


## ORIGINAL RESEARCH

## Surface code design for asymmetric error channels

Utkarsh Azad<sup>1,2,3</sup> | Aleksandra Lipińska<sup>4</sup> | Shilpa Mahato<sup>5</sup> | Rijul Sachdeva<sup>6,7</sup> |  
Debasmita Bhoumik<sup>8</sup> | Ritajit Majumdar<sup>8</sup> 

<sup>1</sup>Center for Computational Natural Sciences and Bioinformatics, International Institute of Information Technology, Hyderabad, India

<sup>2</sup>Center for Quantum Science and Technology, International Institute of Information Technology, Hyderabad, India

<sup>3</sup>Xanadu, Toronto, Ontario, Canada

<sup>4</sup>Faculty of Mathematics and Computer Science, Jagiellonian University, Kraków, Poland

<sup>5</sup>Department of Physics, Indian Institute of Technology Dhanbad, Dhanbad, India

<sup>6</sup>Jülich Supercomputing Center, Forschungszentrum Jülich, Jülich, Germany

<sup>7</sup>RWTH Aachen University, Aachen, Germany

<sup>8</sup>Advanced Computing & Microelectronics Unit, Indian Statistical Institute, Kolkata, India

## Correspondence

Utkarsh Azad, I-674 Govind Puram, Ghaziabad,  
Uttar Pradesh, 201013, India.

Email: [utkarsh.azad@research.iit.ac.in](mailto:utkarsh.azad@research.iit.ac.in)

## Abstract

Surface codes are quantum error correcting codes typically defined on a 2D array of qubits. A  $[d_x, d_z]$  surface code design is being introduced, where  $d_x(d_z)$  represents the distance of the code for bit (phase) error correction, motivated by the fact that the severity of bit flip and phase flip errors in the physical quantum system is asymmetric. We present pseudo-threshold and threshold values for the proposed surface code design for asymmetric error channels in the presence of various degrees of asymmetry of Pauli  $\hat{X}$ ,  $\hat{Y}$ , and  $\hat{Z}$  errors in a depolarisation channel. We demonstrate that compared to symmetric surface codes, our asymmetric surface codes can provide almost double the pseudo-threshold rates while requiring less than half the number of physical qubits in the presence of increasing asymmetry in the error channel. Our results show that for low degree of asymmetry, it is advantageous to increase  $d_x$  along with  $d_z$ . However, as the asymmetry of the channel increases, higher pseudo-threshold is obtained with increasing  $d_z$  when  $d_x$  is kept constant at a low value. Additionally, we also show that the advantage in the pseudo-threshold rates begins to saturate for any possible degree of asymmetry in the error channel as the surface code asymmetry is continued to be increased.

## KEYWORDS

asymmetric noise model, quantum error correction, surface codes

## 1 | INTRODUCTION

Quantum computers use certain quantum mechanical phenomena such as superposition and entanglement to attain a substantial speedup over their conventional classical counterparts [1–3]. Therefore, they are speculated to help solve certain problems that are intractable for even the most powerful classical supercomputers in the areas of drug discovery [4], artificial intelligence [5], material simulations [6] etc. However,

the current generation of quantum hardware, generally referred to as noisy intermediate-scale quantum (NISQ) hardware [7], has limited computational capabilities due to a small number of qubits, restrictive hardware connectivity, and poor qubit quality [7]. A significant effort is being put forward into combating these and possibly developing feasible quantum error correction strategies for mitigating the effects of external noise on the qubits [8, 9] and making large-scale quantum computation viable [10, 11].

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2022 The Authors. *IET Quantum Communication* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology.

One possible approach is to use a collection of physical qubits to construct a logical qubit that would show more resilience against noise. Based on this approach, many quantum error correction codes (QECCs) such as the Shor code [12], Steane code [13] etc. have been proposed in the past decades. However, these codes often suffer from the nearest-neighbouring problem where overhead in the number of gates and error correction cycle is huge in executing quantum circuits on hardware with restrictive topologies due to interaction requirements amongst non-adjacent qubits [14]. Topological codes were introduced to overcome this drawback [15]. Surface code is one such topological QECC acting on a two-dimensional lattice of qubits with nearest-neighbour coupling. They have been shown to have a high tolerance for local errors, that is, errors that can be corrected using just local operators acting on qubits placed in a two-dimensional grid [16]. Due to their structure, they are scalable and at the same time have a high error threshold, of a value approaching 1% [17].

Most of these codes were studied for the symmetric noise model, where each of Pauli  $\hat{X}$  (bit flip),  $\hat{Z}$  (phase flip), and  $\hat{Y} = i\hat{Z}\hat{X}$  errors occur with equal probability. However, with the physical implementations of quantum computers now being realised, it has been observed that many physical quantum channels are biased, that is, the probability of one type of error is generally much higher than other types of errors [18, 19]. Owing to this fact, the quantum error correction schemes must exploit this bias [12]. Among the most recently studied quantum codes for this purpose are the surface codes [20, 21], toric codes [22], skew cyclic codes [23], bosonic codes [24] etc.

## 1.1 | Major Contributions

We propose a generalised and scalable  $[d_x, d_z]$  surface code for the asymmetric depolarisation noise model. Here  $d_x$  and  $d_z$  represent the distance of the code for correcting Pauli  $\hat{X}$  (bit flip) and  $\hat{Z}$  (phase flip) errors, respectively. We demonstrate that it is straightforward to design the code from the existing surface code for the symmetric noise model, thus making it easy to implement in near-term quantum devices. Using the MWPM decoder, we show that this code outperforms the existing surface code model in the presence of asymmetry in the noisy channel. Additionally, using numerical results, we demonstrate the relationship between asymmetry in the surface code and asymmetry in the error channel in relation to the pseudo-threshold and threshold values. We show that our proposed asymmetric surface codes can provide almost double the pseudo-threshold rates while requiring less than half the number of physical qubits compared to symmetric surface codes. For example, the asymmetric surface code [3, 5] (Figure 3) achieves almost twice the pseudo-threshold rates than both [5, 5] and [7, 7] in a channel where the ratio of bit flip errors to phase flip errors is at most 1/10 and requires 59.18% and 70.10% lesser physical qubits than them, respectively. Furthermore, since the design of surface codes has rotational symmetry, a rotation by  $\pi/2$  will be sufficient if we ever have a channel that is biased towards the bit flip errors. Moreover, it is straightforward to

modify this code using more qubits to attain even better performance in the presence of a higher degree of asymmetry. There are other noise models studied in the literature, such as the amplitude damping, phase damping, and erasure models, which we do not consider in this study.

## 1.2 | Structure

Section 2 briefly reviews the theory of surface code, the stabiliser formalism, and the working principle of minimum weight perfect matching (MWPM) based decoder. Section 3 consists of a short introduction to the asymmetric noise model and a summary of existing quantum error correction codes (QECCs) for it. Section 4 describes the structure of our proposed surface code, along with the logical errors that can occur in it and its scalability for different levels of asymmetry. Section 5 presents the decoding results, the performance of our proposed structure, and the comparison of the symmetric and the asymmetric surface codes. Finally, Section 6 summarises and concludes the discussion and the achieved results.

## 2 | SURFACE CODES

Surface codes lie at the intersection of topological codes and stabiliser codes and are implemented on a two-dimensional lattice of physical qubits [25]. They make use of the stabiliser formalism, where the stabiliser operators perform error detection and correction. These operators are abelian sub-groups of Pauli groups, and eigenvalues of their generators encode information regarding any possible occurrence of an error. Overall, the surface codes consist of two types of physical qubits: (i) data qubits, on which the actual quantum computation is performed and (ii) ancilla qubits, on which syndrome measurement occurs for detecting errors. The latter consists of  $X$  syndrome and  $Z$  syndrome qubits, each of which interacts with the neighbouring data qubits through parity checks, that is, the value of the measurement, which is determined by the state of the connected data qubits. In Figure 1, we show one of the smallest surface codes consisting of nine data qubits and eight ancilla qubits. Here, the stabilisers are represented by the square and semi-circular tiles, where ancilla qubits lie in the plaquette (or face of the lattice) and data qubits are on the vertices [26], and their parity check operations can be represented as:

$$\begin{aligned} \mathcal{X}_0 &= X_0X_1 & \mathcal{X}_1 &= X_1X_2X_4X_5 \\ \mathcal{X}_2 &= X_3X_4X_6X_7 & \mathcal{X}_3 &= X_7X_8 \end{aligned} \quad (1)$$

$$\begin{aligned} \mathcal{Z}_0 &= Z_3Z_6 & \mathcal{Z}_1 &= Z_0Z_1Z_3Z_4 \\ \mathcal{Z}_2 &= Z_4Z_5Z_7Z_8 & \mathcal{Z}_3 &= Z_2Z_5 \end{aligned} \quad (2)$$

In each surface code cycle,  $X$  and  $Z$  syndrome qubits are initialised in their ground state and entangled with the neighbouring data qubits using CNOT gates. At the end of the surface code cycle, an error syndrome measurement (ESM) is performed on all the syndrome qubits. These measurements

measure the  $\hat{X}$  or  $\hat{Z}$  stabilisers, that is, their outcome can predict the presence of errors without perturbing the system. If the parity checks for the current surface code cycle are the same as the previous one, we conclude that the state of the data qubits involved in the parity checks has not changed due to any erroneous operation. However, if the state of an odd number of data qubits involved in a parity check is changed, the parity check returns a different value ( $0 \leftrightarrow 1$ ), triggering a *detection* event. Hence, the state of the system  $|\psi\rangle$  that it is prepared into at the end of the first surface code cycle is maintained as long as errors do not occur, and this state is known as the *quiescent state*. The altered values are passed to a decoder, which combines them to identify the possible location of the errors, that is, the indices of the data qubits.

In general, there is no decoding algorithm that provably works best for all scenarios. But, the most widely used one is the MWPM [27], in which a weighted graph is generated where the syndromes serve as nodes, and dummy nodes as boundary nodes [28]. Each node can connect to either of the four boundary nodes by hopping over one or more physical qubit (s), forming an edge. A weight is associated with this edge,

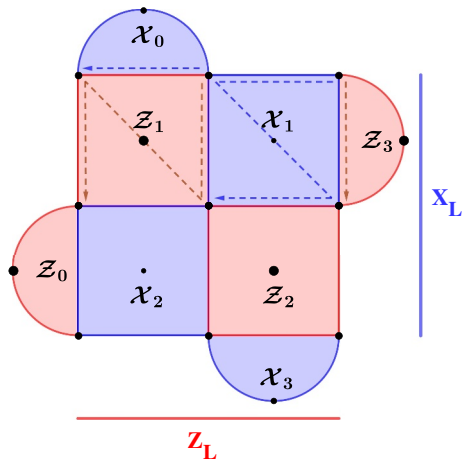


FIGURE 1 A distance 3 surface code

depending upon the number of physical qubits hopped. The edges with the minimum weight are considered, and the physical qubits hopped over are marked as erroneous and are corrected accordingly (Figure 2).

Other than this, look-up table [29] and machine learning (ML) based decoders have been proposed for surface codes as well [26, 30, 31]. Even though ML decoders have been observed to outperform MWPM decoders for the symmetric noise model, it is not known whether the same ML model with the same parameters will provide the best performance for an asymmetric model with any degree of asymmetry (since the structure becomes more and more rectangular). This would be an interesting topic for future work. Since the MWPM decoder carries over naturally from a square lattice to a rectangular one, we stick with the MWPM decoder for this study.

As illustrated in Section 4.1, all of the physical qubits present in the surface can be encoded as one or more logical qubits that are expected to be more resilient against errors. These logical qubits can be manipulated by the logical operators  $\hat{X}_L$  and  $\hat{Z}_L$ , which are defined as a chain of Pauli  $\hat{X}$  and Pauli  $\hat{Z}$  operators spanning between two vertical and horizontal boundaries, respectively. The length of these operators is  $\geq d$ , where  $d$  is defined as the distance of the surface code [26]. The number of errors ( $t$ ) that a surface code can successfully correct is also determined by its distance  $d$  as per the following equation:

$$t = \left\lfloor \frac{d-1}{2} \right\rfloor \quad (3)$$

Thus, a  $d = 3$  surface code (Figure 1) will be successfully able to correct up to a single logical  $\hat{X}$  and logical  $\hat{Z}$  error.

At the moment, our simulation considers errors only on data qubits. The study of the performance where both data and measure qubits can be erroneous is reserved as a future extension of this research. The decoding performance is usually quantified by two metrics: threshold and pseudo-threshold. The *threshold* value helps to characterise the decoding performance of a surface code design and is defined as the point

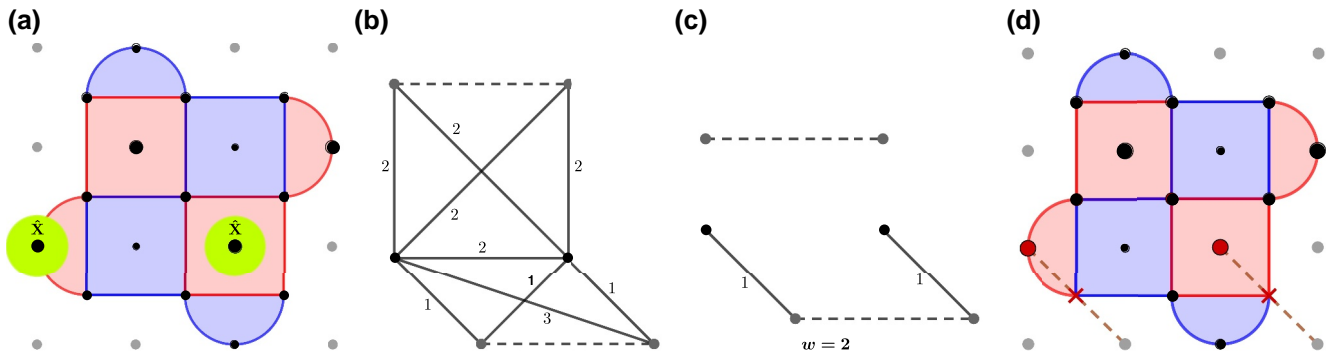


FIGURE 2 Error correction via the minimum weight matching (MWPM) algorithm. (a) Error syndromes are highlighted in green and labelled with the type of error that puts them in the  $-1$  state. (b) Error syndromes (highlighted black nodes) and dummy boundary nodes (grey) are mapped to a weighted graph. The dotted lines here represent the boundaries. (c) The goal is to find a subgraph such that each vertex (error syndrome) with either another vertex or a boundary of the total weight of matched edges is minimal. Here we represent one such possible subgraph with weight  $w = 2$ . (d) Using the subgraph, data qubits on which the error occurred are identified (marked with red crosses). They are corrected, and all error syndromes (displayed in red) are restored to the  $+1$  state again

of intersection of logical error rate curves for different code distances. Beyond this, increasing the distance of the code leads to a higher probability of logical error. The *pseudo-threshold* value determines the highest probability of physical error below which error correction leads to lower logical error probability. It is defined as the point of intersection of logical error rate curves for a given code distance with the curve for which the physical error rate is equal to the logical error rate.

### 3 | THE ASYMMETRIC NOISE MODEL

Quantum error-correcting codes serve to correct errors resulting from noise inherent in modern quantum computers. Studying the nature of the physical noise is crucial for creating QECCs with low levels of redundancy. The noise is typically asymmetric in modern physical quantum devices, and phase errors are much more probable than bit flips. It's possible to characterise the level of noise by two parameters, relaxation time and dephasing time. While dephasing only results in phase flip errors, relaxation results in both phase flip errors and bit flips [18]. This leads to a large asymmetry, where phase flips occur much more frequently than bit flips.

There exist several QECCs that exploit this asymmetry of errors. One of the most popular approaches is to concatenate two QECC codes. The simplest codes of this nature are presented in Ref. [32], where the repetition code is concatenated with the CSS code, and in Ref. [33], where a concatenation of the repetition code and error avoiding code is presented. In Ref. [34], a symmetric code is combined with an asymmetric one, and fault-tolerant circuits are needed to switch between the symmetric and asymmetric encodings. Both Refs. [18, 35] analyse a CSS construction, which uses a classical LDPC code for the more common phase errors and a classical Bose–Chaudhuri–Hocquenghem (BCH) code for bit flips. In Ref. [36], the authors use a different approach, where a higher frequency is used for syndrome measurements of  $X$ -only generators as compared to  $Z$ -only generators. Some approaches have also focussed on creating the shortest possible QECC for the asymmetric noise model. For example, in Ref. [37], the CSS-based codes are constructed using a specific syndrome assignment, and in Ref. [38], a random search of codes of up to a length of nine is performed in the hope of finding ones with desirable properties.

Closest to our work are the approaches that utilise the surface code to tackle the asymmetric noise model. In Ref. [39], a variant of the surface code is introduced, which is concatenated with a two-qubit phase detection code. A decoding scheme based on the toric code was used in Ref. [21] and tested using the MWPM decoder [27], which we also perform on our design. In Refs. [12, 40] authors demonstrate the advantages of using a rectangular form of the traditional surface code, albeit with a different decoding scheme that exploits the symmetries of its syndrome. Finally, in Refs. [41, 42], the performance of XZZX surface codes has been studied for asymmetric noise and the circuit noise model with Kerr-cat qubits, respectively.

This work aims to design a surface code for an asymmetric noise model, increasing the error-correction thresholds while still being easy to implement, and test it using an MWPM decoder.

## 4 | PROPOSED SURFACE CODE DESIGN

In the past, surface codes have generally been designed to protect the qubits against equiprobable Pauli errors, that is, the bit flip  $X$ , phase flip  $Z$  and combined bit-phase flip  $Y$  errors. Inherent to their construction, their error correction capabilities depend upon their distance  $d$ , that is, a surface code of distance  $d$  can correct up to  $t = \lfloor (d-1)/2 \rfloor$  Pauli  $\hat{X}$  and  $\hat{Z}$  errors. Usually, it is assumed that the surface codes are symmetric, that is,  $d_x = d_z = d$ . However, as explained in Section 3, Pauli errors do not occur symmetrically in all quantum channels. Instead, the phase flips errors  $Z$  happen with a different severity than the bit flip errors  $X$  for real quantum systems, majorly due to the complexity of the noise processes. Therefore, it feels natural to include this asymmetry in the design of surface codes as well. We propose to do this by introducing asymmetry in the distance of the surface codes, that is, we represent an asymmetric surface code by  $[d_x, d_z]$ . This way, such a surface code can correct up to  $t_x = \lfloor (d_x-1)/2 \rfloor$  bit flip errors and  $t_z = \lfloor (d_z-1)/2 \rfloor$  phase flip errors, and the symmetric surface codes become a special case of asymmetric surface codes with  $d_x = d_z$ .

Now depending upon the asymmetry in the channel, one can choose whether  $d_x > d_z$  should be true or  $d_x < d_z$ . Since we assume that bit flip errors are less prevalent than phase flip errors, we choose  $d_x < d_z$ . A symmetric surface code of distance  $d$  is a  $d \times d$  square lattice. Since logical  $Z$  (discussed in detail in the next subsection) is, in general, defined as a horizontal operator, our asymmetric surface code will be a  $d_z \times d_x$  lattice. An example design of the asymmetric surface code [3, 5] is illustrated in Figure 3. The position of data and ancilla qubits remains the same as in the symmetric surface code, making this straightforward to design. Our example [3, 5] asymmetric surface code has 8  $X$  stabilisers and 6  $Z$  stabilisers, which can correct up to one bit flip and two phase flip errors, respectively.

### 4.1 | Logical operators

For any given surface code  $[d_x, d_z]$ , we have  $d_x d_z$  data qubits and  $d_z d_x - 1$  measurement qubits, which give  $2^{d_x d_z}$  degrees of freedom and  $2^{(d_x d_z - 1)}$  degrees of constraints. The two unconstrained degrees of freedom allow us to consider the entire lattice as one logical qubit that can be manipulated by logical operators  $\hat{X}_L$  and  $\hat{Z}_L$ . We build these logical operators by looking at multi-qubit operator products that commute with the stabilisers and connect the opposite boundaries. Similar to the case of symmetric surface codes, the logical operators  $\hat{X}_L$  and  $\hat{Z}_L$  are defined by  $\hat{X}$  and  $\hat{Z}$  operations on the data qubits in



each column and row, respectively, where  $\|\hat{X}\| = d_x$  and  $\|\hat{Z}\| = d_z$ . These are represented in the Figure 3.

$$\hat{X}_L \in \left\{ \hat{X}_0 \hat{X}_5 \hat{X}_{10}, \hat{X}_1 \hat{X}_6 \hat{X}_{11}, \hat{X}_2 \hat{X}_7 \hat{X}_{12}, \right. \\ \left. \hat{X}_3 \hat{X}_8 \hat{X}_{13}, \hat{X}_4 \hat{X}_9 \hat{X}_{14} \right\} \quad (4)$$

$$\hat{Z}_L \in \left\{ \hat{Z}_0 \hat{Z}_1 \hat{Z}_2 \hat{Z}_3 \hat{Z}_4, \hat{Z}_5 \hat{Z}_6 \hat{Z}_7 \hat{Z}_8 \hat{Z}_9, \right. \\ \left. \hat{Z}_{10} \hat{Z}_{11} \hat{Z}_{12} \hat{Z}_{13} \hat{Z}_{14} \right\} \quad (5)$$

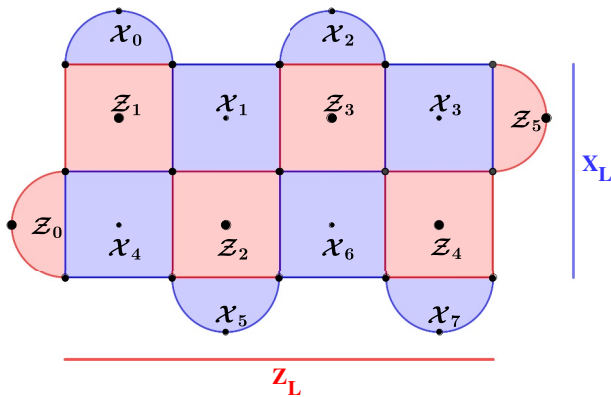
In addition to these, one can choose some other multi-qubit operator that satisfies the above conditions to build other logical operators  $\hat{X}'_L$  and  $\hat{Z}'_L$ . However, these operators will be linearly dependent on the previously defined logical operators  $\hat{X}_L$  and  $\hat{Z}_L$ . For example, consider the following case

$$\begin{aligned} \hat{X}'_L &= \hat{X}_0 \hat{X}_6 \hat{X}_{12} \\ &= \hat{X}_1 \hat{X}_2 \hat{X}_5 \hat{X}_7 \hat{X}_{10} \hat{X}_{11} (\hat{X}_0 \hat{X}_5 \hat{X}_{10}) (\hat{X}_1 \hat{X}_6 \hat{X}_{11}) \\ &\quad \times (\hat{X}_2 \hat{X}_7 \hat{X}_{12}) \\ &= (\hat{X}_1 \hat{X}_2 \hat{X}_7 \hat{X}_6) (\hat{X}_6 \hat{X}_5 \hat{X}_{10} \hat{X}_{11}) \hat{X}_L^0 \hat{X}_L^1 \hat{X}_L^2 \end{aligned} \quad (6)$$

In this case,  $\hat{X}'_L$  turns out to be a few  $\hat{X}_L^i$  operators multiplied by a couple of operator products representing stabilisers  $X_1$  and  $X_4$ , which are stabilised to a  $\pm 1$  eigenvalue by the surface code. Using this fact, we can show that the action of  $\hat{X}'_L$  on the quiescent state  $|\psi\rangle$  will be

$$\hat{X}'_L |\psi\rangle = \hat{X}_{1,2,7,6} \hat{X}_{5,6,10,11} \hat{X}_L^0 \hat{X}_L^1 \hat{X}_L^2 |\psi\rangle = \pm |\psi\rangle \quad (7)$$

This will hold true for any  $\hat{X}'_L$  (or  $\hat{Z}'_L$ ) that can be written as a product of stabilisers times  $\hat{X}_L$  (or  $\hat{Z}_L$ ) operators. Therefore, instead of looking at the complete, exhaustive list of



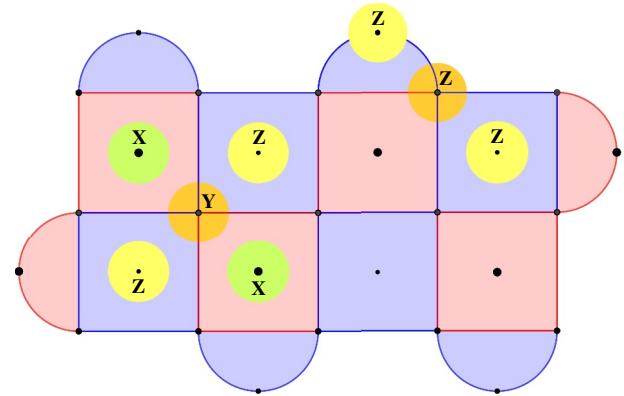
**FIGURE 3** Asymmetric surface code having distance [3, 5]. Z and X measure qubits are shown in red (indexed  $Z_0 - Z_5$ ) and blue (indexed  $X_0 - X_7$ ), respectively

possible logical operators, we can just focus on the logical operators  $\hat{X}_L$  and  $\hat{Z}_L$ .

## 4.2 | Logical errors

Now, having defined logical operators, we can look at logical errors in the asymmetric surface code. Usually, a Pauli error on a single data qubit is more likely, but sometimes Pauli errors can occur on two or more data qubits together, creating error chains. As already explained in Section 2, these are physical errors as they occur on the data (physical) qubits. These can be identified by using the MWPM algorithm [27] and then corrected manually with the help of some control software. For example, in Figure 4, we show an error chain  $\hat{Y}_6 \hat{Z}_3$  acting on the data qubits indexed 6 and 3. The error  $\hat{Y}_6$  would trigger the stabilisers  $Z_1, Z_2, X_1$  and  $X_4$ , whereas the  $\hat{Z}_3$  would trigger the stabilisers  $X_2$  and  $X_3$ . The syndrome for X stabilisers would be  $[0, 1, 1, 1, 0, 0, 0, 0]$  and for Z stabilisers would be  $[0, 1, 1, 1, 0, 0]$ . These are used by MWPM for decoding and identifying which data qubits possibly suffered the errors.

However, there are times when MWPM misidentifies the errors, especially when the error chain formed is not sparse. In such cases, there is a possibility of errors persisting even after the correction. These errors are more dangerous when they are logical errors, that is, they change the state of the logical qubit made by the surface code. It is only those errors, which anti-commute with at least one of the  $\hat{X}$  (or  $\hat{Z}$ ) logical operators forming the Z (or X) logical errors. Therefore, these errors remain undetected and affect the computation henceforth since the logical state has changed. For example, if in the previous example, the error chain would have been  $\hat{Z}_2 \hat{Z}_4 \hat{X}_6 \hat{Z}_{10}$



**FIGURE 4** The working of the asymmetric surface code [3, 5]. Z and X measure qubits shown in red and blue, respectively, are measured in each surface code cycle.  $\hat{X}$  ( $\hat{Z}$ ) errors on an odd number of data qubits involved in the Z (X) syndrome measurements will result in  $-1$  eigenvalues, triggering a detection event shown in green (yellow). Here, the Pauli  $\hat{Y} = i\hat{Z}\hat{X}$  error on the data qubit indexed 6 is captured by the syndromes for both Z and X stabilisers. Whereas the Pauli  $\hat{Z}$  error on the data qubit indexed 3 affects the syndromes for only X stabilisers. The X (Z) syndrome measurement outcomes are represented by a binary vector where all the syndromes involved in the detection events have a value of 1. These binary vectors are sent to the MWPM decoder, which combines them to identify the possible indices of data qubits that suffered the errors

instead of  $\hat{Y}_6\hat{Z}_3$ , the syndromes generated (and hence the decoding result) would still have been the same, leading to an overall error of  $\hat{Z}_2\hat{Z}_4\hat{Z}_3\hat{Y}_6\hat{Z}_{10}$ . This is a logical Z error because it anti-commutes with more than one  $\hat{X}_L$  operator such as  $\hat{X}_0\hat{X}_5\hat{X}_{10}$  or  $\hat{X}_2\hat{X}_7\hat{X}_{12}$ . Therefore, comparing how many such logical errors occur from a given number of decoding trials provides us with estimates of the asymmetric surface code's tolerance against errors, which we present in the next section.

## 5 | RESULTS

We analyse the performance of our proposed surface code design for correcting errors with different levels of asymmetry in (i) code distance pair  $[d_x, d_z]$ , (ii) Pauli errors in the depolarising noise channel, and (iii) a combination of both of them.

As mentioned before, we use pseudo-threshold values to compare the decoding performance of two surface codes with different code distance pairs  $[d_x, d_z^i]$  and  $[d_x, d_z^j]$ . This value is estimated for a given surface code by performing a simulation to calculate logical error rates  $P_L$  for a range of physical error rates  $p$ , where the former is defined for any given value of the latter as the ratio of total logical errors accumulated in all error correction cycles to the number of net error correction cycles performed. Each of the simulations was averaged over 50,000 repetitions, with the range of physical error  $\kappa$  varying between  $\kappa \in [10^{-4}, 5.5 \times 10^{-2}]$ , which corresponded to the physical error rate of  $p = 1 - (1 - \kappa)^8$  for the complete execution of a surface code cycle. Here, the factor of 8 arises from the fact that each cycle consists of eight steps as shown in Ref. [25].

Additionally, for decoding, we have used the PyMatching library [43], which implements the MWPM algorithm [27] to calculate the data qubits that need correction from a provided error syndrome.

### 5.1 | Performance on bit flip (X) and phase flip (Z) errors

In the first set of experiments, we compared the performance of proposed asymmetric surface codes  $[d_x, d_z]$ , where  $d_x, d_z \in \{3, 5, 7\}$  and  $d_x < d_z$ , against their corresponding symmetric counterparts  $[d, d]$ , where  $d \in \{3, 5, 7\}$ , for correcting bit flip (X) and phase flip (Z) errors, independently. To do this, we tested them for error channels with only one kind of error, either Pauli  $\hat{X}$  or  $\hat{Z}$ , and not both. From the results presented in Figure 5, it is evident that there is an overall improvement in the performance of correcting phase errors as the asymmetry in the surface codes  $[d_x, d_z]$  is increased. This can also be inferred from their higher pseudo-threshold values than their symmetric variants  $[d_x, d_x]$  and  $[d_z, d_z]$ . For example, [3, 5] and [3, 7] have much better pseudo-threshold values than [3, 3], [5, 5] and [3, 3], [7, 7], respectively.

However, at the same time, we also notice a considerable decrease in the performance of correcting bit flip errors for

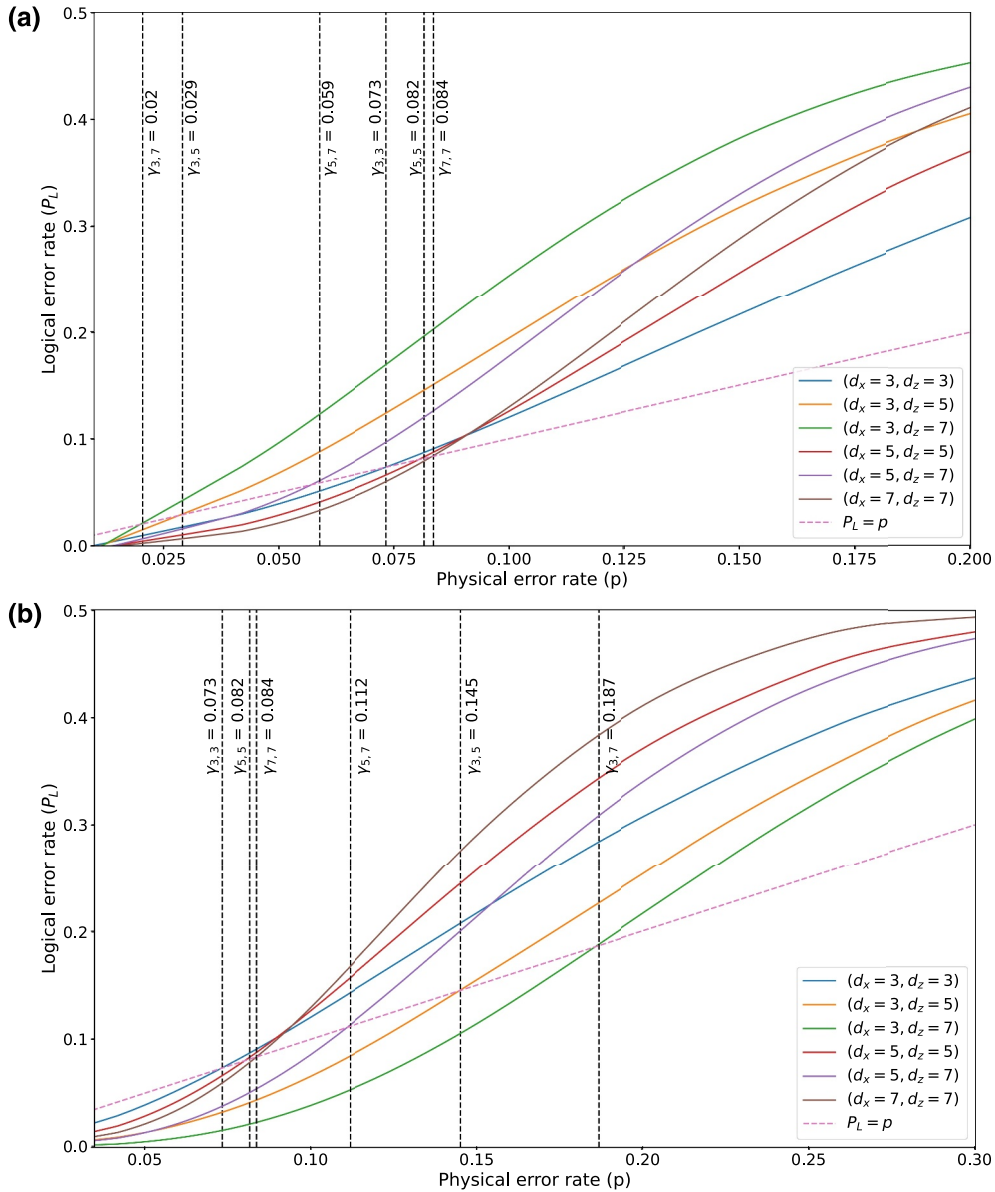
asymmetric surface codes. We attribute this decrease to an insufficient increase in the number of Z stabilisers in comparison to the rise in the number of possible combinations of  $X_L$ , due to an increase in the number of qubits ( $d_x \times d_x \rightarrow d_x \times d_z; d_z > d_x$ ). For example, in the case of [3, 5], the increase in possible combinations of  $X_L$  is from  ${}^9C_3$  to  ${}^{15}C_3$ , while Z stabilisers increment from 4 to only 6. Similar logic can also explain the intermediate performance of [5, 7] in both cases compared to the rest of the codes.

### 5.2 | Performance on symmetric and asymmetric noise models

The graphs in Figure 5 imply that the probability of a bit flip error increases with increasing asymmetry of the surface code structure. This naturally raises the question of whether, for asymmetric depolarising noise, it is suitable to increase only  $d_z$  with asymmetry, or should  $d_x$  also be increased at a regular interval. In the second experiment, we numerically establish that when the degree of asymmetry is low, one should indeed increase  $d_x$  along with  $d_z$  to avoid performance degradation. But with a higher degree of asymmetry, it is better to keep  $d_x$  fixed and increase  $d_z$  only.

We first analysed how the performance of a set of asymmetric surface codes  $[d_x, d_z]$  varies with an increase in asymmetry ( $\Delta$ ) in the depolarising error channel:  $[p_x = p/(\Delta + 2), p_y = p/(\Delta + 2), p_z = p\Delta/(\Delta + 2)]$ , where,  $p$  is the value of the physical error rate for a given physical error  $\kappa$ . As presented in Table 1, for the symmetric noise model, that is, at  $\Delta = 1$ , asymmetric models perform poorly in comparison to their symmetric counterparts. For example, the pseudo-threshold for [3, 5] is much lower both [3, 3] and [5, 5]. This behaviour of asymmetric models is a consequence of their reduced ability to correct for bit flip errors compared to symmetric models, as discussed in the previous subsection.

Furthermore, as the asymmetry increases step-wise ( $\Delta \rightarrow \Delta + 1$ ), we see an improvement in the performance of asymmetric models, as inferred from their increased pseudo-threshold values. This happens because asymmetric codes can now correct overall more number of logical errors as bit flip errors decrease in the creation of logical errors with the simultaneous increase in the contribution of phase flip errors. Additionally, even among the asymmetric models themselves, the ones with  $d_x = 3$  dominate the performance for  $\Delta \geq 4$ . Therefore, we infer from Table 1 that for  $\Delta < 4$ , the asymmetry in the channel is *weak*, and one should increase  $d_x$  regularly with  $d_z$  to obtain optimal performance. However, for  $\Delta \geq 4$ , the asymmetry of the channel becomes *strong* enough to allow an increase in  $d_z$  alone, keeping  $d_x$  constant, to improve the pseudo-threshold. Therefore, as asymmetry in the error channel increases, or in other words, when the channel becomes *sufficiently asymmetric*, one can achieve much better logical error correction with fewer physical qubits simply by increasing the asymmetry in the surface code. From our numerical results, the *crossover point* appears to be  $\Delta = 4$  for our noise model.



**FIGURE 5** Performance of asymmetric surface codes,  $\{[3, 5], [3, 7], [5, 7]\}$ , in comparison to symmetric surface codes,  $\{[3, 3], [5, 5], [7, 7]\}$ , for correcting (a) bit flip errors and (b) phase flip errors. Pseudo-thresholds for each surface code  $(d_x, d_z)$  is represented by  $\gamma_{d_x, d_z}$

Therefore, when the degree of asymmetry is greater than the *crossover point*, our proposed method leads to a better decoding performance while requiring fewer number of qubits. A traditional distance  $d$  surface code creates a single logical qubit using  $d^2$  data qubits. Whereas, our proposed modification requires  $d_x \times d_z$  data qubits to create a single logical qubit. The percentage savings, therefore, is  $(d_z - d_x)/d_z \times 100\%$ . For example, when the degree of asymmetry ( $\Delta$ ) is 10, the (3, 7) code achieves a percentage increase of  $\approx 48\%$  in the pseudo-threshold with a percentage savings of  $\approx 57\%$  in the number of qubits.

Subsequently, we also note that, for some initial values of  $\Delta$ , there is a slight improvement in the performance of symmetric surface codes  $[3, 3]$ ,  $[5, 5]$ , and  $[7, 7]$ . However, these gains soon get diminished as asymmetry is increased further in the depolarisation channel. This initial increment could be due to a slight

dip in net logical errors because the combined decrease in both  $\sigma_x$  and  $\sigma_y$  errors overpowers an increase in  $\sigma_z$  errors.

### 5.3 | Comparison of symmetric and asymmetric surface codes

In our third set of experiments, building upon the results from the previous two subsections, we compared the performance of a series of surface codes  $[d_x, d_z]$ , where  $d_x = 3$  and  $d_z \in \{3, 5, \dots, 17\}$  by analysing the variation of their pseudo-threshold values for  $\Delta \in [1, 10, 20, \dots, 100]$ . The resulting data has been plotted in Figure 6, and it shows that there is an evident increase in the performance of error correction in the presence of asymmetric error channels as the asymmetry in surface codes is increased.

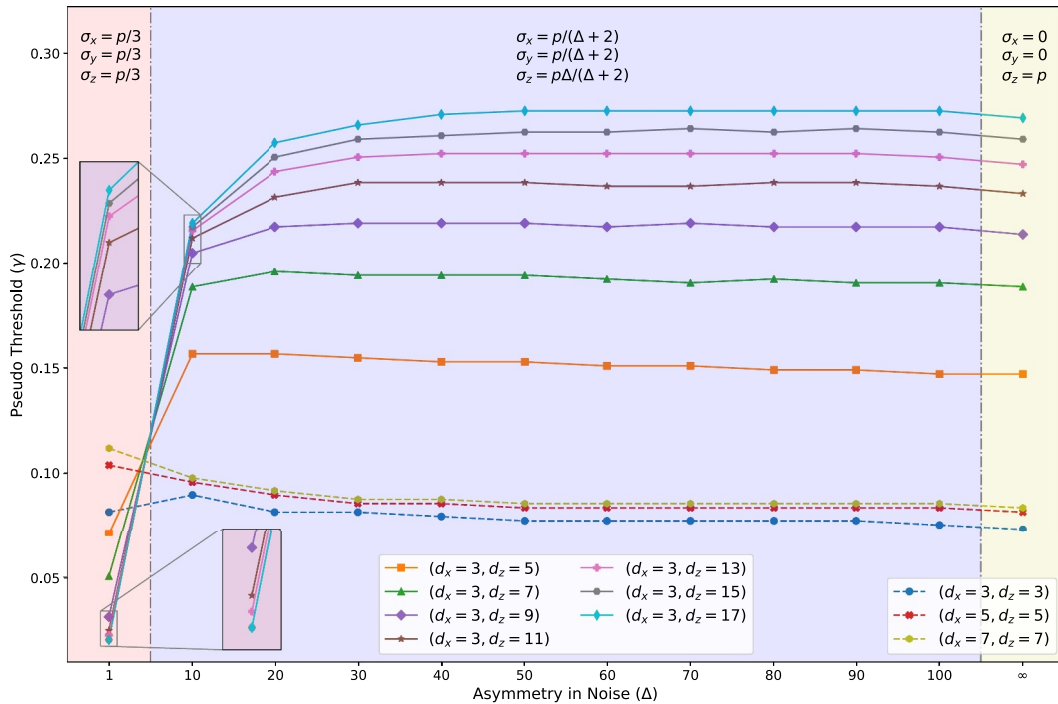
We compare these performances to that of the three symmetric surface codes  $[d_x, d_z]$ , where  $d_x = d_z \in \{3, 5, 7\}$ . We divide the whole range of asymmetry into three regions: (i) symmetric ( $\Delta = 1$ ), (ii) intermediate asymmetry ( $\Delta \in [10, 100]$ ), and (iii) strong asymmetry ( $\Delta \rightarrow \infty$ ). In the first region, we see that the symmetric surface codes perform much better than asymmetric ones for correcting logical errors, majorly due to the decrease in the capability of asymmetric surface codes in solving phase errors, as explained in Section 5.1. However, the

**TABLE 1** Variation in pseudo-threshold ( $\gamma_{d_x, d_z}$ ) values with respect to step-wise change in the asymmetry of depolarising noise

Asymmetry ( $\Delta$ )	Pseudo-thresholds					
	$\gamma_{3,3}$	$\gamma_{3,5}$	$\gamma_{3,7}$	$\gamma_{5,5}$	$\gamma_{5,7}$	$\gamma_{7,7}$
1	0.082	0.073	0.065	0.102	0.096	0.110
2	0.094	0.108	0.096	0.110	0.122	0.118
3	0.094	0.128	0.128	0.110	0.132	0.116
4	0.096	0.137	0.151	0.108	0.136	0.112
5	0.092	0.145	0.167	0.104	0.138	0.108
6	0.094	0.151	0.176	0.102	0.136	0.104
7	0.092	0.153	0.184	0.100	0.134	0.102
8	0.09	0.155	0.187	0.098	0.132	0.100
9	0.089	0.157	0.190	0.096	0.130	0.098
10	0.088	0.155	0.192	0.095	0.128	0.098

asymmetric surface codes become more efficient at correcting logical errors as sufficient asymmetry is introduced in the noise model in the second region. Similar to the trend observed in the previous section for low asymmetry ( $\Delta \in [2, 9]$ ), we see a steady increase in the values of pseudo-thresholds as the asymmetry is increased in the surface code  $[d_x, d_z]$ , that is, the value of  $d_z$  is incremented while keeping  $d_x$  constant. In contrast to symmetric surface codes, where the pseudo-threshold values appear to be saturated right from the beginning, for asymmetric surface codes, we first observe a steady decrease in the improvement of pseudo-threshold values. Only then, it appears to be saturating for sufficiently bigger asymmetric surface codes, that is, the number of data qubits being greater than 50.

Overall, once there is a *sufficient* increase in the asymmetry in the noise model, the performance of both symmetric and asymmetric surface codes remains almost constant. A remarkable consequence of this observation is that by using asymmetric surface codes, one can get significantly improved pseudo-threshold rates while requiring less than half the number of physical qubits than the symmetric surface codes. For example, for any  $\Delta > 10$ , we were able to almost double the pseudo-threshold rates with our  $[3, 5]$  surface code in comparison to both  $[5, 5]$  and  $[7, 7]$ , decreasing physical qubit counts by 59.18% and 70.10% respectively. Furthermore, there is an observable slight dip in the final region where only phase errors exist as  $\Delta \rightarrow \infty$ . This slight decrease in this region can be attributed to the fact that the complete absence of any  $\sigma_x$  and  $\sigma_y$  errors allows for slightly more logical  $Z$  errors to come up from the possible  $d_x d_z C_{d_z}$  combinations.



**FIGURE 6** Analysing change in pseudo-threshold values for different asymmetric and symmetric surface codes with the asymmetry in the noise. Here, we consider three regions of asymmetry demarcated by different values of  $\Delta$ —(i) symmetric:  $\Delta = 1$  (red), (ii) intermediate asymmetry:  $\Delta \in [10, 100]$  (blue), and (iii) strong asymmetry:  $\Delta \rightarrow \infty$  (yellow)



**TABLE 2** Variation in threshold ( $\gamma^*$ ) values with respect to change in the asymmetry of Pauli errors in the depolarising channel

Asymmetry ( $\Delta$ )	Thresholds ( $\gamma^*$ )
1	0.261
10	0.278
20	0.363
30	0.383
40	0.396
50	0.407
60	0.413
70	0.423
80	0.437
90	0.443
100	0.450
$\infty$	0.500

## 5.4 | Thresholds values for asymmetric surface codes

As seen in previous subsections, the relationship between  $P_L$  and  $p$  changes with (i) asymmetric distances of the surface code, represented by  $d_x$  and  $d_z$ , and (ii) asymmetry in the Pauli errors occurring in the channel. In this experiment as well, we have kept  $d_x = 3$  constant, while varying the  $d_z \in \{5, 7, \dots, 17\}$ . For all values of asymmetry  $\Delta \in \{1, 10, 20, \dots, 90, 100, \infty\}$ , we see that the values of  $P_L$  increases (or decreases) with an increase (or decrease) in  $p$  as the  $d_z$  is increased until a certain value of  $p = \gamma^*$  is reached, after which the trend reverses itself. This value  $\gamma^*$  is known as the threshold rate, and we list them down for asymmetric surface code design in Table 2. Following a similar trend as pseudo-threshold rates, the threshold rates follow a monotonous increasing trend as the asymmetry in the Pauli errors is increased.

## 6 | CONCLUSIONS

This paper investigates a novel design for asymmetric surface codes for the quantum asymmetric Pauli channels. We have proposed the design of a  $[d_x, d_z]$  surface code, where  $d_z > d_x$  in general, for better error correction when the underlying channel is more biased towards Pauli Z errors. In the first look, our rectangular structure ( $d_x \cdot d_z$ ) seems to use more qubits than the traditional square ( $d_x \cdot d_x$ ) surface code. However, we show that as the asymmetry of the channel increases, our proposed  $[d_x, d_z]$  code, where  $d_x < d_z$ , provides a higher pseudo-threshold than the traditional surface code, where  $d_x = d_z$ , using a much smaller number of qubits. When the degree of asymmetry is low, it seems advantageous to somewhat increase  $d_x$  regularly with the increase in  $d_z$  to retain the optimum performance. But with a higher degree of asymmetry,

we obtain better performance for a constant  $d_x$  which is kept fixed at the minimum value of 3, thus providing percentage savings of  $(d_z - d_x)/d_z \times 100\%$  in qubits. Moreover, by varying  $d_x$  instead of  $d_z$ , these results immediately apply to the inverse case of the noise model studied here, that is, where bit flips (rather than phase flips) are prevalent. Therefore, we conclude that the proposed asymmetric surface code is more advantageous than symmetric surface codes in the presence of asymmetry in the channel's noise.

## 6.1 | Note added

During the revision of this paper, we were made aware of some recent articles [44–46] that also describe codes tailored to biased noise. These work are based on  $XY$  surface codes,  $XYZ^2$  stabiliser codes, and Clifford-deformed surface codes, respectively, unlike the  $XXZZ$  surface codes we have used in our work.

## ACKNOWLEDGEMENT

This project was started as part of the QIntern 2021 event. Authors acknowledge and thank the QWorld Association for hosting it.

## CONFLICT OF INTEREST

There is no conflict of interest.

## DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

## ORCID

Ritajit Majumdar  <https://orcid.org/0000-0003-0730-0084>

## REFERENCES

1. Harrow, A.W., Montanaro, A.: Quantum computational supremacy. *Nature*. 549(7671), 203–209 (2017)
2. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.* 26(5), 1484–1509 (1997)
3. Aaronson, S.: The limits of quantum computers. *Sci. Am.* 298(3), 62–69 (2008)
4. Zinner, M., et al.: Quantum computing's potential for drug discovery: early stage industry dynamics. *Drug Discov. Today*. 26(7), 1680–1688 (2021)
5. Acampora, G.: Quantum machine intelligence. *Quan. Mach. Intell.* 1(1–2), 1–3 (2019)
6. Ma, H., Govoni, M., Galli, G.: Quantum simulations of materials on near-term quantum computers. *npj Comput. Mat.* 6(1) (2020)
7. Preskill, J.: Quantum computing in the NISQ era and beyond. *Quantum*. 2, 79 (2018)
8. Chamberland, C., et al.: Topological and subsystem codes on low-degree graphs with flag Qubits. *Phys. Rev. X*. 10, 011022 (2020)
9. O'Brien, T.E., et al.: Error mitigation via verified phase estimation. *PRX Quan.* 2 (2021)
10. Chao, R., Reichardt, B.W.: Fault-tolerant quantum computation with few qubits. *Nature*. 4(1) (2018)
11. Litinski, D.: A game of surface codes: large-scale quantum computing with lattice surgery. *Quantum*. 3, 128 (2019)
12. Tuckett, D.K., et al.: Tailoring surface codes for highly biased noise. *Phys. Rev. X*. 9(4) (2019)

13. Steane, A.M.: Error correcting codes in quantum theory. *Phys. Rev. Lett.* 77(5), 793–797 (1996)
14. Majumdar, R., Basu, S., Sur-Kolay, S.: A method to reduce resources for quantum error correction. In: *International Conference on Reversible Computation*, pp. 151–161. Springer (2017)
15. Kitaev, A.Y.: Fault-tolerant quantum computation by anyons. *Ann. Phys.* 303(1), 2–30 (2003)
16. Wang, C., Harrington, J., Preskill, J.: Confinement-Higgs transition in a disordered gauge theory and the accuracy threshold for quantum memory. *Ann. Phys.* 303(1), 31–58 (2003)
17. Fowler, A.G., Stephens, A.M., Groszkowski, P.: High-threshold universal quantum computation on the surface code. *Phys. Rev. A* 80(5) (2009)
18. Ioffe, L., Mézard, M.: Asymmetric quantum error-correcting codes. *Phys. Rev. A* 75, 032345 (2007)
19. Calderbank, A.R., Shor, P.W.: Good quantum error-correcting codes exist. *Phys. Rev. A* 54(2), 1098–1105 (1996)
20. Chamberland, C., Campbell, E.T.: Universal quantum computing with twist-free and temporally encoded lattice surgery. *PRX Quan.* 3(1), 010331 (2022)
21. Nickerson, N.H., Brown, B.J.: Analysing correlated noise on the surface code using adaptive decoding algorithms. *Quantum* 3, 131 (2019)
22. Hansen, J.P.: Asymmetric Quantum Codes on Toric Surfaces (2017). arXiv e-prints
23. Muchtadi-Alamsyah, I., Irwansyah: Asymmetric quantum codes from skew cyclic codes over B1. *AIP Conf. Proc.* 2192(1), 040008 (2019)
24. Chamberland, C., et al.: Building a fault-tolerant quantum computer using concatenated cat codes. *PRX Quan.* 3(1) (2022)
25. Fowler, A., et al.: Surface codes: towards practical large-scale quantum computation. *Phys. Rev. A* 86 (2012)
26. Varsamopoulos, S., Bertels, K., Almudever, C.G.: Comparing neural network based decoders for the surface code. *IEEE Trans. Comput.* 69(2), 300–311 (2020)
27. Fowler, A.G.: Minimum weight perfect matching of fault-tolerant topological quantum error correction in average  $O(1)$  parallel time. *Quant. Inf. Comput.* 15, 145–158 (2015)
28. Hadfield, S.: Error Rate Thresholds for Physical Implementations of the Surface Code. PhD thesis, School of Physics, University of Melbourne (2008)
29. Varsamopoulos, S., Criger, B., Bertels, K.: Decoding small surface codes with feedforward neural networks. *Quant. Sci. Technol.* 3(1), 015004 (2017)
30. Bhoumik, D., et al.: Efficient Decoding of Surface Code Syndromes for Error Correction in Quantum Computing (2021). arXiv preprint arXiv:2110.10896
31. Sweke, R., et al.: Reinforcement learning decoders for fault-tolerant quantum computation. *Mach. Learn.: Sci. Technol.* 2(2), 025005 (2020)
32. Aliferis, P., Preskill, J.: Fault-tolerant quantum computation against biased noise. *Phys. Rev. A* 78, 052331 (2008)
33. Cafaro, C., et al.: Quantifying the performance of quantum codes. *Open Syst. Inf. Dynam.* 18(01), 1–31 (2011)
34. Stephens, A.M., et al.: Asymmetric quantum error correction via code conversion. *Phys. Rev. A* 77(6) (2008)
35. Sarvepalli, P.K., Klappenecker, A., Rötteler, M.: Asymmetric quantum codes: constructions, bounds and performance. *Proc. Math. Phys. Eng. Sci.* 465(2105), 1645–1672 (2009)
36. Evans, Z.W.E., et al.: Error Correction Optimisation in the Presence of X/Z Asymmetry (2007). arXiv e-prints
37. Chiani, M., Valentini, L.: Design of short codes for quantum channels with asymmetric Pauli errors. In: Krzhizhanovskaya, V.V., et al. (eds.) *Computational Science – ICCS 2020*, pp. 638–649. Springer International Publishing, Cham (2020)
38. Jackson, T., Grassl, M., Zeng, B.: Codeword stabilized quantum codes for asymmetric channels. In: 2016 IEEE International Symposium on Information Theory (ISIT) (2016)
39. Xu, X., et al.: High-threshold code for modular hardware with asymmetric noise. *Phys. Rev. Appl.* 12(6) (2019)
40. Tuckett, D.K., et al.: Fault-tolerant thresholds for the surface code in excess of 5% under biased noise. *Phys. Rev. Lett.* 124(13) (2020)
41. Pablo Bonilla Ataides, J., et al.: The XXXX surface code. *Nat. Commun.* 12(1) (2021)
42. Darmawan, A.S., et al.: Practical quantum error correction with the XXXX code and Kerr-Cat Qubits. *PRX Quan.* 2(3) (2021)
43. Higgott, O.: PyMatching: A Python Package for Decoding Quantum Codes with Minimum-Weight Perfect Matching (2021). arXiv e-prints
44. Higgott, O., et al.: Fragile Boundaries of Tailored Surface Codes (2022). arXiv e-prints
45. Srivastava, B., Kockum, A.F., Granath, M.: The XYZ<sup>2</sup> Hexagonal Stabilizer Code (2021). arXiv e-prints
46. Dua, A., et al.: Clifford-Deformed Surface Codes (2022). arXiv e-prints

**How to cite this article:** Azad, U., et al.: Surface code design for asymmetric error channels. *IET Quant. Comm.* 3(3), 174–183 (2022). <https://doi.org/10.1049/qtc.2.12042>