The XZZX surface code

J. Pablo Bonilla-Ataides,[1] David K. Tuckett,[1] Stephen D. Bartlett,[1] Steven T. Flammia,[2] and Benjamin J. Brown[1]

[1] *Centre for Engineered Quantum Systems, School of Physics, University of Sydney, Sydney, New South Wales 2006, Australia*

[2] *AWS Center for Quantum Computing, Pasadena, CA 91125, USA*

(Dated: August 24, 2020)

We show that a variant of the surface code—the XZZX code—offers remarkable performance for fault-tolerant quantum computation. The error threshold of this code achieves the zero-rate hashing bound for *every* single-qubit Pauli noise channel; it is the first explicit code shown to have this universal property. We present numerical evidence that this threshold even exceeds the hashing bound for an experimentally relevant range of noise parameters. Focusing on the common situation where qubit dephasing is the dominant noise, we show that this code has a practical, high-performance decoder and surpasses all previously known thresholds in the realistic setting where syndrome measurements are unreliable. We go on to demonstrate the favorable sub-threshold resource scaling that can be obtained by specializing a code to exploit structure in the noise. We show that it is possible to maintain all of these advantages when we perform fault-tolerant quantum computation. We finally suggest some small-scale experiments that could exploit noise bias to reduce qubit overhead in two-dimensional architectures.

I. INTRODUCTION

A large-scale quantum computer must be able to reliably process data encoded in a nearly noiseless quantum system. To build such a quantum computer using physical qubits that experience errors from noise and faulty control, we require an architecture that operates fault-tolerantly [1–4], using quantum error correction to repair errors that occur throughout the computation.

For a fault-tolerant architecture to be practical, it will need to correct for physically-relevant errors with only a modest overhead. That is, quantum error correction can be used to create near-perfect logical qubits if the rate of relevant errors on the physical qubits is below some threshold, and a good architecture should have a sufficiently high threshold to be achievable in practice. This fault-tolerant designs should also be efficient, using a reasonable number of physical qubits to achieve the desired logical error rate. The most common architecture for fault-tolerant quantum computing is based on the surface code [5]. It offers very favorable thresholds against depolarizing noise, but recent results have shown that its performance against more structured noise can be considerably improved by tailoring the code to the noise model [6–8]. While the thresholds are favorable, the overheads for the surface code architecture are daunting [5, 9]. For fault-tolerant quantum computing to become practical, there is a need for new architectures that provide high thresholds against a variety of structured noise models and that offer reduced overheads through efficiencies in physical qubits and logic gates.

In this paper, we present a highly efficient fault tolerant architecture design that exploits the common structures in the noise experienced by physical qubits. Our central tool is a variant of the surface code [10–12] where the check operators are given by the product XZZX of Pauli operators around each face on a square lattice [13]. This seemingly innocuous local change of basis offers a number of significant advantages over its more conventional counterpart for noise models that deviate from depolarizing noise.

We first consider preserving a logical qubit in a quantum memory using this XZZX code. While the some 2D codes have been shown to have high error thresholds for certain types of biased noise [6, 14], we find that the XZZX code gives exceptional thresholds for *all* single-qubit Pauli noise channels, matching what is known to be achievable with random coding [15], [16, Theorem 24.6.2]. It is a particularly striking consequence of our results that we can operate a fault-tolerant quantum computer experiencing any single-qubit Pauli error model with the same threshold performance of a random code, while retaining the practical benefits of local stabilizers and an efficient decoder. Intriguingly, we have numerical evidence to suggest that the XZZX threshold in fact *exceeds* this random coding (hashing) bound for noise that is strongly biased towards X (or Z), and so our code can operate in a regime where random coding may not work. Thus, this code could potentially provide a practical demonstration of the superadditivity of the coherent information (a topic of considerable theoretical interest [17–21]).

We show that this high threshold persists with efficient, practical decoders, by using a generalisation of a matching decoder in the regime where dephasing noise is dominant. In the fault-tolerant setting when stabilizer measurements are unreliable, we obtain thresholds in the biased-noise regime that surpass all previously known thresholds.

Along with offering high thresholds against errors, we show that the XZZX code can also offer a significant reduction in the overhead because of a favorable sub-threshold scaling of the logical failure rate. This scaling determines how many physical qubits will be needed for a code to obtain a target logical failure rate. Specifically, we show that this sub-threshold scaling can be affected by structured noise in two ways. When the physical noise

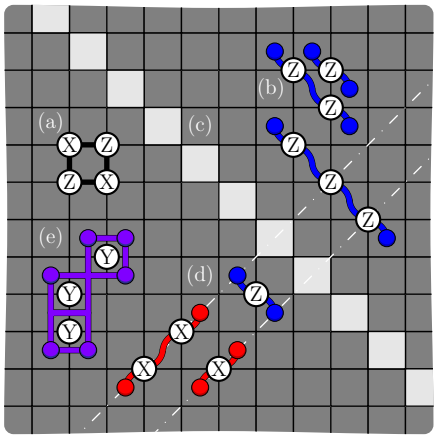

FIG. 1. The XZZX surface code. Qubits lie on the vertices of the square lattice. The codespace is the common $+1$ eigenspace of its stabilizers S_f for all faces of the lattice f. (a) An example of a stabilizer S_f. Unlike the conventional surface code, all stabilizer take the same form for every face. (b) Pauli-Z errors give rise to string-like errors that align along a common direction, enabling a one-dimensional decoding strategy. (c) The product of stabilizer operators along a diagonal give rise to the symmetries under an infinite bias dephasing noise model [8, 30]. (d) Pauli-X errors align along lines with an orthogonal orientation. At finite bias, errors in conjugate bases couple the lines. (e) Pauli-Y errors can be decoded as in Ref. [8].

rate is p and is biased towards dephasing with noise bias η, the logical failure rate scales like $O((p/\sqrt{\eta})^{d/2})$, where d is the distance of the code. Here $\eta \sim 1$ corresponds to depolarising noise and $\eta \to \infty$ to a dephasing noise model. Thus, experimentally plausible values of η (say, between 10 and 1000 [22, 23]) could lead to large improvements in the logical failure rate with a finite number of biased physical qubits. We also show that at small system sizes and near to the threshold, the logical failure rate has a quadratically improved scaling with code distance, as $O(p^{d^2/2})$; prior work [7] demonstrated this d^2 scaling only at infinite bias.

Finally, we consider fault-tolerant quantum computation with biased noise [24–26], and we show that the advantages of the XZZX code persist in this context. We show how to implement low-overhead fault-tolerant Clifford gates by taking advantage of the noise structure as the XZZX code undergoes measurement-based deformations [27–29]. With an appropriate lattice orientation, noise with bias η is shown to yield a reduction in the required number of physical qubits by a factor of $\sim \log \eta$ in a large-scale quantum computation. These advantages already manifest at code sizes attainable using present-day quantum devices.

II. THE XZZX SURFACE CODE

The XZZX surface code is locally equivalent to the conventional surface code [10, 12, 31]. The stabilizer generators S_f are associated with each face and are given by the product of two Pauli-X terms and two Pauli-Z terms as shown in Fig. 1(a). This variant of the surface code was first presented in Ref. [13], and was subsequently considered as a topological memory [32]. To contrast the XZZX surface code with its conventional counterpart, we refer to the latter as the CSS surface code since it is of Calderbank-Shor-Steane type [33, 34].

Together with a choice of code, we require a decoding algorithm to determine which errors have occurred and correct for them. We will consider Pauli errors $E \in \mathcal{P}$, and we say that E creates a defect at face f if $S_f E = (-1)ES_f$. A decoder takes as input the error syndrome (the locations of the defects) and returns a correction that will recover the encoded information with high probability.

Because of the local change of basis, the XZZX surface code responds differently to Pauli errors compared with the CSS surface code. We can take advantage of this difference to design better decoding algorithms. Let us consider the effect of different types of Pauli errors, starting with Pauli-Z errors. A single Pauli-Z error gives rise to two nearby defects. In fact, we can regard a Pauli-Z error as a segment of a string where defects lie at the endpoints of the string segment, and where multiple Pauli-Z errors compound into longer strings, see Fig. 1(b).

A key feature of the XZZX code that we will exploit is that Pauli-Z error strings align along the same direction, as shown in Fig. 1(b). We can understand this phenomenon in more formal terms from the perspective of symmetries [8, 30]. Indeed, the product of face operators along a diagonal such as that shown in Fig. 1(c) commute with Pauli-Z errors. This symmetry guarantees that defects created by Pauli-Z errors will respect a parity conservation law on the faces of a diagonal oriented along this direction. Using this property, we can decode Pauli-Z errors on the XZZX code as a series of disjoint repetition codes. It follows that, for a noise model described by independent Pauli-Z errors, this code has a threshold error rate of 50%.

Likewise, Pauli-X errors act similarly to Pauli-Z errors, but with Pauli-X error strings aligned along the orthogonal direction to the Pauli-Z error strings. In general, we would like to be able to decode all local Pauli errors, where error configurations of Pauli-X and Pauli-Z errors violate the one-dimensional symmetries we have introduced, e.g. Fig. 1(d). As we will see, we can generalize conventional decoding methods to account for finite but high bias of one Pauli operator relative to others and maintain a very high threshold.

We finally remark that the XZZX surface code responds to Pauli-Y errors in the same way as the standard surface code. Each Pauli-Y error will create four defects on each of their adjacent faces; see Fig. 1(e). We there-

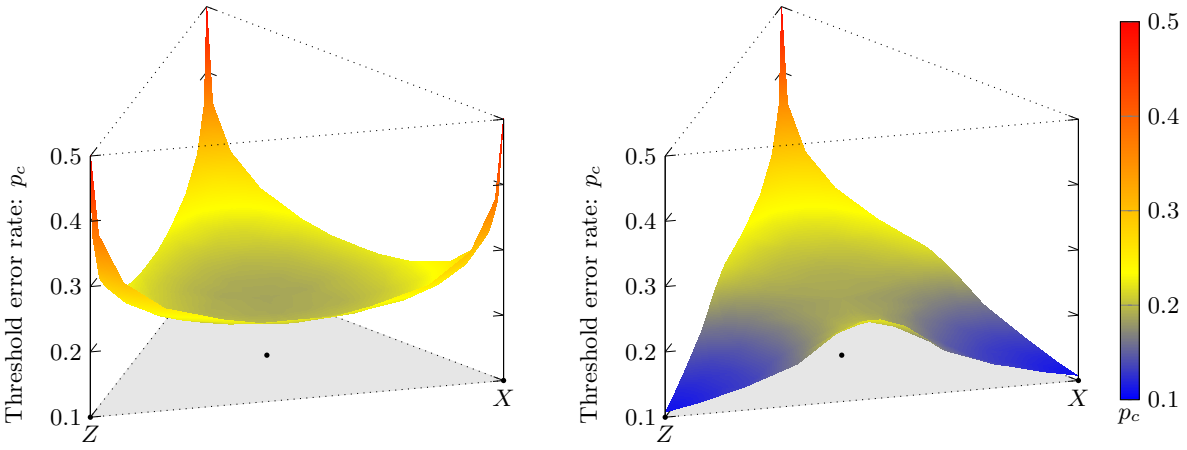

FIG. 2. (Left) Threshold estimates for the XZZX surface code over all single-qubit Pauli channels using approximate maximum-likelihood decoding. The gray triangle represents the surface of all single-qubit Pauli channels, where the center corresponds to standard depolarizing noise, the labeled vertices correspond to pure X and Z noise, and the third vertex corresponds to pure Y noise. The estimates closely match the hashing bound (not shown) for all single-qubit Pauli channels. (Right) Threshold estimates for the CSS surface code over all single-qubit Pauli channels using approximate maximum-likelihood decoding. The estimates closely match the hashing bound (not shown) for Y-biased noise but fall well below for X-biased (and, by code symmetry, Z-biased) noise. All estimates use code distances $d \in \{13, 17, 21, 25\}$.

fore see that the high-performance decoders presented in Refs. [6–8] are readily adapted for the XZZX code in this limit.

III. OPTIMAL THRESHOLDS

The XZZX code has exceptional thresholds for *all* single-qubit Pauli noise channels. We demonstrate this using an efficient maximum-likelihood decoder [35]. This decoder gives the optimal threshold attainable with the code for any given noise model. Remarkably, we find that the XZZX surface code achieves threshold error rates that closely match the zero-rate hashing bound for all single-qubit Pauli noise channels, and appears to exceed this bound in some regimes.

We define the general single-qubit Pauli noise channel

$$\mathcal{E}(\rho) = (1-p)\rho + p(r_X X\rho X + r_Y Y\rho Y + r_Z Z\rho Z) \quad (1)$$

where p is the probability of any error on a single qubit and the channel is parameterized by the stochastic vector $\boldsymbol{r} = (r_X, r_Y, r_Z)$, where $r_X, r_Y, r_Z \geq 0$ and $r_X + r_Y + r_Z = 1$. The surface of all possible values of $\boldsymbol{r}$ parametrize an equilateral triangle, where the centre point $(1/3, 1/3, 1/3)$ corresponds to standard depolarizing noise, and vertices $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$ correspond to pure X, Y and Z noise, respectively. We also define *biased noise* channels, which are restrictions of this general noise channel, parameterized by the scalar η; for example, in the case of Z-biased noise, we define $\eta = r_Z/(r_X + r_Y)$ where $r_X = r_Y$, such that $\eta = 0.5$ corresponds to standard depolarizing noise and the limit $\eta \to \infty$ corresponds to pure Z noise.

We estimate the threshold error rate as a function of $\boldsymbol{r}$ for both the XZZX surface code with boundaries and the CSS surface code with boundaries, see Fig. 13, using a tensor network decoder that gives a controlled approximation to the maximum-likelihood decoder [6, 7, 35]. Our results are summarized in Fig. 2. We find that the thresholds of the XZZX surface code closely match or slightly exceed (as discussed below), the zero-rate hashing bound for all investigated values of $\boldsymbol{r}$, with a global minimum $p_c = 18.7(1)\%$ at standard depolarizing noise and peaks $p_c \sim 50\%$ at pure X, Y and Z noise. We find that the thresholds of the CSS surface code closely match the hashing bound for Y-biased noise, where Y errors dominate, consistent with prior work [6, 7], as well as for channels where $r_Y < r_X = r_Z$ such that X and Z errors dominate but are balanced. In contrast to the XZZX surface code, we find that the thresholds of the CSS surface code fall well below the hashing bound as either X or Z errors dominate with a global minimum $p_c = 10.8(1)\%$ at pure X and pure Z noise.

In some cases, our estimates of XZZX surface code thresholds appear to exceed the zero-rate hashing bound; we now investigate this further. For the values of $\boldsymbol{r}$ investigated for Fig. 2, the mean difference between our estimates and the hashing bound is $\overline{p_c - p_{\text{h.b.}}} = -0.1(3)\%$ and our estimates never fall more than 1.1% below the hashing bound. However, for high bias, $\eta \geq 100$, we observe an asymmetry between Y-biased noise and X-biased (or, by code symmetry, Z-biased) noise. In particular, with observe that, while threshold estimates with Y-biased noise match the hashing bound to within error bars, threshold estimates with highly-biased X-noise, significantly exceed the hashing bound. Our results with X-biased noise are summarized in Fig. 3, where, since

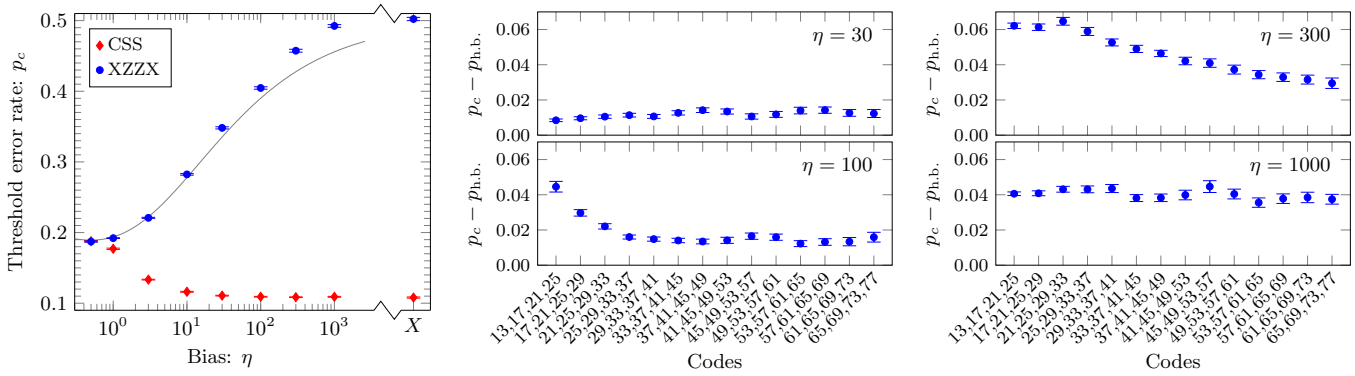

FIG. 3. (Left) Threshold estimates, p_c, for the XZZX and CSS surface codes as a function of bias, η, with X-biased (or, by code symmetry, Z-biased) noise using approximate maximum-likelihood decoding. Error bars indicate one standard deviation relative to the fitting procedure. The gray line is the hashing bound for the associated Pauli noise channel. For high bias, $\eta \geq 30$, the estimates for the XZZX surface code exceed the hashing bound. To investigate this surprising effect, estimates for the XZZX surface code with $30 \leq \eta \leq 1000$ use large code distances, $d \in \{65, 69, 73, 77\}$; other estimates use code distances $d \in \{13, 17, 21, 25\}$ (as used for Fig. 2). (Center and right) Difference between threshold estimates for the XZZX surface code with X-biased noise and the hashing bound, $p_c - p_{\text{h.b.}}$, as a function of code distances used in the estimation. Data is shown for biases $\eta = 30$, 100, 300, 1000. Error bars indicate one standard deviation relative to the fitting procedure. Threshold estimates exceed the hashing bound in all cases. The gap reduces, in most cases, with sets of greater code distance, but it persists and appears to stabilize for $\eta = 30$, 100 and 1000.

thresholds are defined in the limit of infinite code distance, we provide estimates with sets of increasing code distance for $\eta \geq 30$. Although the gap typically reduces, it appears to stabilize for $\eta = 30$, 100, 1000, where we find $p_c - p_{\text{h.b.}} = 1.2(2)\%$, $1.6(3)\%$, $3.7(3)\%$, respectively, with the largest code distances; for $\eta = 300$, the gap exceeds 3% but has clearly not yet stabilized. This evidence for exceeding the hashing bound appears to be robust, but warrants further study.

IV. A FAULT-TOLERANT DECODER

Having demonstrated the remarkable code capacity thresholds of the XZZX code, we now demonstrate how to translate these high thresholds into practice using a symmetry-adapted matching decoder [8]. With this decoder, we calculate record-breaking *fault-tolerant* thresholds, which allow for noisy measurements, with respect to a biased phenomenological noise model. We also use a matching decoder to demonstrate the advantages of the XZZX code at low error rates.

The minimum-weight perfect matching algorithm takes a graph with weighted edges and returns a perfect matching using the edges of the input graph such that the sum of the weights of the edges is minimal. To employ this algorithm for decoding the XZZX code we prepare an input graph by assigning each defect a vertex of a complete graph, and we weight each edge according to the logarithm of the probability that the error model introduced the pair of defects connected by the edge. Our decoder is discussed in detail in Sec. C. The edges returned in the perfect matching correspond to the defects that should be paired by the correction.

We test the performance of the decoder using two noise models: the biased noise model parameterized by η as we defined above, and a phenomenological biased noise model where qubits experience *high-rate* Pauli-Z errors with probability $p_{\text{h.r.}}$ per unit time, and *low-rate* Pauli X or Pauli-Y errors each occurring with probability $p_{\text{l.r.}}$ per unit time. The noise bias with this phenomenological noise model is defined as $\eta = p_{\text{h.r.}}/(2p_{\text{l.r.}})$. One time unit is the time it takes to make a stabilizer measurement, and we assume we can measure all the stabilizers in parallel. Each stabilizer measurement returns the incorrect outcome with probability $q = p_{\text{h.r.}} + p_{\text{l.r.}}$. To leading order, this measurement error rate is consistent with a measurement circuit where an ancilla is prepared in the state $|+\rangle$ and subsequently entangled to the qubits of S_f with bias preserving controlled-not and controlled-phase gates before its measurement in the Pauli-X basis. With such a circuit, Pauli-Y and Pauli-Z errors on the ancilla will alter the measurement outcome. At $\eta = 1/2$ this noise model interpolates to a conventional noise model where $q = 2p/3$ [36]. We also remark that hook errors [36, 37] are low-rate events with this circuit as the control qubit of entangling gates commutes with the high-rate Pauli-Z errors.

Using this phenomenological noise model we repeat measurements over d units of time, and we define a defect as the parity of the outcomes of two sequential measurements of each stabilizer. This allows us to adapt the minimum-weight perfect-matching algorithm for decoding using the prescription given above.

Intuitively, the decoder will preferentially pair defects along the diagonals associated with the dominant error. In the limit of infinite bias, the decoder effectively corrects the Pauli-Z errors as d independent repetition

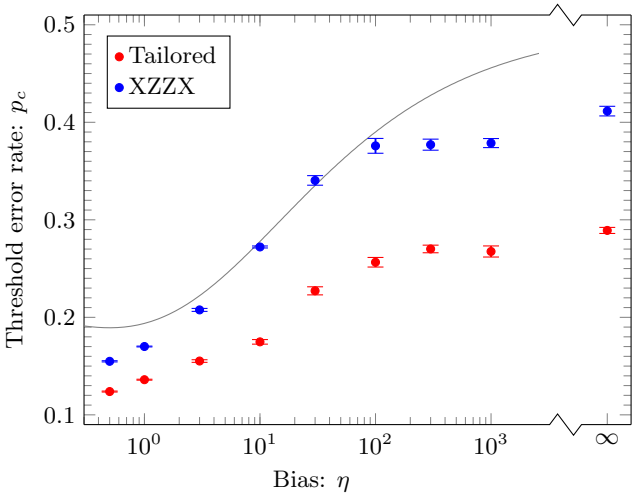

FIG. 4. Threshold values for square $L \times L$ lattices with periodic boundary conditions using MWPM for the XZZX (blue) and X/Y tailored (red) [8] surface code architectures, as a function of noise bias η. The zero rate hashing bound is shown in grey. All thresholds are calculated with $N = 30000$ Monte Carlo simulations for lattice dimensions $L = 24, 28, 32, 36, 40$ for $\eta = 0.5, 1, 3, 10, 30, 100, 300, 1000$ and $L = 48, 56, 64, 72, 80$ for $\eta = \infty$.

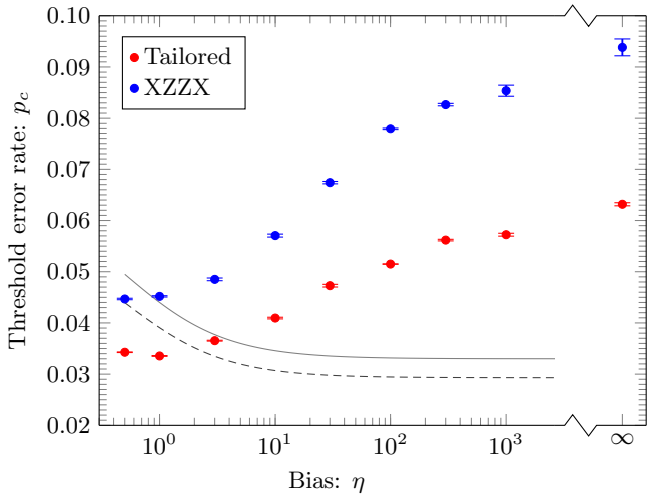

FIG. 5. Fault tolerant threshold values for square $L \times L$ lattices with periodic boundary conditions using MWPM for the XZZX (blue) and Tailored (red) [8] surface code architectures. The noise model used has the measurement errors occurring at a rate of $q = p_{\text{h.r.}} + p_{\text{l.r.}}$. The solid curve depicts the theoretical optimal performance of a 2-sector decoder which deals with bit-flip noise and dephasing noise independently. η quantifies the bias towards dephasing noise. The blue and red points are calculated with N=30000 Monte Carlo simulations for lattice dimensions $L = 12, 14, 16, 18, 20$ for $\eta = 0.5, 1, 3, 10, 30, 100, 300, 1000$ and $L = 24, 28, 32, 36, 40$ for $\eta = \infty$.

codes for the ideal measurement case and d decoupled copies of the two-dimensional surface code for the infinite bias phenomenological noise model. In both the case of phenomenological noise, and the case with ideal measurements, when $\eta = 1/2$ the minimum-weight perfect-matching decoder is equivalent to the conventional decoder presented throughout the literature, e.g. [12, 38]. We use these observations to check that our decoder behaves correctly in these limits.

In Figs. 4 and 5 we present our thresholds for the idealized biased noise model with $q = 0$, and phenomenological biased noise model with $q = p_{\text{h.r.}} + p_{\text{l.r.}}$, respectively, as a function of η. We find for the ideal case, our decoder closely follows the hashing bound at high bias. We observe our data drops below the hashing bound at very high bias, but we attribute this to a small size effect since the success of the decoder depends on effectively decoding $\sim d$ independent copies of the repetition code. In the fault tolerant case, we find our decoder tends towards a threshold of $\sim 10\%$. In both cases we find our decoder significantly surpasses the thresholds found using the decoder presented in Ref. [8], shown by red data points.

V. SUB-THRESHOLD SCALING

We now show that the exceptional error thresholds of the XZZX surface code are accompanied by significant advantages in terms of overheads for fault-tolerant quantum computing. Specifically, we evaluate the logical failure rate that a tailored code can achieve below threshold for a biased noise model with both high- and low-rate errors. In Ref. [7] it was demonstrated that at infinite bias a code that can tolerate $\sim d^2$ dephasing errors has a logical failure rate scaling like $O(p_{\text{h.r.}}^{d^2/2})$. In practice, at finite bias, other mechanisms can cause logical failures. For instance, low-rate errors can align along a path of length $\sim d$. In general, the common failure mechanisms for this code will depend on the system parameters. We use the XZZX code with periodic boundary conditions and dimensionality $d \times (d+1)$ in the ideal measurement setting to identify two regimes where the two aforementioned failure mechanisms are dominant.

Let us examine the different failure mechanisms for the XZZX code. Like the coprime codes in Ref. [7], the single logical qubit of this code has a least-weight logical operator of Pauli-Z terms supported on $O(d^2)$ qubits. We therefore expect a failure due to only high-rate errors to occur with probability

$$\overline{P}_{\text{h.r.}} \sim N_{\text{h.r.}} p_{\text{h.r.}}^{(d^2+1)/2}, \tag{2}$$

at low p where $N_{\text{h.r.}} \sim 2^{d^2}$ is the number of configurations that can cause a failure. To identify two distinct regimes, we compare this probability to the probability of a logical failure due to a string of $d/4$ high rate errors and $d/4$ low rate errors. We thus consider the ansatz

$$\overline{P}_{\text{l.r.}} \sim N_{\text{l.r.}} (1-p)^{n-d/2} (p_z + p_y)^{d/4} (p_x + p_y)^{d/4} \tag{3}$$

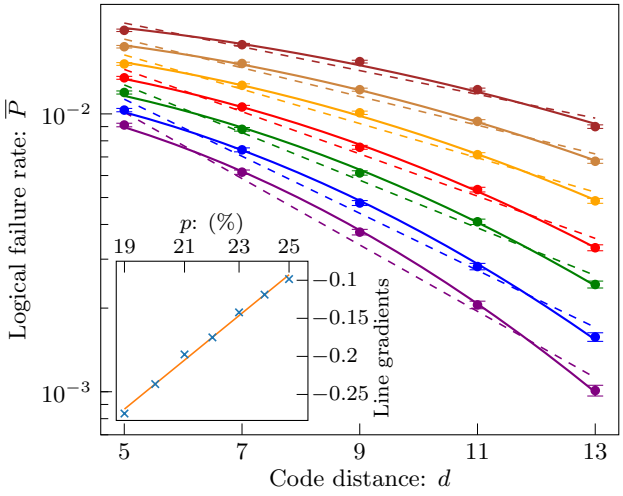

FIG. 6. Depiction of quadratic scaling achieved with coprime $d \times (d+1)$ codes for $d = 5, 7, 9, 11$ and 13 at bias $\eta = 300$. The data was collected using $N = 500000$ iterations of MC simulations. The physical error rates used are, from the bottom to the top curves in the main plot, $p = 0.19, 0.20, 0.21, 0.22, 0.23, 0.24$ and 0.25. The solid lines are a fit of the data to $\overline{P} = Ae^{Bd^2}$ and the dashed lines a fit to $\overline{P} = Ae^{Bd}$. The quadratic trend observed is supported by the gradients of the best fit dashed lines, showed on the inset plot. The best-fit line to these gradients, as calculated using least-squares, has gradient 0.643 exceeding the required 0.5 consistent with $O(d)$ scaling for this MWPM decoder.

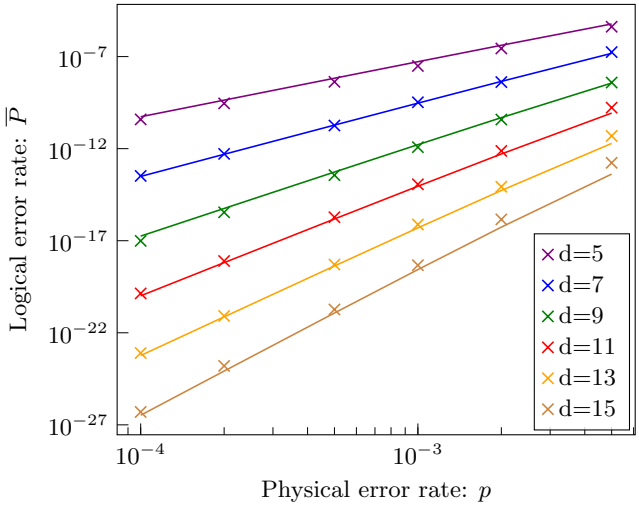

FIG. 7. The data (crosses) was collected at bias $\eta = 3$ and coprime $d \times (d+1)$ code dimensions of $d = 5, 7, 9, 11, 13$ and 15 using the metropolis algorithm and splitting method presented in [40, 41] as Monte Carlo is intractable at this low physical error rate. The solid lines represent the prediction of Eqn. 3. As p decreases the agreement between data and ansatz approaches unity.

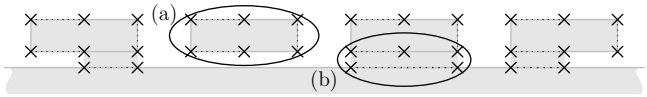

FIG. 8. Fault-tolerant quantum computation by generalised lattice surgery [29, 47]. Pairs of qubits are encoded on surface codes with six twist defects lying on their boundaries. This layout allows a large ancillary surface code, shown at the bottom of the figure, from performing arbitrary multi-qubit Pauli matrices between the encoded qubits via code deformation. Here we will focus on the initialisation of a surface code, (a), and a single code deformation procedure, (b), to show a single lattice surgery procedure.

where $N_{\text{l.r.}} \sim 2^{\gamma d}$ is an entropy term with $3/2 \lesssim \gamma \lesssim 2$ [39]. We justify this ansatz in the Methods.

We first look at the regime where $\overline{P}_{\text{h.r.}} \gg \overline{P}_{\text{l.r.}}$ where we expect that the logical failure rate will decay super exponentially in code distance d. Fig. 6 depicts the data for $\eta = 300$ with lattices of distance $d = 5, 7, 9, 11, 13$ and physical error rate $0.19 \leq p \leq 0.25$. Our data shows good agreement with the fitting depicted by solid lines that follow a quadratic scaling Ansatz where $\overline{P} = Ae^{Bd^2}$. We also compare our data to a fit linear scaling where $\overline{P} = Ae^{Bd}$. This fitting shows that our data is inconsistent with a model where $\sim d$ high-rate errors can cause a logical failure. Indeed, the inset shows a plot of the gradient of each fitting shown as a function of p. Fitting to these data points we find a gradient $\sim 0.67 \gg 1/2$. We would expect a gradient of $1/2$ for a model where the logical failure rate decayed like $\overline{P} \sim p_{\text{h.r.}}^{d/2}$ where $d/2$ high rate errors might cause the decoder to fail.

At small p and modest η where $\overline{P}_{\text{l.r.}} \gg \overline{P}_{\text{h.r.}}$ we expect the dominant source of failure to be an error of weight $O(d)$. In Fig. 7 we show the above ansatz fit to the data with $\gamma \sim 1.8$. The data were collected using the splitting method and metropolis sampling algorithm presented in [40, 41]. This remarkable correspondence to our data shows that our decoder is capable of decoding up to $\sim d/4$ low rate errors, even with a large fraction of high-rate errors will commonly occur on the lattice.

VI. LOW-OVERHEAD FAULT-TOLERANT QUANTUM COMPUTATION

Let us now consider how to perform fault-tolerant quantum computation using code deformations [42–44] with surface codes tailored to deal with bias. The idea is to perform stabilizer measurements that project the surface code onto a different encoding where the logical information has been displaced over the lattice of qubits. These deformations can be used to perform Clifford operations with the surface code. Together with the noisy initialisation of magic states to perform magic state distillation [45], Clifford gates are sufficient for universal quantum computation. Although initialisation circuits have been proposed to exploit noise bias [46], here we focus on fault-tolerant Clifford operations and the fault-tolerant preparation of ancilla qubits in the computational basis.

Many approaches for code deformations have been proposed that, in principle, could be implemented in a way

to take advantage of bias using a tailored surface code. These approaches include braiding punctures [42–44, 48], lattice surgery [27, 47, 49] and computation with twist defects [28, 50, 51]. Moreover, all of these methods can be be unified [28]. Here let us focus on a single example proposed by Litinski in Refs. [29, 47]; see Fig. 8 and its caption for a brief review. Because the general methods are well known, we will provide only a high-level overview. We leave open the interesting question of implementing these ideas and concretely estimating thresholds for fault-tolerant quantum computation using this scheme.

Our layout for fault-tolerant quantum computation requires the fault-tolerant initialization of a hexon surface code, i.e., a surface code with six twist defects at its boundaries; see Fig. 9. Note that the lattice geometry has been rotated; see a stabilizer in Fig. 9(a). We will justify this choice shortly. We can fault-tolerantly initialize this code in eigenstates of the computational basis by first preparing its qubits in a product state before we begin measuring its stabilizers. We must check that we correct the errors that may occur before we begin measuring the XZZX surface code stabilizers.

We initialize the qubits of the hexon surface code in eigenstates of Pauli-X and Pauli-Z. In Fig. 9 we show qubits initialized in the +1 eigenvalue eigenstate of Pauli-X (Pauli-Z) in red (blue). This choice initializes the system in the +1 eigenvalue eigenstate of the stabilizers on shaded plaquettes, e.g. Fig. 9(a), as well as the +1 eigenvalue eigenstate of the two logical Pauli-Z operators, e.g., 9(b). Measuring the stabilizers of the XZZX surface code will project the system onto an eigenstate of the stabilizers of the unshaded plaquettes. We remark that the reverse operation, where we measure qubits of the XZZX surface code in this same product basis, will read the code out while respecting the properties required to be robust to the noise bias.

Let us check that this method of initialization is robust to our biased noise model. We must be able to correct high-rate dephasing errors on the red qubits. On the other hand, Pauli-Z errors act trivially on the blue qubits in eigenstates of the Pauli-Z operator during preparation. Similarly, we should be weary of low-rate Pauli-X errors being introduced to these qubits. Given that the initial state is already in an eigenstate of some of the stabilizers of the XZZX-surface code, we can detect Pauli-Z errors on the red qubits, see, e.g. 9(c). The shaded plaquettes will identify defects due to the Pauli-Z errors. Moreover, as we discussed before, strings created by Pauli-Z errors align along horizontal lines using the XZZX-surface code. We therefore benefit from the advantages of the XZZX surface code for dealing with bias errors during initialization.

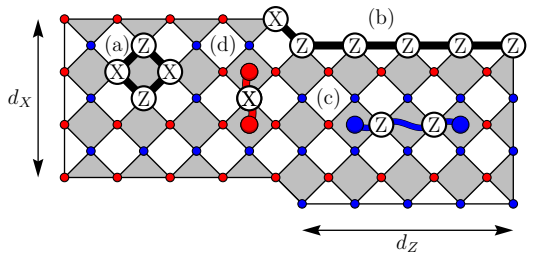

FIG. 9. Initialising a hexon surface code. Red(blue) vertices are initialised in Pauli-X(Pauli-Z) basis. The system is prepared in an eigenstate of shaded plaquettes (a) and the logical Pauli-Z operators, (b). This initialisation strategy is robust to biased noise. Pauli-Z errors that can occur on red vertices are detected by the shaded plaquettes (c). We can also detect low-rate Pauli-X errors on blue vertices with using this method of initialisation, see (d).

Low rate errors can also be detected on blue qubits, as in Fig. 9(d). The bit-flip errors violate the stabilizers we initialize when we prepare the initial product state. As such we can detect these errors. We note that with this lattice geometry, as the low rate error strings align along the vertical direction only, we can change the dimensions of the lattice without compromising the performance of the code significantly. In practice, we can have that $d_X \ll d_Z$. This choice may have a dramatic effects on the resource scaling of large scale quantum computation. We will make estimates on the optimal choice of d_Z/d_X later where we discuss small codes [52].

Lattice surgery amounts to initializing and reading out different patches of a large surface code lattice to perform fault-tolerant parity measurements between logical qubits. As such demonstrating that we can perform arbitrary lattice surgery operations while preserving the protection that the XZZX surface code offers to bias does not become more complicated than what we have already demonstrated. The one exception is where we perform a logical Pauli-Y measurement. In this case, we introduce a twist to the lattice [50] and, as such, we need to reexamine the symmetries of the lattice. We show the twist in the centre of Fig. 10 together with its weight-five stabilizer operator. We shade the stabilizers that are members of the symmetry with respect to Pauli-Z errors in a lighter color. We find that the twist introduces a branch in the one-dimensional symmetries of the XZZX surface code. It should be straightforward to adapt a minimum-weight perfect-matching decoder to account for this branch. Moreover, we do not expect that a branch on a single location on the lattice will have a significant impact on the performance of the code given that, even with a twist on the lattice, the majority of the lattice is decoded as a series of one-dimensional repetition codes in the infinite bias limit.

VII. DISCUSSION

We have shown how fault-tolerant quantum architectures based on the XZZX surface code yield remarkably high memory thresholds and low overhead as compared with the conventional surface code approach. Our gener-

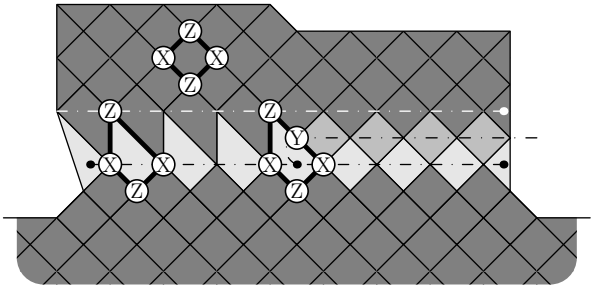

FIG. 10. The hexon surface code fused to the ancillary surface code to perform a logical Pauli-Y measurement. The lattice surgery procedure introduces a twist in the centre of the lattice. We show the symmetry with respect to the Pauli-Z errors by lightly colored plaquettes. Again, decoding this model in the infinite bias limit is reduced to decoding one-dimensional repetition codes, except at the twist where there is a single branching point.

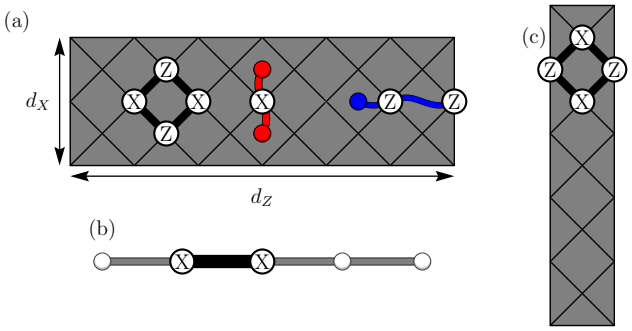

FIG. 11. Some small codes; an error-correcting surface code of dimensions 3×7, a repetition code, which can be regarded as a 1×5 surface code in this family of rectangular surface codes. An error detecting surface code with distance 2 for only low-rate errors.

alised fault-tolerant decoder offers these advantages over a broad range of biased error models representing what is observed in experiments for a variety of physical qubits. The code capacity thresholds of the XZZX code match the performance of random coding (hashing) theory, suggesting that this code may be approaching the limits of what is possible. In contrast to this expectation, the XZZX surface code threshold is numerically observed to exceed this hashing bound for certain error models, opening the enticing possibility that random coding is not the limit for practical thresholds. The highest achievable error threshold for fault-tolerant quantum computing remains an open question.

We emphasize that the full potential of our results lies not just in the demonstrated advantages of using this particular architecture, but rather the indication that further innovations in codes and architectures may still yield significant gains in thresholds and overheads. We have shown that substantial gains on code capacity thresholds can be found when the code and decoder are tailored to the relevant noise model. While the standard approach to decoding the surface code considers Pauli-X and Pauli-Z errors separately, our results demonstrate that this approach is sub-optimal except at the exceptional point where these error rates are exactly equal and uncorrelated. There is a clear avenue to generalize our methods and results to the practical setting involving correlated errors arising from more realistic noise models as we perform fault-tolerant logic. We suggest that the theory of symmetries [8, 30] may offer a formalism to make progress in this direction.

Because our decoder is based on minimum-weight matching, there are no fundamental obstacles to adapt it to the more complicated setting of circuit noise [36, 43, 53]. We expect that the high numerical thresholds we observe for phenomenological noise will, when adapted to circuit level noise, continue to outperform the conventional surface code, especially when using gates that preserve the structure of the noise [25, 26]. We expect that the largest performance gains will be obtained by using information from a fully characterized Pauli noise model [54] that goes beyond the single-qubit error models considered here.

Along with high thresholds, the XZZX surface code architecture can yield significant reductions in the overheads for fault-tolerant quantum computing, through improvements to the sub-threshold scaling of logical error rates. It is in this direction that further research into tailored codes and decoders may provide the most significant advances, bringing down the astronomical numbers of physical qubits needed for fault-tolerant quantum computing. A key future direction of research would be to carry these improvements over to codes and architectures that promise improved (even constant) overheads [55, 56]. Recent research on fault-tolerant quantum computing using low-density parity check (LDPC) codes that generalize concepts from the surface code [57–59] provide a natural starting point.

Given that the XZZX code has the same hardware requirements as the conventional surface code, it is natural to consider realizing it on current generation quantum computers. As in Fig. 11, there are several small-scale experiments that might showcase the benefits of the XZZX code. Indeed, the smallest instance of a rotated variant of the XZZX code that can take advantage of biased noise is a repetition code. We therefore see that this rectangular variation of the code may offer us a gentle path for the experimental development towards the first generation of large-scale quantum computers.

ACKNOWLEDGMENTS

We are grateful to A. Darmawan, A. Grimsmo and S. Puri for discussions, and especially to J. Wootton for recommending consideration of the XZZX code for biased noise [60]. This work is supported by the Australian Research Council via the Centre of Excellence in Engineered Quantum Systems (EQUS) project number

CE170100009. BJB also received support from the University of Sydney Fellowship Programme. Access to high-performance computing resources was provided by the National Computational Infrastructure (NCI Australia), an NCRIS enabled capability supported by the Australian Government, and the Sydney Informatics Hub, a Core Research Facility of the University of Sydney.

Appendix A: Threshold estimation

For the estimation of a threshold error rate p_c, we use the critical exponent method of Ref. [38]. According to this method, if we define a correlation length $\xi = (p - p_c)^{-\nu}$ in terms of physical error rate p and some critical exponent ν, then, for sufficiently large code distance d, we expect the logical failure rate f to depend only on the dimensionless ratio d/ξ. Consequently, we define a rescaled variable $x = (d/\xi)^{1/\nu} = (p - p_c)d^{1/\nu}$ such that the failure rate, expanded as a power series in x, is explicitly scale invariant at the critical point p_c corresponding to $x = 0$. It is then natural to model the failure rate by a truncated Taylor expansion about p_c. We use a quadratic model, $f = A + Bx + Cx^2$, and then fit to this model to find p_c, ν and the nuisance parameters A, B, C. See Ref. [61] for a discussion of the validity of this method. Unless otherwise stated, error bars are obtained by jackknife resampling, i.e. the standard deviation in threshold estimates when reapplying the fitting method with a single code distance d removed, over all simulated code distances d. As a representative illustration of this method, we estimate the threshold error rate p_c of the XZZX surface code under X-biased noise with $\eta = 30$, assuming perfect measurements in Fig. 12.

Appendix B: Optimal thresholds methods

For the simulations in Section III, we use maximum-likelihood decoding in order to highlight features of the codes independent of any particular heuristic decoding algorithm. Maximum-likelihood decoding, which selects a correction from the most probable logical coset of error configurations consistent with a given syndrome, is, by definition, optimal. Exact evaluation of the coset probabilities is, in general, inefficient. An algorithm due to Bravyi, Suchara, and Vargo [35] efficiently approximates maximum-likelihood decoding by mapping coset probabilities to tensor-network contractions. Contractions are approximated by reducing the size of the tensors during contraction through Schmidt decomposition and retention of only the χ largest Schmidt values. This approach, appropriately adapted, has been found to converge well with modest values of χ for a range of Pauli noise channels and surface code layouts [7, 35]. A full description of the tensor network used in our simulations with the rotated CSS surface code (see Fig. 13, left) is provided in Ref. [7]; adaptation to the XZZX surface code (see Fig. 13, right) is a straightforward redefinition of tensor element values for the uniform stabilizers.

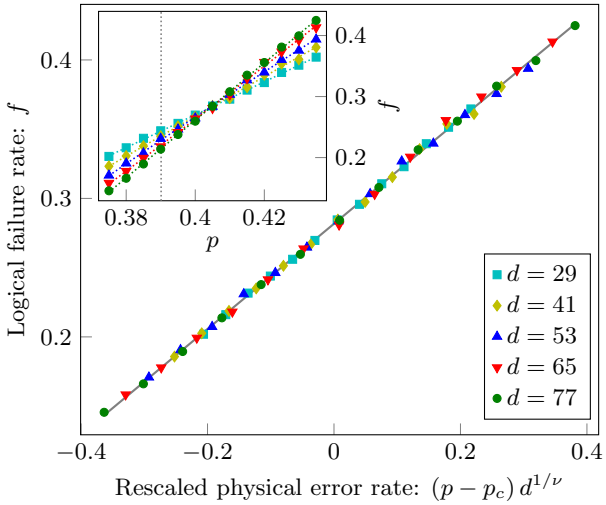

FIG. 12. Illustration of the critical exponent method of threshold estimation. Here we consider the XZZX surface code under X-biased noise with $\eta = 30$, assuming perfect measurements. We plot logical failure rate f as a function of the rescaled physical error rate $x = (p - p_c)d^{1/\nu}$ for a range of code distances d. The solid line is the best fit to the model $f = A + Bx + Cx^2$, which yields $p_c = 40.4(1)\%$. The inset shows the raw sample means over 60 000 runs for various values of physical error rate p, and the dotted gray vertical line indicates the zero-rate hashing bound.

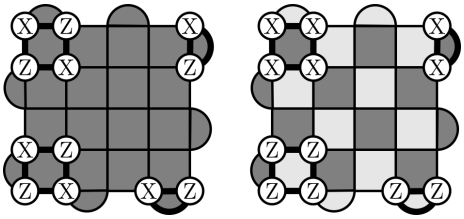

FIG. 13. (Left) XZZX surface code with boundaries. Qubits are on vertices and stabilizers apply uniformly around all faces. (Right) CSS surface code with boundaries. Qubits are on vertices and X-type (Z-type) stabilizers apply around dark (light) faces. In both codes, boundary stabilizers are defined by restricting to two qubits.

Figure 2, which shows threshold values over all single-qubit Pauli noise channels for CSS and XZZX surface codes, is constructed as follows. Each threshold surface is formed using Delaunay triangulation of 211 threshold values. Since both CSS and XZZX surface codes are symmetric in the exchange of Pauli X and Z, 111 threshold values are estimated for each surface. Sample noise channels are distributed radially such that the spacing reduces quadratically towards the sides of the triangle representing all single-qubit Pauli noise channels, see Fig. 14. Each threshold is estimated over four code distances $d \in \{13, 17, 21, 25\}$, at least six physical error

probabilities, and 30 000 simulations per code distance and physical error probability. In all simulations a tensor-network decoder approximation parameter of $\chi = 16$ is used to achieve reasonable convergence over all sampled single-qubit Pauli noise channels for the given code sizes.

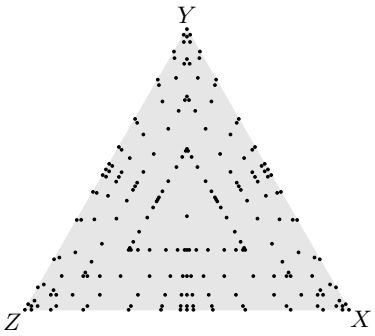

FIG. 14. Distribution of 211 samples over the surface of all single-qubit Pauli noise channels. By code symmetry, thresholds are estimated for 111 of these samples to construct each threshold surface seen in Fig. 2.

Figure 3, which investigates threshold estimates exceeding the hashing bound for the XZZX surface code with X-biased noise, is constructed as follows. For bias $30 \leq \eta \leq 1000$, where XZZX threshold estimates exceed the hashing bound, we run compute-intensive simulations; each threshold is estimated over sets of four code distances up to $d \in \{65, 69, 73, 77\}$, at least fifteen physical error probabilities, and 60 000 simulations per code distance and physical error probability. Interestingly, for the XZZX surface code with X-biased noise, we find the tensor-network decoder converges extremely well, as summarized in Fig. 15 for code distance $d = 45$, allowing us to use $\chi = 8$. For $\eta = 30$, the shift in logical failure rate between $\chi = 8$ and the largest χ shown is less than one tenth of a standard deviation over 30 000 simulations, and for $\eta > 30$ the convergence is complete. All other threshold estimates in Fig. 3, are included for context and use the same simulation parameters as described above for Fig. 2.

Appendix C: The minimum-weight perfect-matching decoder

The technique of decoding by matching defects in the syndrome state of a code is generally accepted to be one of the leading procedures to achieve efficient quantum error correction and one which is readily scalable to the fault tolerant regime [5, 12, 30, 38]. An undirected complete graph is formed with the defects as vertices and with edge weights characterised by a distance metric between two defects, whose particularities depend on the error model in consideration. The grouping of defects is realised by solving the discrete combinatorial optimisation problem of minimum weight perfect matching (MWPM)

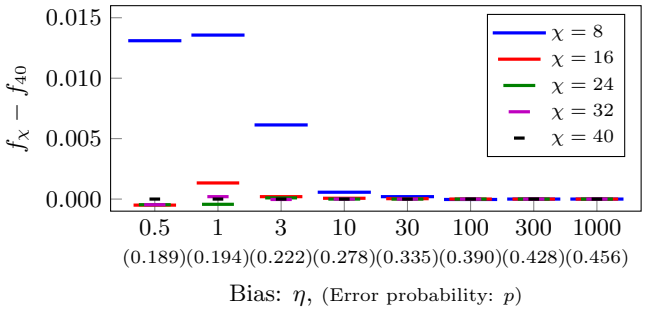

FIG. 15. Tensor-network decoder convergence for the 45×45 XZZX surface code with X-biased noise, represented by shifted logical failure rate $f_\chi - f_{40}$, as a function of χ at a physical error probability p near the hashing bound for the given bias η. Each data point corresponds to 30 000 runs with identical errors generated across all χ for a given bias.

[62, 63]. Two matched defects create a path of physical qubits between them through which we apply Pauli gates to bring the system back to the code space, from which an inference about the logical error state of the system can be devised.

Denote by L_i the diagonals that Z creates plaquette defects though (gray line in 1) and let L_i' be the orthogonal diagonals. Note that due to the periodic boundary conditions, for an $L \times L$ lattice there are L such L_i diagonals.

First consider the idealised purely dephasing noise channel $\mathcal{E}^Z = \langle Z_v \rangle$ on the square $L \times L$ rotated surface code with periodic boundary conditions from Fig. 1.

Any error drawn from $\mathcal{E}^Z$ respects the L_i diagonal symmetries of this code. Indeed, $S = \prod_{f \in L_i} S_f$ is a symmetry with respect to $\mathcal{E}^Z$ since it can be shown that it is a product of Z operators which trivially commutes with any errors drawn from $\mathcal{E}^Z$. We turn the standard 2D matching problem across the whole surface of the code into L 1D matching problems along the diagonals L_i. We can always match defects along these diagonals perfectly because any error $E \in \mathcal{E}^Z$ always produces an even number of defects along each of the diagonals L_i. Matching defects from homogeneous Pauli errors along a single diagonal in this periodic structure is topologically equivalent to matching defects from bit-flip noise on physical qubits arranged in a circle, which is the repetition code.

The parallelized decoding procedure described vastly improves the speed of decoding, especially when exploiting the modern multi-processor classical computational environments.

Next, consider a model of noise biased toward dephasing, $\mathcal{E}^Z(\eta) = \langle Z_v, X_v, Y_v \rangle$ where Z errors occur at a higher rate, $p_{\text{h.r.}}$, than X and Y errors and the latter two occur at the same rate, $p_{\text{l.r.}}$. If p is the rate of physical errors then

$$p_{\text{h.r.}} = \frac{p\eta}{\eta + 1} \quad \text{and} \quad p_{\text{l.r.}} = \frac{p}{2(1+\eta)},$$

where η is the bias of the model. For $\eta = 0.5$ we obtain

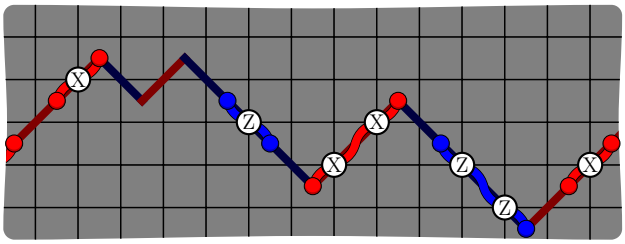

FIG. 16. A highly probable error configuration that will lead to a logical failure. The error consists of $\sim d/4$ high rate errors and $\sim d/4$ low rate errors along the support of a weight d logical operator.

the depolarising noise model, where Z, X and Y errors occur at the same rate, and in the limit of $\eta \to \infty$ we obtain the purely dephasing noise model $\mathcal{E}^Z$.

To perform MWPM under this noise model we note that the nature of high rate Z errors is to extend plaquette defects along the diagonals L_i whereas lower rate errors extend the plaquette defects along the orthogonal diagonals, L_i'. Hence, we favour matching along the L_i diagonals and penalise matching along the L_i' diagonals. We decrease the penalisation of L_i' matching as the rate of X and Y errors increase. This is done by giving a higher weight, $w_{l.r.}$, to the L_i' diagonal steps between the 2 defects and a lower weight, $w_{h.r.}$, to the L_i diagonal steps between the two defects, as given by [8]:

$$w_{l.r.} = -\log(\frac{p_{l.r.}}{1-p}), \quad w_{h.r.} = -\log(\frac{p_{h.r.}}{1-p}) \tag{C1}$$

where $w_{h.r.} \geq w_{h.r.}$ with equality if and only if $\eta = 0.5$. The total weight assigned to a pair of defects u and v is therefore: $\omega(u,v) = w_{h.r.}H(u,v) + w_{l.r.}L(u,v)$ where $H(u,v)$ and $L(u,v)$ are the number of steps along L_i diagonals and L_i' diagonals respectively which compose the path between u and v.

Once the complete graph is formed with the defects as vertices and the edge weights as described above we use Blossom V [63] to perform the MWPM sub-routine to obtain the pairing of defects. For the path between defects in each pair we apply Z and X operators to those qubits traversed along L_i and L_i' diagonals respectively. This matching procedure leaves the code devoid of plaquette defects and therefore in the code space from which the logical error state of the code is deduced.

MWPM on the XZZX code naturally extends to the fault tolerance regime by repeating stabiliser measurements over time and stacking the syndrome state of the code at each time stamp [12]. This builds a $2+1$ dimensional structure with the 2D spatial dimension characterised by the surface of the code and the orthogonal time direction.

A complete graph is formed with the defects in the $2+1$D structure as vertices and the edge weight between defects u and v takes the form $\omega_t(u,v) = \omega(u,v) + w_t T(u,v)$ where $\omega(u,v)$ is the edge weight in the spatial direction between u and v, $T(u,v)$ is the time difference between the location of u and v and $w_t = -\log\left(\frac{q}{1-q}\right)$ [8], with q being the probability of a stabiliser measurement error on any one trial. We set $q = p_{\text{h.r.}} + p_{\text{l.r.}}$ which interpolates to standard MWPM on the surface code for the depolarising noise channel ($\eta = 0.5$) [38].

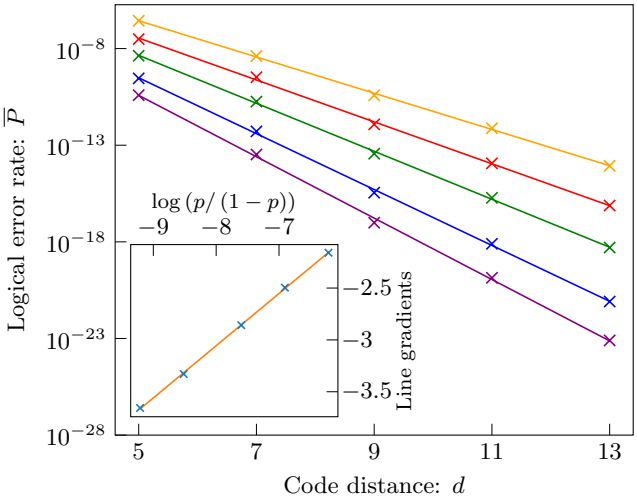

FIG. 17. Plot showing logical failure rate $\overline{P}$ as a function of code distance d for data where $\eta = 3$. The physical error rates used are, from the bottom to the top curves in the main plot, $0.0001, 0.0002, 0.0005, 0.001$ and 0.002. We estimate $G(p,\eta)$ for each value of p by taking the gradient of each line. Low error rate data is collected using the method proposed in Refs. [40, 41]. The inset plot shows the gradients $G(p,\eta)$ as function of $\log[p/(1-p)]$ for $\eta = 3$. Values for $G(p,\eta)$, see Eqn. (D2), are estimated using the linear fittings shown in Fig. 17. Gradient of line of best fit to these data points is 0.503(4) in agreement with the expected gradient 1/2.

Appendix D: Ansatz at low error rates

In the main text we proposed a regime at low error rates where the most common cause of logical failure is a sequence of $\sim d/4$ low-rate and high-rate errors along the support of a weight d logical operator, see Fig. 16. We model the logical failure rate in this regime with Eqn. (3). Here we justify this Ansatz and compare the model with numerical data.

We take the logarithm of Eqn. (3) to obtain

$$\log \overline{P} \sim n\log(1-p) + \gamma d \log 2 + (d/2)\log\left[\frac{p}{1-p}\right] + (d/4)\log\left[\frac{\eta+1/2}{(\eta+1)^2}\right]. \tag{D1}$$

Neglecting the small term $n\log(1-p)$ we can express the this equation as $\log \overline{P} \approx G(p,\eta)d$ where we have the

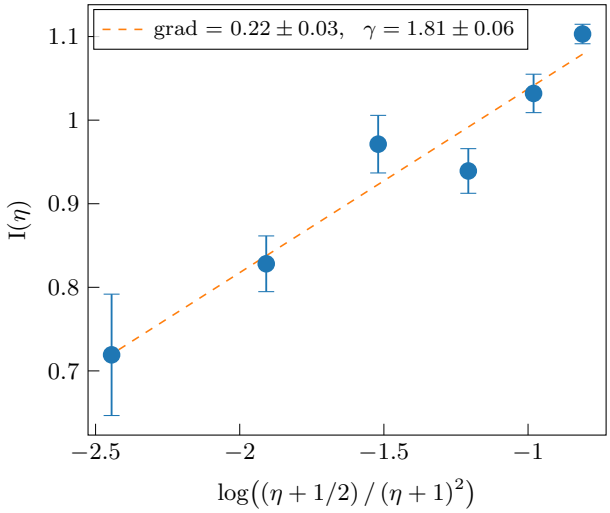

FIG. 18. Plot showing intercepts $I(\eta)$ shown as a function of $\nu = \log[(\eta+1/2)/(\eta+1)^2]$. The intercept function is defined in Eqn. D3 and estimated from the intercept of lines such as that shown in the inset of Fig. 17. The gradient 0.22 is consistent with a gradient 1/4 is consistent with the Ansatz that predicts $d/4$ low-rate errors typically causing logical errors. Moreover, the intercept $\sim 1.17 \approx 1.7 \log 2$ is consistent with the entropy term we expect with $3/2 \leq \gamma \leq 2$.

gradient

$$G(p,\eta) = \frac{1}{2}\log\left[\frac{p}{1-p}\right] + \gamma \log 2 + \frac{1}{4}\log\left[\frac{\eta+1/2}{(\eta+1)^2}\right]. \tag{D2}$$

In Fig. 17 we plot the data shown in the main text in Fig. 7 as a function of d to read the gradient $G(p,\eta)$ from the graph. We then plot $G(p,\eta)$ as a function of $\beta = \log[p/(1-p)]$ in the inset of Fig. 17. The plot reveals a gradient ~ 0.5, consistent with the Ansatz model where we expect a gradient 1/2.

Interrogating the data further we find good agreement between our data and the physics we expect in this regime. Expressing $G(p,\eta) = \beta/2 + I(\eta)$ where $\beta = \log[p/(1-p)]$ and

$$I(\eta) = \gamma \log 2 + \frac{1}{4}\log\left[\frac{\eta+1/2}{(\eta+1)^2}\right], \tag{D3}$$

we find a function of η and γ. We estimate $I(\eta)$ using the intercepts from figures, such as that the inset of Fig. 17, and plot it as a function of $\log[(\eta+1/2)/(\eta+1)^2]$. We show this plot in Fig. 18. From this plot we find a line of best fit with gradient ~ 0.22, within the error bars of 1/4. This is consistent with typical error configurations that lead to logical failure with $\sim d/4$ low-rate errros. Moreover, we find an intercept $\sim 1.25 \approx 1.8 \log 2$, which is consistent with $3/2 \leq \gamma \leq 2$, as we expect [39].

[1] P. W. Shor, in *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, FOCS '96 (IEEE Computer Society, USA, 1996) p. 56.
[2] D. Aharonov and M. Ben-Or, in *STOC '97 Proceedings of the twenty-ninth annual ACM symposium on Theory of computing* (1997) p. 176.
[3] E. Knill, R. Laflamme, and W. Zurek, quant-ph/9610011 (1996).
[4] A. Y. Kitaev, Russian Math. Surveys **52**, 1191 (1997).
[5] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Phys. Rev. A **86**, 032324 (2012).
[6] D. K. Tuckett, S. D. Bartlett, and S. T. Flammia, Phys. Rev. Lett. **120**, 050505 (2018).
[7] D. K. Tuckett, A. S. Darmawan, C. T. Chubb, S. Bravyi, S. D. Bartlett, and S. T. Flammia, Phys. Rev. X **9**, 041031 (2019).
[8] D. K. Tuckett, S. D. Bartlett, S. T. Flammia, and B. J. Brown, Phys. Rev. Lett. **124**, 130501 (2020).
[9] C. Gidney and M. Ekerå, How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits (2019), arXiv:1905.09749 [quant-ph].
[10] A. Y. Kitaev, Ann. Phys. **303**, 2 (2003).
[11] S. B. Bravyi and A. Y. Kitaev, arXiv:quant-ph/9811052 (1998).
[12] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, J. Math. Phys. **43**, 4452 (2002).
[13] X.-G. Wen, Phys. Rev. Lett. **90**, 016803 (2003).
[14] M. Li, D. Miller, M. Newman, Y. Wu, and K. R. Brown, Physical Review X **9**, 021041 (2019), arXiv:1809.01193.
[15] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters, Physical Review A **54**, 3824 (1996), arXiv:quant-ph/9604024.
[16] M. M. Wilde, *Quantum Information Theory*, 2nd ed. (Cambridge University Press, 2017).
[17] P. W. Shor and J. A. Smolin, Quantum error-correcting codes need not completely reveal the error syndrome (1996), arXiv:quant-ph/9604006 [quant-ph].
[18] D. P. DiVincenzo, P. W. Shor, and J. A. Smolin, Phys. Rev. A **57**, 830 (1998).
[19] G. Smith and J. A. Smolin, Physical Review Letters **98**, 030501 (2007), arXiv:quant-ph/0604107.
[20] J. Fern and K. B. Whaley, Physical Review A **78**, 062335 (2008), arXiv:0708.1597.
[21] J. Bausch and F. Leditzky, Error thresholds for arbitrary pauli noise (2019), arXiv:1910.00471.
[22] R. Lescanne, M. Villiers, T. Peronnin, A. Sarlette, M. Delbecq, B. Huard, T. Kontos, M. Mirrahimi, and Z. Leghtas, Nature Physics **16**, 509 (2020), arXiv:1907.11729.
[23] A. Grimm, N. E. Frattini, S. Puri, S. O. Mundhada, S. Touzard, M. Mirrahimi, S. M. Girvin, S. Shankar, and M. H. Devoret, Nature **584**, 205 (2020), arXiv:1907.12131.
[24] P. Aliferis and J. Preskill, Phys. Rev. A **78**, 052331 (2008).
[25] S. Puri, L. St-Jean, J. A. Gross, A. Grimm, N. E. Frattini, P. S. Iyer, A. Krishna, S. Touzard, L. Jiang, A. Blais, S. T. Flammia, and S. M. Girvin, arXiv:1905.00450

(2019).
[26] J. Guillaud and M. Mirrahimi, Phys. Rev. X **9**, 041053 (2019).
[27] C. Horsman, A. G. Fowler, S. Devitt, and R. V. Meter, New J. Phys. **14**, 123011 (2012).
[28] B. J. Brown, K. Laubscher, M. S. Kesselring, and J. R. Wootton, Phys. Rev. X **7**, 021029 (2017).
[29] D. Litinski, Quantum **3**, 128 (2019).
[30] B. J. Brown and D. J. Williamson, Phys. Rev. Research **2**, 013303 (2020).
[31] S. B. Bravyi and A. Y. Kitaev, Quantum codes on a lattice with boundary (1998), arXiv:quant-ph/9811052.
[32] A. Kay, Phys. Rev. Lett. **107**, 270502 (2011).
[33] A. R. Calderbank and P. W. Shor, Phys. Rev. A **54**, 1098 (1996).
[34] A. Steane, Proc. R. Soc. A **452**, 2551 (1996).
[35] S. Bravyi, M. Suchara, and A. Vargo, Phys. Rev. A **90**, 032326 (2014).
[36] A. M. Stephens, Phys. Rev. A **89**, 022321 (2014), arXiv:1311.5003.
[37] A. G. Fowler, D. S. Wang, and L. C. L. Hollenberg, Quant. Info. Comput. **11**, 0008 (2011).
[38] C. Wang, J. Harrington, and J. Preskill, Ann. Phys. **303**, 31 (2003).
[39] M. E. Beverland, B. J. Brown, M. J. Kastoryano, and Q. Marolleau, J. Stat. Mech.:Theor. Exp. **2019**, 073404 (2019).
[40] C. H. Bennett, Journal of Computational Physics **22**, 245 (1976).
[41] S. Bravyi and A. Vargo, Phys. Rev. A **88**, 062308 (2013), arXiv:1308.6270.
[42] R. Raussendorf, J. Harrington, and K. Goyal, Ann. Phys. **321**, 2242 (2006).
[43] R. Raussendorf and J. Harrington, Phys. Rev. Lett. **98**, 190504 (2007).
[44] H. Bombin and M. A. Martin-Delgado, J. Phys. A: Math. Theor. **42**, 095302 (2009).
[45] S. Bravyi and A. Kitaev, Phys. Rev. A **71**, 022316 (2005).
[46] P. Webster, S. D. Bartlett, and D. Poulin, Phys. Rev. A **92**, 062309 (2015).
[47] D. Litinski and F. von Oppen, Quantum **2**, 62 (2018).
[48] A. G. Fowler, A. M. Stephens, and P. Groszkowski, Phys. Rev. A **80**, 052312 (2009).
[49] T. J. Yoder and I. H. Kim, Quantum **1**, 2 (2017).
[50] H. Bombin, Phys. Rev. Lett. **105**, 030403 (2010).
[51] M. B. Hastings and A. Geller, Quant. Inf. Comp. **15**, 0962 (2015).
[52] P. Brooks and J. Preskill, Phys. Rev. A **87**, 032310 (2013).
[53] C. Chamberland, G. Zhu, T. J. Yoder, J. B. Hertzberg, and A. W. Cross, Physical Review X **10**, 011022 (2020), arXiv:1907.09528.
[54] R. Harper, S. T. Flammia, and J. J. Wallman, Nature Physics (2020), arXiv:1907.13022.
[55] D. Gottesman, arXiv:1310.2984 (2013).
[56] O. Fawzi, A. Grospellier, and A. Leverrier, arXiv preprint arXiv:1808.03821 (2018).
[57] A. Krishna and D. Poulin, Phys. Rev. Research **2**, 023116 (2020).
[58] A. Krishna and D. Poulin, Fault-tolerant gates on hypergraph product codes (2019), arXiv:1909.07424 [quant-ph].
[59] N. P. Breuckmann and V. Londe, Single-shot decoding of linear rate ldpc quantum codes with high performance (2020), arXiv:2001.03568 [quant-ph].
[60] J. R. Wootton, scirate.com/arxiv/1708.08474 (2017).
[61] F. H. E. Watson and S. D. Barrett, New J. Phys. **16**, 093045 (2014), arXiv:1312.5213.
[62] V. Kolmogorov, Math. Prog. Comp. **1**, 43 (2009).
[63] J. Edmonds, Canad. J. Math. **17**, 449 (1965).