz, A. Krajenbrink, C. Senko, V. Vuletic, M. Greiner, and M. D. Lukin, Atom-by-atom assembly of defect-free one-dimensional cold atom arrays, Science **354**, 1024 (2016).

[40] 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 (2022).

[41] D. Bluvstein, S. J. Evered, A. A. Geim, S. H. Li, H. Zhou, T. Manovitz, S. Ebadi, M. Cain, M. Kalinowski, D. Hangleiter, J. P. B. Ataides, N. Maskara, I. Cong, X. Gao, P. S. Rodriguez, T. Karolyshyn, G. Semeghini, M. J. Gullans, M. Greiner, V. Vuletić, and M. D. Lukin, Logical quantum processor based on reconfigurable atom arrays, Nature 10.1038/s41586-023-06927-3 (2023).

[42] S. Debnath, N. M. Linke, C. Figgatt, K. A. Landsman, K. Wright, and C. Monroe, Demonstration of a small programmable quantum computer with atomic qubits, Nature **536**, 63 (2016).

[43] C. Figgatt, A. Ostrander, N. M. Linke, K. A. Landsman, D. Zhu, D. Maslov, and C. Monroe, Parallel entangling operations on a universal ion-trap quantum computer, Nature **572**, 368 (2019).

[44] I. Pogorelov, T. Feldker, C. D. Marciniak, L. Postler, G. Jacob, O. Krieglsteiner, V. Podlesnic, M. Meth, V. Negnevitsky, M. Stadler, B. Höfer, C. Wächter, K. Lakhmanskiy, R. Blatt, P. Schindler, and T. Monz, Compact ion-trap quantum computing demonstrator, PRX Quantum **2**, 020343 (2021).

[45] L. Postler, F. Butt, I. Pogorelov, C. D. Marciniak, S. Heußen, R. Blatt, P. Schindler, M. Rispler, M. Müller, and T. Monz, Demonstration of fault-tolerant steane quantum error correction (2023), arXiv:2312.09745 [quant-ph].

[46] S. Huang, K. R. Brown, and M. Cetina, Comparing shor and steane error correction using the bacon-shor code (2023), arXiv:2312.10851 [quant-ph].

[47] D. Rosenberg, D. Kim, R. Das, D. Yost, S. Gustavsson, D. Hover, P. Krantz, A. Melville, L. Racz, G. O. Samach, S. J. Weber, F. Yan, J. L. Yoder, A. J. Kerman, and W. D. Oliver, 3d integrated superconducting qubits, npj Quantum Information **3**, 42 (2017).

[48] 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 (2023), arXiv:2308.07915 [quant-ph].

[49] J. Kelly *et al.*, State preservation by repetitive error detection in a superconducting quantum circuit, Nature **519**, 66 (2015).

[50] C. Neill, P. Roushan, K. Kechedzhi, S. Boixo, S. V. Isakov, V. Smelyanskiy, A. Megrant, B. Chiaro, A. Dunsworth, K. Arya, R. Barends, B. Burkett, Y. Chen, Z. Chen, A. Fowler, B. Foxen, M. Giustina, R. Graff, E. Jeffrey, T. Huang, J. Kelly, P. Klimov, E. Lucero, J. Mutus, M. Neeley, C. Quintana, D. Sank, A. Vainsencher, J. Wenner, T. C. White, H. Neven, and J. M. Martinis, A blueprint for demonstrating quantum supremacy with superconducting qubits, Science **360**, 195 (2018).

[51] E. Lucero, R. Barends, Y. Chen, J. Kelly, M. Mariantoni, A. Megrant, P. O'Malley, D. Sank, A. Vainsencher, J. Wenner, T. White, Y. Yin, A. N. Cleland, and J. M. Martinis, Com-

puting prime factors with a josephson phase qubit quantum processor, Nature Physics **8**, 719 (2012).

[52] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, Nature **549**, 242 (2017).

[53] K. X. Wei, I. Lauer, S. Srinivasan, N. Sundaresan, D. T. Mc-Clure, D. Toyli, D. C. McKay, J. M. Gambetta, and S. Sheldon, Verifying multipartite entangled greenberger-horne-zeilinger states via multiple quantum coherences, Phys. Rev. A **101**, 032343 (2020).

[54] 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 (2022).

[55] 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, M. 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 G. Q. AI, Suppressing quantum errors by scaling a surface code logical qubit, Nature **614**, 676 (2023).

[56] A. Kitaev, Fault-tolerant quantum computation by anyons, Annals of Physics **303**, 2 (2003).

[57] S. B. Bravyi and A. Y. Kitaev, Quantum codes on a lattice with boundary (1998), arXiv:quant-ph/9811052 [quant-ph].

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

[59] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Surface codes: Towards practical large-scale quantum computation, Phys. Rev. A **86**, 032324 (2012).

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

[61] J. M. Gambetta, J. M. Chow, and M. Steffen, Building logical

qubits in a superconducting quantum computing system, npj Quantum Information **3**, 2 (2017).

[62] D. Gottesman, Stabilizer codes and quantum error correc- tion, Ph.D. thesis, arXiv:9705052(1997) (1997).

[63] A. M. Stephens, Fault-tolerant thresholds for quantum error correction with the surface code, Phys. Rev. A **89**, 022321 (2014).

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

[65] D. S. Wang, A. G. Fowler, A. M. Stephens, and L. C. L. Hollenberg, Threshold error rates for the toric and planar codes, Quantum Info. Comput. **10**, 456–469 (2010).

[66] P. Aliferis, D. Gottesman, and J. Preskill, Quantum accuracy threshold for concatenated distance-3 codes, Quantum Inf. Comput. **6**, 97 (2005).

[67] K. M. Svore, D. P. Divincenzo, and B. M. Terhal, Noise threshold for a fault-tolerant two-dimensional lattice architecture, Quantum Info. Comput. **7**, 297–318 (2007).

[68] L. DiCarlo, J. M. Chow, J. M. Gambetta, L. S. Bishop, B. R. Johnson, D. I. Schuster, J. Majer, A. Blais, L. Frunzio, S. M. Girvin, and R. J. Schoelkopf, Demonstration of two-qubit algorithms with a superconducting quantum processor, Nature **460**, 240 (2009).

[69] 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, Phys. Rev. Lett. **129**, 030501 (2022).

[70] Y. Ye, T. He, H.-L. Huang, Z. Wei, Y. Zhang, Y. Zhao, D. Wu, Q. Zhu, H. Guan, 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, 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, Logical magic state preparation with fidelity beyond the distillation threshold on a superconducting quantum processor (2023), arXiv:2305.15972 [quant-ph].

[71] R. S. Gupta, N. Sundaresan, T. Alexander, C. J. Wood, S. T. Merkel, M. B. Healy, M. Hillenbrand, T. Jochym-O'Connor, J. R. Wootton, T. J. Yoder, A. W. Cross, M. Takita, and B. J. Brown, Encoding a magic state with beyond break-even fidelity, Nature **625**, 259 (2024).

[72] J. M. Chow, A. D. Córcoles, J. M. Gambetta, C. Rigetti, B. R. Johnson, J. A. Smolin, J. R. Rozen, G. A. Keefe, M. B. Rothwell, M. B. Ketchen, and M. Steffen, Simple all-microwave entangling gate for fixed-frequency superconducting qubits, Phys. Rev. Lett. **107**, 080502 (2011).

[73] S. Sheldon, E. Magesan, J. M. Chow, and J. M. Gambetta, Procedure for systematically tuning up cross-talk in the cross-resonance gate, Phys. Rev. A **93**, 060302 (2016).

[74] M. D. Reed, L. DiCarlo, S. E. Nigg, L. Sun, L. Frunzio, S. M. Girvin, and R. J. Schoelkopf, Realization of three-qubit quantum error correction with superconducting circuits, Nature **482**, 382 (2012).

[75] J. M. Chow *et al.*, Implementing a strand of a scalable fault-tolerant quantum computing fabric, Nature Communications **5**, 4015 EP (2014), article.

[76] A. D. Córcoles, E. Magesan, S. J. Srinivasan, A. W. Cross, M. Steffen, J. M. Gambetta, and J. M. Chow, Demonstration of a quantum error detection code using a square lattice of

four superconducting qubits, Nature Communications **6**, 6979 (2015).

[77] 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, Phys. Rev. Lett. **117**, 210505 (2016).

[78] M. Takita, A. W. Cross, A. D. Córcoles, J. M. Chow, and J. M. Gambetta, Experimental demonstration of fault-tolerant state preparation with superconducting qubits, Phys. Rev. Lett. **119**, 180501 (2017).

[79] C. Chamberland, G. Zhu, T. J. Yoder, J. B. Hertzberg, and A. W. Cross, Topological and subsystem codes on low-degree graphs with flag qubits, Phys. Rev. X **10**, 011022 (2020).

[80] R. Chao and B. W. Reichardt, Quantum error correction with only two extra qubits, Phys. Rev. Lett. **121**, 050502 (2018).

[81] R. Chao and B. W. Reichardt, Flag fault-tolerant error correction for any stabilizer code, PRX Quantum **1**, 010302 (2020).

[82] Y. Kim, J. Kang, and Y. Kwon, Design of quantum error correcting code for biased error on heavy-hexagon structure, Quantum Information Processing **22**, 230 (2023).

[83] M. McEwen, D. Bacon, and C. Gidney, Relaxing Hardware Requirements for Surface Code Circuits using Time-dynamics, Quantum **7**, 1172 (2023).

[84] D. Bacon and A. Casaccino, Quantum error correcting subsystem codes from two classical linear codes (2006), arXiv:quant-ph/0610088 [quant-ph].

[85] P. Aliferis and A. W. Cross, Subsystem fault tolerance with the bacon-shor code, Phys. Rev. Lett. **98**, 220502 (2007).

[86] J. Napp and J. Preskill, Optimal bacon-shor codes, Quantum Inf. Comput. **13**, 490–510 (2013).

[87] J. Haah and M. B. Hastings, Boundaries for the Honeycomb Code, Quantum **6**, 693 (2022).

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

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

[90] C. Gidney, M. Newman, and M. McEwen, Benchmarking the Planar Honeycomb Code, Quantum **6**, 813 (2022).

[91] D. C. McKay, I. Hincks, E. J. Pritchett, M. Carroll, L. C. G. Govia, and S. T. Merkel, Benchmarking quantum processor performance at scale (2023), arXiv:2311.05933 [quant-ph].

[92] A. Y. Kitaev, Quantum error correction with imperfect gates, in *Quantum Communication, Computing, and Measurement*, edited by O. Hirota, A. S. Holevo, and C. M. Caves (Springer US, Boston, MA, 1997) pp. 181–188.

[93] D. P. DiVincenzo and P. W. Shor, Fault-tolerant error correction with efficient quantum codes, Phys. Rev. Lett. **77**, 3260 (1996).

[94] A. M. Steane, Active stabilization, quantum computation, and quantum state synthesis, Phys. Rev. Lett. **78**, 2252 (1997).

[95] E. Knill, Quantum computing with realistically noisy devices, Nature **434**, 39 (2005).

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

[97] H. Bombin, Topological subsystem codes, Phys. Rev. A **81**, 032301 (2010).

[98] M. Li, D. Miller, M. Newman, Y. Wu, and K. R. Brown, 2d compass codes, Phys. Rev. X **9**, 021041 (2019).

[99] C. Gidney and D. Bacon, Less bacon more threshold (2023), arXiv:2305.12046 [quant-ph].

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

[101] L. Grans-Samuelsson, R. V. Mishmash, D. Aasen, C. Knapp, B. Bauer, B. Lackey, M. P. da Silva, and P. Bonderson, Improved pairwise measurement-based surface code (2023), arXiv:2310.12981 [quant-ph].

[102] D. Ristè, J. G. van Leeuwen, H.-S. Ku, K. W. Lehnert, and L. DiCarlo, Initialization by measurement of a superconducting quantum bit circuit, Phys. Rev. Lett. **109**, 050507 (2012).

[103] D. Ristè, M. Dukalski, C. A. Watson, G. de Lange, M. J. Tiggelman, Y. M. Blanter, K. W. Lehnert, R. N. Schouten, and L. DiCarlo, Deterministic entanglement of superconducting qubits by parity measurement and feedback, Nature **502**, 350 (2013).

[104] W. P. Livingston, M. S. Blok, E. Flurin, J. Dressel, A. N. Jordan, and I. Siddiqi, Experimental demonstration of continuous quantum error correction, Nature Communications **13**, 2307 (2022).

[105] D. Gottesman, An introduction to quantum error correction and fault-tolerant quantum computation (2009), arXiv:0904.2557 [quant-ph].

[106] A. Paetznick, C. Knapp, N. Delfosse, B. Bauer, J. Haah, M. B. Hastings, and M. P. da Silva, Performance of planar floquet codes with majorana-based qubits, PRX Quantum **4**, 010310 (2023).

[107] D. Gottesman, The heisenberg representation of quantum computers (1998), arXiv:quant-ph/9807006 [quant-ph].

[108] S. Aaronson and D. Gottesman, Improved simulation of stabilizer circuits, Phys. Rev. A **70**, 052328 (2004).

[109] C. Gidney, Stim: a fast stabilizer circuit simulator, Quantum **5**, 497 (2021).

[110] O. Higgott and C. Gidney, Sparse blossom: correcting a million errors per core second with minimum-weight matching (2023), arXiv:2303.15933 [quant-ph].

[111] A. Townsend-Teague, J. Magdalena de la Fuente, and M. Kesselring, Floquetifying the colour code, Electronic Proceedings in Theoretical Computer Science **384**, 265–303 (2023).

[112] P. Aliferis and J. Preskill, Fault-tolerant quantum computation against biased noise, Phys. Rev. A **78**, 052331 (2008).

[113] D. K. Tuckett, S. D. Bartlett, and S. T. Flammia, Ultrahigh error threshold for surface codes with biased noise, Phys. Rev. Lett. **120**, 050505 (2018).

[114] W. Dür, M. Hein, J. I. Cirac, and H.-J. Briegel, Standard forms of noisy quantum operations via depolarization, Phys. Rev. A **72**, 052326 (2005).

[115] Z. Cai and S. C. Benjamin, Constructing smaller pauli twirling sets for arbitrary error channels, Scientific Reports **9**, 11281 (2019).

[116] M. R. Geller and Z. Zhou, Efficient error models for fault-tolerant architectures and the pauli twirling approximation, Phys. Rev. A **88**, 012314 (2013).

[117] Z. Chen, K. J. Satzinger, J. Atalaya, A. N. Korotkov, A. Dunsworth, D. Sank, C. Quintana, M. McEwen, R. Barends, P. V. Klimov, S. Hong, C. Jones, A. Petukhov, D. Kafri, S. Demura, B. Burkett, C. Gidney, A. G. Fowler, A. Paler, H. Putterman, I. Aleiner, F. Arute, K. Arya, R. Babbush, J. C. Bardin, A. Bengtsson, A. Bourassa, M. Broughton, B. B. Buckley, D. A. Buell, N. Bushnell, B. Chiaro, R. Collins, W. Courtney, A. R. Derk, D. Eppens, C. Erickson, E. Farhi, B. Foxen, M. Giustina, A. Greene, J. A. Gross, M. P. Harrigan, S. D. Harrington, J. Hilton, A. Ho, T. Huang, W. J. Huggins, L. B. Ioffe, S. V. Isakov, E. Jeffrey, Z. Jiang, K. Kechedzhi, S. Kim, A. Kitaev, F. Kostritsa, D. Landhuis, P. Laptev, E. Lucero, O. Martin, J. R. McClean, T. McCourt, X. Mi, K. C. Miao, M. Mohseni, S. Montazeri, W. Mruczkiewicz, J. Mutus, O. Naaman, M. Neeley, C. Neill, M. Newman, M. Y.

Niu, T. E. O'Brien, A. Opremcak, E. Ostby, B. Pató, N. Redd, P. Roushan, N. C. Rubin, V. Shvarts, D. Strain, M. Szalay, M. D. Trevithick, B. Villalonga, T. White, Z. J. Yao, P. Yeh, J. Yoo, A. Zalcman, H. Neven, S. Boixo, V. Smelyanskiy, Y. Chen, A. Megrant, J. Kelly, and G. Q. AI, Exponential suppression of bit or phase errors with cyclic error correction, Nature **595**, 383 (2021).

[118] B. Pokharel and D. A. Lidar, Demonstration of algorithmic quantum speedup, Phys. Rev. Lett. **130**, 210602 (2023).

[119] H. K. Ng, D. A. Lidar, and J. Preskill, Combining dynamical decoupling with fault-tolerant quantum computation, Phys. Rev. A **84**, 012305 (2011).

[120] G. A. Paz-Silva and D. A. Lidar, Optimally combining dynamical decoupling and quantum error correction, Scientific Reports **3**, 1530 (2013).