

Minimising surface code failures using a color code decoder

Asmae Benhemou¹, Kaavya Sahay², Lingling Lao¹, and Benjamin J. Brown³

¹Department of Physics and Astronomy, University College London, London WC1E 6BT, United Kingdom

²Department of Applied Physics, Yale University, New Haven, Connecticut 06511, USA

³Niels Bohr International Academy, Niels Bohr Institute, Blegdamsvej 17, 2100 Copenhagen, Denmark

The development of practical, high-performance decoding algorithms reduces the resource cost of fault-tolerant quantum computing. Here we propose a decoder for the surface code that finds low-weight correction operators for errors produced by the depolarising noise model. The decoder is obtained by mapping the syndrome of the surface code onto that of the color code, thereby allowing us to adopt more sophisticated graph-based decoding protocols used for the color code. Analytical arguments and exhaustive testing show that the resulting decoder can find a least-weight correction for all weight $d/2$ depolarising errors for even code distance d . This improves the logical error rate by an exponential factor $O(2^{d/2})$ compared with decoders that treat bit-flip and dephasing errors separately. We demonstrate this improvement with analytical arguments and supporting numerical simulations at low error rates. Of independent interest, we also demonstrate an exponential improvement in logical error rate for our decoder used to correct independent and identically distributed bit-flip errors affecting the color code compared with more conventional color code decoding algorithms.

1 Introduction

We envisage that a large-scale quantum computer will operate by performing fault-tolerant logic gates on qubits encoded using quantum error-correcting codes (QEC) [1, 2, 3, 4, 5, 6]. As quantum information is processed, a classical decoding algorithm [7, 8, 9, 10, 11, 12, 13, 14] will be used to interpret syndrome information that is collected by making parity measurements over

the physical qubits of a code, to determine the errors it experiences. It is important to design high-performance and practical decoding algorithms to minimise the number of failure mechanisms that lead to a logical qubit error in real quantum systems. Such algorithms will reduce the high resource cost of encoding logical qubits, as better decoding algorithms will allow us to achieve a target rate of logical failures using fewer physical qubits.

Topological codes, such as the surface code [15, 16, 7, 17] and color code [18, 19, 20, 21], are among the most promising quantum error-correcting codes to be realised with quantum technology that is now under development [22, 23, 24, 25, 26, 27]. This is due to their layout which can be realised with a planar array of qubits with only nearest-neighbour interactions, and their demonstrated high threshold error rates, such that they can function under significant laboratory noise. A well-established class of decoders for the surface code [7, 28, 29, 17] is based on the minimum-weight perfect-matching algorithm (MWPM) [30, 7, 31]. We colloquially refer to these decoders as matching decoders. Matching decoders give rise to high thresholds for topological codes, and have been demonstrated to be highly versatile in that they can be adapted to correct for different noise models [32, 33, 34, 35, 36, 37, 38, 39, 40], as well as different codes [41, 42, 43], including the color code [44, 45, 46, 47, 48, 49] and surface code variants [50, 51, 52, 53, 54]. See the perspective article, Ref. [55] for a discussion.

A matching decoder exploits the structure of the code to identify errors which likely caused the error syndrome. Specifically, we take subsets of the syndrome data and represent them on a lattice, such that this syndrome data respects certain symmetries of the code [15, 41, 55]. These symmetries are characterised by the fact

that errors produce syndrome violations, commonly known as defects, in pairs over the lattice. Given such a lattice, the MWPM algorithm can be used to pair spatially local defects. In turn, this pairing can be interpreted to identify an error that is likely to have given rise to the syndrome, such that a correction that recovers the encoded state can be determined.

The performance of a matching decoder is determined by the choice of syndrome lattice. The surface code has consistently demonstrated high thresholds by concentrating on syndrome lattices which correct for bit-flip errors and dephasing errors separately. However, without additional decoding steps [56, 45, 57], such decoders cannot identify correlations between these two error types that occur in common error models such as depolarising noise. See for example Refs. [8, 56] where candid discussions are presented on the limitations of matching decoder variations correcting for depolarising noise. In this work, we introduce a syndrome lattice for the surface code that accounts for correlations between bit-flip and dephasing errors, which we obtain by exploiting a correspondence between the surface code and the color code, commonly known as an unfolding map [58, 59, 60, 61]. Through this mapping, we can adopt color code decoding algorithms to correct noise on the surface code. Indeed, a syndrome lattice for our decoder naturally ensues by adapting the decoding methods introduced in Ref. [48].

We find that our unified decoder determines a least-weight correction for all weight $d/2$ depolarising noise errors for surface codes with even distance d . As such, in the limit of low error rate, we obtain optimal logical failure rates. In contrast, conventional matching decoders that do not account for correlated errors have a logical failure rate $O(2^{d/2})$ a factor higher than the decoder we present. We adopt a number of methods to analyse the performance of our decoder at low error rates. We first evaluate the number of weight $d/2$ errors that should lead to a logical failure for both our matching decoder, as well as a conventional matching decoder. We verify our expressions with exhaustive searching of weight $d/2$ errors. In addition, we measure the logical failure rate of our decoder using the splitting method [62, 63]. We find the numerical results we obtain to be consistent with our analytic results in the limit of

vanishing physical error rate, thereby verifying the performance of our decoder in comparison to conventional decoders. We also show that the threshold of our decoder is comparable to more conventional decoding methods. Finally, we investigate the performance of our scheme to decode the color code undergoing bit-flip noise. We find our analysis sheds light on the role of entropy in quantum error correction with the matching decoders that are central to our study [64, 65, 66].

The remainder of this manuscript is structured as follows. In Sec. 2, we briefly review the quantum error-correcting codes of relevance in this study, namely the surface code and color code. In Sec. 3, we review the symmetries of the color code, and how they are used to obtain matching decoders. We also explain why the unified decoder is capable of decoding high-weight errors. In Sec. 4, we describe the mapping from surface to color code, and introduce a noise mapping from depolarising errors on the surface code to a bit-flip noise on the color code. We present the results supporting the out-performance of the unified decoder over its restricted counterpart under depolarising noise on the surface code in Sec. 5, and we present results using the same decoder to correct bit-flip noise on the color code in Sec. 6. We finally offer concluding remarks in Sec. 7.

2 Preliminaries

In this section we introduce the *surface code* and the *color code*, both of which are central to this work. We begin by stating the stabilizer formalism we use to describe these codes.

2.1 The stabilizer formalism

Quantum-error correcting codes are designed to protect logical information that is encoded in a subspace of a larger physical Hilbert space. This subspace is called the code space. A large class of QEC codes are readily described using the stabilizer formalism [67], whereby a QEC code is defined by an Abelian subgroup of the Pauli group \mathcal{P} , known as the stabilizer group $\mathcal{S} \subset \mathcal{P}$. Up to phases, the Pauli group on n qubits is generated by the operators n -qubit Pauli matrices X_q and Z_q which respectively denote the standard Pauli-X and Pauli-Z operators acting on qubit q . The code space of a stabilizer code is

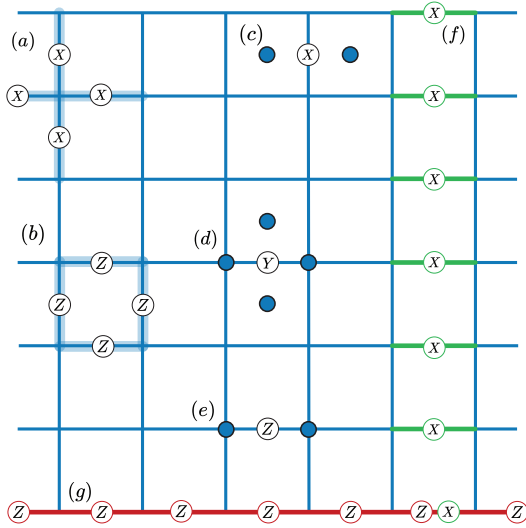


Figure 1: The distance-7 unrotated surface code with qubits on the edges. (a) An X -type (star) parity check operator. (b) A Z -type (face) parity check operator. (c,d,e) Single-qubit X , Y and Z errors and their syndromes. (f) A vertical \bar{X} logical operator. (g) A horizontal \bar{Z} logical operator.

the $+1$ eigenspace of all of its elements, namely $s|\psi\rangle = |\psi\rangle \forall s \in \mathcal{S}$, where the code space is spanned by eigenstates $|\psi\rangle$.

A stabilizer code is capable of detecting errors locally, without revealing information of the global encoded state. We consider Pauli errors E that act on the code qubits. If an error E anti-commutes with a subset of the stabilizer generators, when these are measured we obtain a set of syndrome defects to be used for error correction. We refer to a syndrome defect where a stabilizer measurement returns a negative outcome, i.e. $S_f E = -E S_f$.

A correction C can be found and applied to return the code to the ground subspace such that $CE \in \mathcal{S}$, and recover the encoded logical state $|\psi\rangle$. We obtain C using a decoding algorithm that is designed to determine a highly probable error giving rise to a measured syndrome, given a known error model where we assume errors are introduced at a low rate. In general, an $[[n, k, d]]$ quantum code is characterised by the number of physical qubits n , the number of logical qubits k that they encode, and a code distance d given by the minimum weight of an error configuration triggering a logical error.

2.2 The surface code

We define the surface code on a two-dimensional square lattice with boundaries, see Fig. 1 [15, 7]. A single physical qubit is supported on each edge e of the lattice. The stabilizer group of the surface code is generated by vertex operators and face operators, denoted A_v and B_f , respectively. A vertex operator is defined at each site v of the lattice, acting on the qubits located on the incident edges E_v on the vertex, such that $A_v = \prod_{e \in E_v} X_e$. Similarly, we define an operator $B_f = \prod_{e \in E_f} Z_e$ on each face f of the lattice, where the set E_f of edges is adjacent to the face f . On this planar variant of the surface code, the product of Z (X) stabilizer operators creates a smooth (rough) boundary at the top and bottom (left and right) of the lattice. A string of Pauli operators terminating at the rough (smooth) boundaries generate logical Z (X) operations on this encoded qubit, as shown in Fig. 1., see Refs. [16, 7] for details.

In this work, we propose and implement a matching decoder to correct for the surface code undergoing an independent and identically distributed depolarising noise model. This error model is such that each qubit experiences either a Pauli- X , $-Y$, or $-Z$ error with probability $p/3$, or no error with probability $(1-p)$. We indicate examples of these single-qubit errors and their corresponding syndromes in Fig. 1.

2.3 The color code

We obtain a decoder for the surface code undergoing depolarising noise by mapping its syndrome onto that of the color code. Here we briefly describe the basic features of the color code, introduced in Ref. [18].

The color code is defined on a three-colorable two-dimensional lattice. We will focus on the regular square-octagon color code with square boundaries, see Fig. 2. Physical qubits are located on the vertices of the lattice, v , and each face f supports two stabilizer operators $S_f^P = \prod_{v \in E_f} P_v$ where $P \in \{X, Z\}$ and E_f denotes the qubits on the boundaries of face f .

We define a set of color labels red, green and blue in bold $\mathcal{C} = \{\mathbf{r}, \mathbf{g}, \mathbf{b}\}$, and each face of the code is assigned a label $\mathbf{u} \in \mathcal{C}$ such that no two faces of the same color touch.

We also assign colors to boundaries. The lattice of interest in Fig. 2 has differently colored

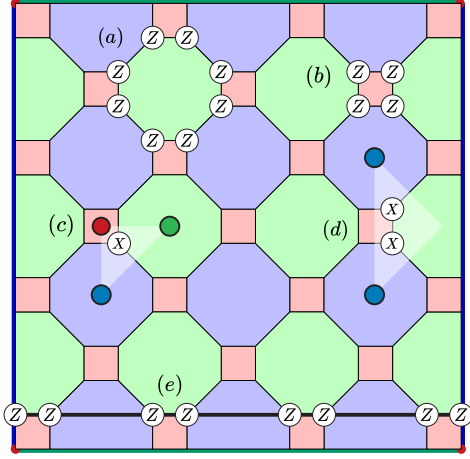


Figure 2: The distance-8 square-octagonal color code with square boundaries. (a) An octagonal stabilizer generator. (b) A square stabilizer generator. (c) a single-qubit X -error. (d) A two-qubit X -error. (e) A horizontal logical operator.

boundaries on its four sides and corners, outlined with their appropriate colors. To be precise, we attribute a color label to a boundary such that the qubits lying on a boundary of color $\mathbf{u} \in \mathcal{C}$ do not support a face of color \mathbf{u} . A corner on the lattice is attributed a color \mathbf{u} if its vertex only supports one face of color \mathbf{u} . We find a \mathbf{u} -colored corner at a vertex where two boundaries of colors \mathbf{v} and \mathbf{w} overlap where $\mathbf{u} \neq \mathbf{v} \neq \mathbf{w} \neq \mathbf{u}$. We note that a corner of color \mathbf{u} is also a member of its two adjacent boundaries that must necessarily have colors \mathbf{v} and \mathbf{w} .

The color code lattice we consider in Fig. 2 features two green boundaries, two blue boundaries, and four red corners. We henceforth refer to the instance of the square-octagon color code shown in Fig. 2 merely as the color code, and alternatively as the square-boundary color code when paying specific attention to its boundary conditions.

The color code encodes two logical qubits with an even code distance d , using $n = 2(d-1)^2 + 2$ physical qubits. We define representatives of the logical operators of the code as follows:

$$\bar{X}_{\mathbf{u}} = \prod_{v \in \delta \mathbf{u}} X_v \text{ and } \bar{Z}_{\mathbf{u}} = \prod_{v \in \delta \mathbf{u}} Z_v \quad (1)$$

where the product is taken over qubits lying on a boundary $\delta \mathbf{u}$ of color $\mathbf{u} = \mathbf{g}, \mathbf{b}$, and obey the commutation relation $\bar{X}_{\mathbf{u}} \bar{Z}_{\mathbf{v}} = -\bar{Z}_{\mathbf{v}} \bar{X}_{\mathbf{u}}$ if and only if $\mathbf{u} \neq \mathbf{v}$. Otherwise, logical operators commute. These represent strings terminating at opposite

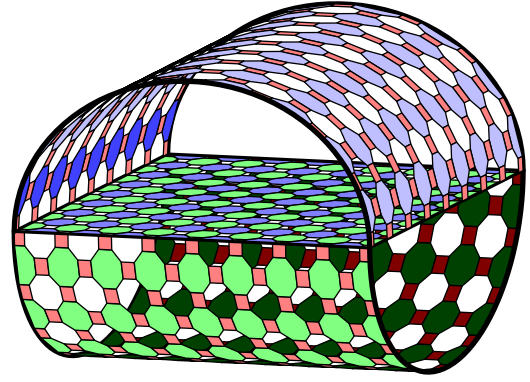


Figure 3: The manifold which preserves the global symmetry of the square-octagonal color code is a punctured torus.

boundaries of the same color, of which we show an example in Fig. 2(e). For now, we restrict our interest to the Pauli- Z stabilizer generators of the color code, and as such we omit the superscript label indicating the stabilizer type, such that $S_f = S_f^Z$. We show examples of an octagonal and a square stabilizer S_f in Fig. 2 (a) and (b) respectively. We note that a defect is attributed the color from \mathcal{C} of the violated stabilizer it lies on, and discuss the decoding of the color code in Sec. 3.

3 Materialised symmetries and decoding

In this section, we review the concept of materialised symmetries for topological quantum codes introduced in Refs. [15, 41, 48, 55]. These symmetries give rise to a parity conservation law on syndrome defects, that is, any error creates an even number of defects on the symmetry. We exploit this parity conservation law to make a matching decoder for the square-boundary color code. In Ref. [48], a global symmetry termed the *unified lattice* was derived for the triangular color code with a single red, blue and green boundary. Notably, the unified lattice presented in Ref. [48] had the topology of a Möbius strip. Due to the unified lattice's improved capacity to handle correctable errors at the boundary [68], Ref. [48] reported improvements in both the threshold and logical failure rates for the triangular color code using decoding on the unified lattice. Motivated by this, we introduce a unified decoder for the color code presented in Sec. 2.3. In addition to introducing the unified lattice decoder, we will also

review the matching decoder on the restricted lattice [44, 45, 46, 47, 49]. In later sections, we will compare the restricted-lattice decoder to the unified-lattice decoder.

3.1 Symmetries of the color code

We use symmetries to obtain matching decoders. We define a symmetry as a subset of stabilizers Σ such that

$$\prod_{s \in \Sigma} s = \mathbf{1}. \quad (2)$$

This definition is such that the elements of the symmetry Σ necessarily give rise to an even number of syndrome defects under any error. Given an appropriate choice of Σ we can thus design a matching decoder, demonstrated with a number of examples [7, 44, 45, 35, 36, 37, 42, 69, 48, 49, 43, 70] which are summarised in Ref. [55].

We find the symmetries of the color code by considering subsets of its stabilizer generators with specific colors [45, 48]. Our goal is to derive a single ‘unifying’ symmetry for the color code with boundaries. We also investigate other color code decoders based on restricted lattices. Interestingly, we find that the square-boundary color code has a unified lattice with a topology that is distinct from that in Ref. [48], and is given by the manifold in Fig. 3. In this subsection, we show how to derive this unified lattice for the square boundary color code, which we will adopt in our matching decoder implementation.

Prior work has established a specific set of bulk symmetries, referred to as restricted lattices on the color code [45]. A \mathbf{uv} -restricted lattice, denoted by $\mathcal{R}_{\mathbf{w}}$ (where $\mathbf{u} \neq \mathbf{v} \neq \mathbf{w} \neq \mathbf{u}$), is defined as

$$\mathcal{R}_{\mathbf{w}} = \{\{S_f\}_{\text{col}(f) \neq \mathbf{w}}\}. \quad (3)$$

We show the restricted lattices $\mathcal{R}_{\mathbf{b}}, \mathcal{R}_{\mathbf{r}}$, and $\mathcal{R}_{\mathbf{g}}$ in Fig. 4(a). Errors in the bulk of the color code preserve defect parity on any single restricted lattice. We show examples of bulk errors on the restricted lattices in Fig. 4(a), where the bottom left error on the lattice flips a single qubit and creates a single defect of each color, and the top right error flips two qubits, and creates two green defects.

Note that the restricted lattices are not symmetries for the color code with boundaries. Errors at the boundary may give rise to a single defect on the restricted lattice, see the error in Fig. 4(a)

(bottom right). We must therefore introduce a boundary operator to complete the restricted lattice symmetry; see Refs. [48, 55] for discussions on boundary operators.

We introduce a boundary operator for each restricted lattice, defined as

$$b_{\mathbf{u}} = \prod_{\text{col}(f) \neq \mathbf{u}} S_f \quad (4)$$

where we take the product over all faces of the restricted lattice, i.e. those not colored \mathbf{u} . This operator is highlighted for each restricted lattice in Fig. 4(b). We note that the boundary operator $b_{\mathbf{u}}$ is supported on all boundaries and corners that are not \mathbf{u} -colored.

We find that the inclusion of the boundary operator with its restricted lattice gives rise to a materialised symmetry [48]. A set that includes this boundary operator, $b_{\mathbf{u}}$, together with the face operators of its respective restricted lattice, $\mathcal{R}_{\mathbf{u}}$, completes a symmetry and thereby allows us to match lone defects on the restricted lattice to the boundary. Explicitly, we write this symmetry as follows

$$\Sigma_{\mathbf{u}} = \{b_{\mathbf{u}}, \mathcal{R}_{\mathbf{u}}\}, \quad \prod_{s \in \Sigma_{\mathbf{u}}} s = \mathbf{1}. \quad (5)$$

We now derive a global symmetry by taking products of smaller symmetries. Relationships among the boundary operators reveal that restricted lattices can be unified by their boundaries, $b_{\mathbf{r}} b_{\mathbf{g}} b_{\mathbf{b}} = \mathbf{1}$. This is discussed in generality in Ref. [48]. Let us describe this unification procedure in steps. We start by examining the subproduct $b_{\mathbf{r}} b_{\mathbf{g}}$ of this equation; this is essentially the combination of the \mathbf{gb} and \mathbf{rb} restricted lattices, as in Fig. 4(b) (centre right). Because the qubits at the right (left) of $\mathcal{R}_{\mathbf{r}}$ are the same as those on the left (right) of a horizontally flipped $\mathcal{R}_{\mathbf{g}}$, this product stitches the symmetries together, thus forming a cylinder. We now add $b_{\mathbf{b}}$, the product of all faces on $\mathcal{R}_{\mathbf{b}}$ to this subproduct. Analogously, the qubits at the top (bottom) of $\mathcal{R}_{\mathbf{r}}$ are the same as that on the bottom (top) of a vertically flipped $\mathcal{R}_{\mathbf{b}}$, and we join them accordingly, as in Fig. 4(b) (left). This creates the manifold seen in Fig. 3.

We now provide an alternate explanation of the stitching procedure adopted to produce the unified lattice. To do so, we consider the syndromes created by errors at the lattice boundary. Let us specifically consider a single qubit

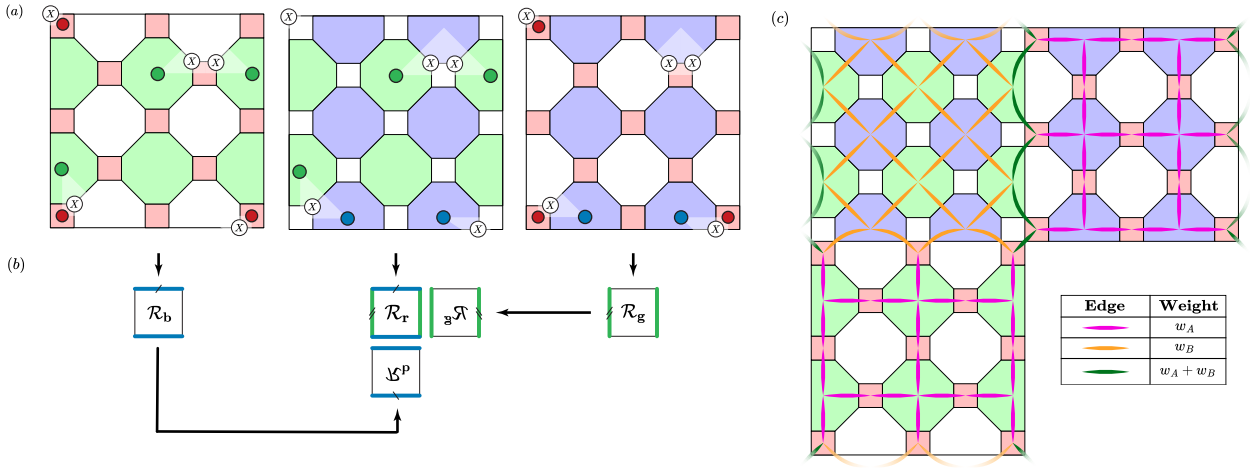


Figure 4: (a) One and two-qubit errors with the corresponding syndromes on the three restricted lattices of the $d = 6$ color code. (b) Stitching the lattices together along the boundary operators. (c) Edge weights for the matching graph. This configuration is empirically chosen to correct for all expected minimum-weight uncorrectable errors, but is non-unique.

error at the lattice boundary, such as the single error shown at the green boundary in Fig. 4(a) (bottom right). We concentrate our attention on the two single defects created by this error on both the \mathcal{R}_r and \mathcal{R}_b lattices. Ideally, these defects should be paired by a single low-weight edge. As such, we can consider stitching these two restricted lattices together along the green boundary, such that these two defects are adjacent on the unified lattice. Next, by analogy, we join \mathcal{R}_r with \mathcal{R}_g along the \mathbf{b} boundaries. The system takes on the topology of a torus with a disk-shaped puncture, again resulting in the manifold of Fig. 3.

Let us additionally examine a single qubit error at the corner of the lattice to show how we stitch the corners of the unified lattice. See for example the top left error in Fig. 4(a). This error produces syndromes only on \mathcal{R}_b and \mathcal{R}_g , not on \mathcal{R}_r . Consequently, we would like our matching decoder to pair defects across any corner of \mathcal{R}_b to the equivalent corner of \mathcal{R}_g . This manifests on the unified lattice as a string stretching from the top of the cylinder diagonally across to the sheet-like handle.

3.2 Matching decoders and code symmetries

We can use symmetries to design decoders based on minimum-weight perfect matching. See Ref. [55] for a discussion. Here, let us briefly review the restricted-lattice decoder and the unified lattice decoder that we use throughout this work.

Prior work has proposed using the individual restricted lattices for error correction [49]. In the restriction-lattice decoder, defects on each of the \mathcal{R}_g and \mathcal{R}_b lattices are matched within the lattice or to the boundary operator that completes the symmetry to produce local correction operators $\{\mathcal{C}_g, \mathcal{C}_b\}$. In Fig. 5(a), for instance, these consist of Pauli corrections on the qubits lying along the highlighted paths. In order to apply a global correction, the union of the local correction operators $\mathcal{C}_g \cup \mathcal{C}_b$ is applied to the underlying physical qubits.

The unified lattice is obtained from the global symmetry of the color code described in the previous subsection, on which errors give rise to pairs of defects. Given a collection of defects on the unified lattice, the matching-based decoding problem reduces to finding a pairing between these defects that is created by an underlying error with maximal probability. Towards this goal, a matching between a pair of defects is assigned a weight inversely dependent on the probability of errors on all the qubits along the shortest path connecting the two. As a result, decoding is equivalent to finding the perfect matching of minimum weight on this weighted graph on the unified lattice.

It was seen in Ref. [48] that the choice of edge weights on this graph can be determined by examining simple error patterns at the bulk, boundary and corners of the unified lattice. We apply the same basic concept in this work, but we set

the edge weights as free parameters w_A to bulk edges on \mathcal{R}_g and \mathcal{R}_b , and w_B on \mathcal{R}_r , as shown in Fig 4(c), where we adopt a matching graph which allows for the correction of all expected errors when the global symmetry is respected. In the following sections, specific weights were found by numerical tests depending on the noise model.

Given a matching on a particular lattice – unified or restricted – we now lay out the procedure to obtain the relevant correction to the logical qubit subjected to X errors on the underlying physical qubits. Note that the following procedure is valid for both the restricted and unified lattices. We first apply the Pauli- Z corrections to qubits along the shortest paths determined by the matching algorithm. This returns the logical qubit to the code space. During this process, we track the parity of a logical operator, here the vertical Z logical depicted in Fig. 5. We successfully return the encoded qubit to the correct logical state if the decoder has determined that the parity of matched paths *across* this logical is the same as the parity of physical Pauli errors on the qubits *along* the logical, a method discussed in more detail in Ref. [55, 48]. For example, in Fig. 5, if the decoder finds the yellow matching, the corresponding correction, together with the initial error, applies \bar{X} to the encoded qubit; the grey matching returns the lattice to the correct state. Henceforth, we will refer to the MWPM-based decoder applied to syndrome data on the restricted (unified) lattice as the restricted (unified) decoder.

4 Mapping surface code errors to color code errors

We have introduced the surface code and the color code in the previous section, and proposed a decoding procedure for bit-flip errors acting on the color code by matching on either the restricted lattice or the unified lattice. We now describe the local mapping from the surface code onto the square-octagonal color code [59], and introduce a mapping from depolarising errors on the former to correlated bit-flip errors on the latter. We exploit this mapping to adopt the restricted-lattice and unified-lattice color code decoder for the surface code undergoing depolarising noise.

The two-dimensional color code is locally

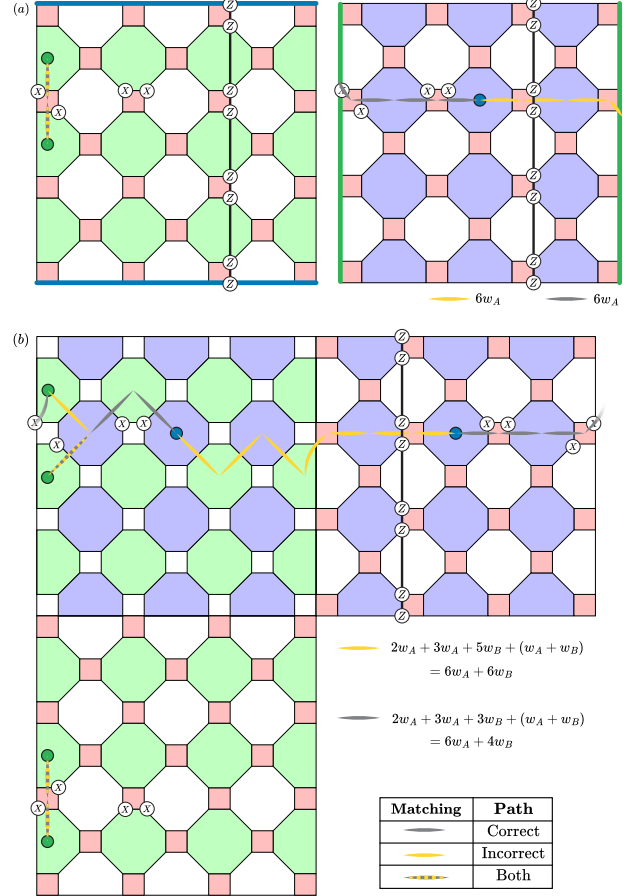


Figure 5: Here we consider a weight-4 error on a $d = 8$ color code lattice, which can be interpreted as a weight-2 error on the $d = 4$ surface code as per Sec. 4, further shown in Fig. 7. We examine correction operators found by matching using (a) the restricted decoder, and (b) the unified decoder. A correct (incorrect) pairing of defects that returns the encoded qubit to the correct (incorrect) state in the logical subspace is depicted in grey (yellow). Matching on the restricted lattice is unable to reliably determine the correct defect pairing. Matching on the unified lattice correctly pairs the syndrome defects when the error string contains a Pauli- Y error, by having a demonstrably lower weight.

equivalent to two copies of the surface code [71, 58, 60, 59, 61]. We first describe this equivalence via operators, before providing a formalism to reconstruct it. As a guide for the mapping, we show the surface code in blue overlaid with the color code in Fig. 6(a). The figure additionally depicts a second copy of the surface code in green, laid out on the dual lattice, that duplicates the syndrome of the blue copy. Considering two copies of the duplicate syndrome facilitates the unfolding map between two copies of the surface code and the color code.

Let us begin by stating the equivalence between

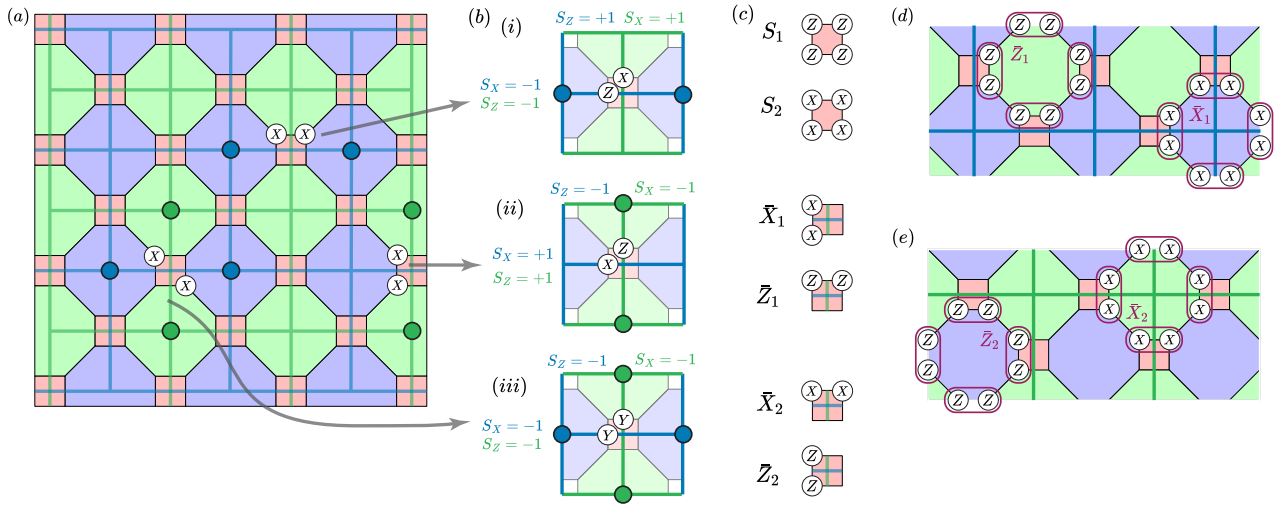


Figure 6: The surface-color code mapping. (a) The color code with an overlay of two surface codes - blue and green - to indicate their positioning with respect to the color code stabilizers, along with weight-2 bit-flip errors and their syndromes on the color code. (b) Correspondence of the indicated weight-2 bit-flip errors on the color code to single-qubit depolarising noise on the surface codes. For a given error on the color code in (a), the corresponding error for the blue (green) surface code is shown on the blue (green) edge across the red squares in (i-iii), with the surface code stabilizer measurements to its left and top displayed in the same color. (c) Stabilizers and logical operators of the $[[4, 2, 2]]$ code defined on a single square with qubits on its vertices. The displayed orientation is used for a red square traversed by a horizontal (vertical) edge of the blue (green) surface code. In the alternate case, an orientation rotated by 90° applies. (d,e) Stabilizers of the (d) blue and (e) green surface codes obtained from the mapping in Sec. 4. The star operators of the blue (green) surface code are Pauli- X stabilizers, where each X operator is encoded as the logical $\bar{X}_1(\bar{X}_2)$ operator of the $[[4, 2, 2]]$ codes shown in (c). The face operators are Pauli- Z stabilizers where each Z operator is encoded as the logical $\bar{Z}_1(\bar{Z}_2)$ operator of the $[[4, 2, 2]]$ codes.

the different lattice elements under this mapping, by considering the surface code displayed with a blue lattice in Fig. 6(a). First, we map the vertex and face operators of this blue surface code onto octagonal face operators of the color code. Specifically, each surface code star operator corresponds to a blue octagon S_f of the color code and, likewise, each face operator of the surface code corresponds to a green octagonal face operator in the color code picture S_f . This is shown in Fig. 6(a), where the star and face operators of the blue surface code align with their corresponding color code stabilizers exactly, thereby illustrating this mapping. We also show the surface code displayed with a green lattice, which represents a rotation of the blue counterpart, and where a similar argument holds.

Now that we have identified stars and plaquettes of each surface code with octagonal face operators of the color code, we must choose errors in the color code picture that give rise to equivalent syndromes in the surface code picture. A Pauli- Z error on the surface code will violate its two adjacent vertex operators, as shown in Fig. 1. In the color code picture, this error will violate the cor-

responding blue octagonal operators as shown in Fig. 6(b)(i) where we indicate this Z -flip on the blue surface code edge. To obtain this violation in the color code picture, we add two horizontal bit-flip errors onto the single red face of the color code on which the edge lies such that its two adjacent blue octagons support a syndrome, see Fig. 6(a). Similarly, a Pauli- X error in the surface code picture will violate two adjacent face operators, which correspond to green octagons in the color code picture in Fig. 6(b)(ii). We find that the error in the color code picture that violates these appropriate color code octagons is a pair of bit-flip errors lying vertically on a single red square-shaped face operator shown in the corresponding diagram in Fig. 6(a).

We can obtain a Pauli- Y error by adding a Pauli- Z error and a Pauli- X error to the same red face of the color code, which results in a diagonal bit-flip pair on the latter. Since the surface code defined using the green lattice is rotated, so is the orientation of the errors it detects, namely a horizontal(vertical) pair of bit-flips on such a red plaquette in the color code picture corresponds to a Pauli- $X(Z)$ error, also shown in Fig. 6(a) and

(b).

We have now proposed an error model that violates pairs of either blue or green face operators according to the mapping from the surface code undergoing a depolarising noise model. As we have alluded, single qubit errors in the surface code picture correspond to two-qubit errors supported on individual red faces of the color code. Indeed, we see that the edges in the surface code picture each have an underlying red face operator in the color code picture, where the red square inherits an orientation from the underlying edge. We therefore identify qubits in the surface code picture with red faces in the color code picture. In what follows, it will be helpful to keep this identification in mind as we describe the mapping more rigorously. We note that this mapping is non-trivial in the sense that the commutation relations of the errors are not preserved. Nevertheless, we find a valid decoder for the surface code with the equivalence we have proposed.

Let us make this mapping more concrete using the more rigorous language of the unfolding map [58, 60, 59]. Specifically, we duplicate the surface code supporting the syndrome, where the second copy differs from the first by a transversal Hadamard rotation. We show the duplicate surface code syndrome on the dual lattice in green in Fig. 6. We then put the two duplicate copies through a folding map, see e.g. Refs. [58, 59, 61]. Specifically, we will consider the unfolding map of Ref. [61] where pairs of surface code qubits are encoded with the $[[4, 2, 2]]$ -code. See also Ref. [72] on this equivalence. This four-qubit code can be regarded as an inner code for a concatenated model with the surface code as the outer code. All together, this concatenated model gives the square-octagon color code we have studied throughout this work.

Under this mapping, the inner $[[4, 2, 2]]$ code can be regarded as a red face of the color code. Indeed each disjoint red face supports the four qubits of the $[[4, 2, 2]]$ code lying on its vertices as shown in Fig. 6(c). Furthermore, the stabilizer group of the inner code is generated by $S^X = XXXX$ and $S^Z = ZZZZ$, which coincide with the red face operators of the square-octagon color code. Moreover, the inner code encodes two logical qubits represented by the operators $\bar{X}_1 = XIXI$, $\bar{Z}_1 = ZZII$, $\bar{X}_2 = XXII$ and $\bar{Z}_2 = ZIZI$ visualised per the convention in

Fig. 6(c).

We then express the stabilizers of the blue and green surface codes respectively, using the two logical operators of the inner code $\{\bar{X}_1, \bar{Z}_1\}$ and $\{\bar{X}_2, \bar{Z}_2\}$, where the first(second) qubit encodes the blue(green) copy of the surface code, as shown in Fig. 6(d) (Fig. 6(e)). We emphasise that the orientation of the edge qubit of the surface code dictates the orientation of the encoding of the $[[4, 2, 2]]$ code. See Fig. 6(c). Indeed, a red square associated with a horizontal (vertical) edge of the blue (green) surface code is encoded as per Fig. 6(c), but when traversed by a vertical edge, this encoding must be rotated by 90° to preserve our mapping.

Hence, in this mapping, a Pauli operator on a surface code qubit corresponds to a pair of Pauli operators on the red stabilizer lying on the associated edge, where this pair defines the appropriate logical operator of the $[[4, 2, 2]]$ code. This is summarised in the following equivalences between stabilizers of the surface codes, color codes, and product of logical operators of the inner code

$$S_{SC,b}^X \equiv S_{f,b}^X \equiv \prod_{r \in \partial f} \bar{X}_1^r \quad (6)$$

$$S_{SC,b}^Z \equiv S_{f,g}^Z \equiv \prod_{r \in \partial f} \bar{Z}_1^r \quad (7)$$

$$S_{SC,g}^X \equiv S_{f,g}^X \equiv \prod_{r \in \partial f} \bar{X}_2^r \quad (8)$$

$$S_{SC,g}^Z \equiv S_{f,b}^Z \equiv \prod_{r \in \partial f} \bar{Z}_2^r \quad (9)$$

where the product over $r \in \partial f$ indicates that the logical operator is applied to all four red square operators on the boundary of the appropriate color code stabilizer.

Finally, since the surface code on the green lattice is a duplicate of its blue counterpart up to a Hadamard rotation, we identify certain logical operators of the inner code to complete our mapping. Indeed, under the duplication, with the Hadamard rotation, we have the following relations between pairs of logical operators of the inner code, namely $\bar{X}_1 \equiv \bar{Z}_2$ and $\bar{Z}_1 \equiv \bar{X}_2$. This ensures the syndrome correspondence in Fig. 6(a) such that all depolarising errors on the surface code now have a one-to-one correspondence with a commuting two-qubit bit-flip error model. Most importantly, this enables us to use a decoder for a CSS code [3, 73] to correct after depolarising noise on the surface code, represented in the color code

picture.

5 Surface code performance under depolarising noise

In this section, we study the surface code under depolarising Pauli noise on data qubits and perfect measurements. We compare the performance of the surface code decoder we have introduced to the conventional matching decoder. Under the mapping, these correspond to the unified decoder and the restricted decoder, respectively.

Our analysis begins by counting the number of minimum-weight uncorrectable errors on the lattice for each decoder. The unified decoder provides an advantage that results in a logical error rate suppression in the low physical error-rate regime. We interrogate our analytical estimates numerically using the splitting method [63]. Additionally, we investigate the threshold performance of the code numerically using `PyMatching` [31].

5.1 Correcting Pauli-Y errors on the surface code

Under depolarising noise, $p(Y) = p(X) = p(Z)$. Since $Y = XZ$, breaking Y s down into X and Z creates correlated X and Z errors and corresponding correlated syndromes among the star and plaquette stabilizers. However, these correlations generally elude decoders which treat X and Z errors independently. As we will argue, the unified decoder identifies these correlations.

We use the unified decoder on the syndrome of the surface code mapped onto the color code lattice to correct for a depolarising noise model. This is motivated by the ability of such a decoder to identify Pauli- Y errors on the surface code, since each one gives rise to a distinct syndrome pattern on the unified lattice. Indeed, a Pauli- X error on the blue surface code introduces a syndrome defect on \mathcal{R}_r and \mathcal{R}_b as shown in Fig. 6. Similarly, a Pauli- Z error leads to a syndrome on \mathcal{R}_r and \mathcal{R}_g , and finally a Pauli- Y error is distinguished by its syndrome featuring on all three restricted lattices. Hence, we find that in the presence of at least one Y error, decoding on the unified lattice captures correlations that any one or pair of restricted lattices remain oblivious to.

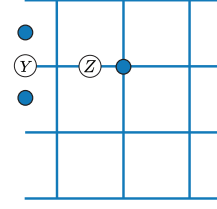


Figure 7: The $d = 4$ surface code, with depolarising errors giving rise to the corresponding set of syndrome defects seen in Fig. 5 on the color code.

To illustrate our argument, we provide an example of a minimum-weight uncorrectable error on the surface code with one Pauli- Y that corresponds to a weight $d/2$ bit-flip error on the color code in Fig. 5. Note that due to the presence of a diagonal bit-flip pair on a red stabilizer of the color code, the syndrome on the unified decoder spans all three sub-lattices and a lower-weight matching path emerges for the successful correction (Fig. 5(a)). We numerically find that this improvement from the unified decoder is valid for a range of values of weights w_A and w_B in the matching graph depicted in Fig. 4(c). For example, $w_A = w_B = 1$ achieves an optimal number of successful corrections.

5.2 Path-counting of failure mechanisms

The advantages of using the unified-lattice decoder are very apparent in the limit where physical error rates p are low. In the regime of asymptotically low p , the logical failure rate is dominated by minimum-weight failure mechanisms, such that the following holds

$$\bar{P}_{0,n} := \lim_{p \rightarrow 0} \bar{P}(p, n) \sim N_{\text{fail}} p^{d/2} \quad (10)$$

where N_{fail} is the entropic term that denotes the number of least-weight errors leading to a logical failure.

In what follows we evaluate the entropic term for both the restricted-lattice decoder and the unified lattice decoder, $N_{\text{fail}}^{\text{res}}$ and $N_{\text{fail}}^{\text{uni}}$ respectively. We find the restricted-lattice decoder has an exponentially larger number of failure mechanisms in the low error rate regime than the unified-lattice decoder as a function of the code distance:

$$N_{\text{fail}}^{\text{res}}/N_{\text{fail}}^{\text{uni}} = 2^{d/2}. \quad (11)$$

We attribute this factor to the ability of the unified-lattice decoder to identify the occurrence

of Pauli-Y errors. First, we count $N_{\text{fail}}^{\text{res}}$ and $N_{\text{fail}}^{\text{uni}}$ exactly. In the following section we interrogate the relative performance of the two decoders numerically, and verify our expressions with exhaustive testing.

We now evaluate the number of least-weight errors that lead the surface code to failure using the restricted-lattice decoder. This decoder cannot distinguish between Pauli-Z or Pauli-Y errors. As such we count the number of weight $d/2$ configurations of errors along a major row of the color code lattice, where we say a major row is a row of d red plaquettes. In contrast, a minor row of the lattice is a row of $d - 1$ red color code plaquettes. Since the lattice is square, it supports d major rows and $d - 1$ minor rows.

The probability of $d/2$ Pauli-Z or Pauli-Y errors occurring is $(1 - p)^{n-d/2}(2p/3)^{d/2}$. There are d major rows on which one of these error configurations can occur, and on each of these rows, there are $d! \times (d/2)!^{-2}$ distinct configurations the $d/2$ errors can find on the d plaquettes of the major row. We therefore find that

$$\bar{P}_{0,n}^{\text{res}} = (1 - p)^n \frac{d}{2} \binom{d}{\frac{d}{2}} \left(\frac{2p}{3(1 - p)} \right)^{\frac{d}{2}} \quad (12)$$

$$\approx \frac{d}{2} \binom{d}{\frac{d}{2}} \left(\frac{2}{3} \right)^{\frac{d}{2}} p^{\frac{d}{2}}. \quad (13)$$

where we include an additional prefactor of one half due to the matching decoder correctly guessing a correction that successfully corrected the error with probability $1/2$ and, on the right hand side, we assume that $1 - p \approx 1$ in the limit of very small p . Expressing the equation this way allows us to read off the entropic term

$$N_{\text{fail}}^{\text{res}} = \frac{d}{2} \binom{d}{\frac{d}{2}} (2/3)^{d/2}. \quad (14)$$

We verify Eq. 14 by exhaustively testing the restricted-lattice decoder for all configurations of weight $d/2$ across the lattice of distances $d = 4, 6$ and 8 .

Let us next evaluate the entropic term for the unified-lattice decoder. As discussed in Sec. 5.1, unlike the restricted-lattice decoder, the unified-lattice decoder allows us to identify correlations between the syndromes caused by Pauli-Y errors to distinguish Pauli-Y errors from Pauli-Z errors. The unified-lattice decoder fails when $d/2$ Pauli-Z errors are configured along a major

row of the lattice. This occurs with probability $(1 - p)^{n-d/2}(p/3)^{d/2}$. These errors may be configured along the d sites of one of the d major rows. We therefore obtain a logical failure rate in the low error rate limit

$$\bar{P}_{0,n}^{\text{uni}} = (1 - p)^n \frac{d}{2} \binom{d}{\frac{d}{2}} \left(\frac{2p}{3(1 - p)} \right)^{\frac{d}{2}} \quad (15)$$

$$\approx \frac{d}{2} \binom{d}{\frac{d}{2}} \left(\frac{1}{3} \right)^{\frac{d}{2}} p^{\frac{d}{2}}. \quad (16)$$

where, as in the restricted-lattice decoder case, a factor $1/2$ is included to account for error configurations where the decoder guesses the correct solution. We can therefore identify the entropic term for the unified-lattice decoder

$$N_{\text{fail}}^{\text{uni}} = \frac{d}{2} \binom{d}{\frac{d}{2}} (1/3)^{d/2}. \quad (17)$$

We again verify this numerically by exhaustively generating all errors of weight $d/2$ on the lattice, and decoding each error using the unified lattice for surface code distances $d = 4, 6$ and 8 , using different values of w_A and w_B . We find that a range of weight configurations minimises the entropic term by achieving Eq. 17, with the constraint that $0 < w_B < \delta$, where δ is a constant changing with code distance. We numerically find that when $w_A \approx w_B$, the entropic term is recovered. With Eq. 14 and Eq. 17 evaluated, the ratio stated at the beginning of this subsection, Eq. 11 is readily checked.

5.3 Low error-rate performance

We adopt the splitting method [63] for the simulation of rare events, in order to numerically investigate logical failure rates at low physical error rates p . These estimates are used to numerically verify the analytical expressions derived from path-counting in Sec. 5.2.

As discussed in Refs. [63, 66], the splitting method is used to evaluate ratios of logical failure rates $R_j = \bar{P}(p_j, n) / \bar{P}(p_{j+1}, n)$ evaluated at physical error rates p_j and p_{j+1} that are similar. The product of many such ratios can be used to interpolate between the high error rate regime where logical failure rates can be obtained using Monte Carlo sampling, and the path-counting regime discussed in the previous subsection where logical error rates are very small. Each term in a given ratio is evaluated by averaging over a

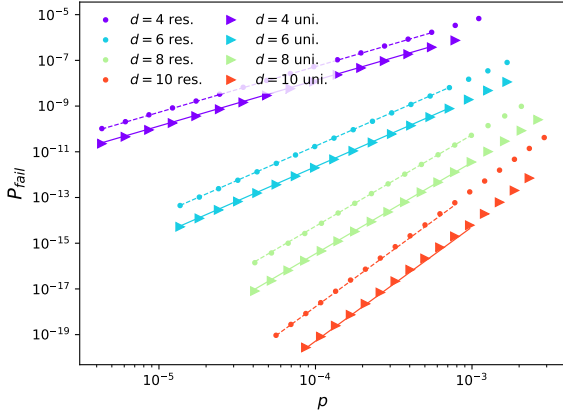


Figure 8: The logical failure rate data (in logarithmic scale) for the surface code under low physical error rates computed using the splitting method [63] to test the restricted and unified decoders for even surface code distances. The dashed and solid lines respectively indicate the fits obtained using Eqs. 14 and 17 for the restricted and unified decoders.

sample of failure paths drawn from the appropriate error distribution. We note that when sampling these errors, we heuristically chose a number of steps such that statistical fluctuations are smaller than the desired logical error rate precision, and discard 50% of the total generated Metropolis samples to ensure that the mixing time of the Markov process was surpassed. We adopt the heuristic sequence of ratios R_j with physical error rates $p_{j+1} = p_j 2^{\pm 1/\sqrt{w_j}}$, as proposed in Ref. [63] to minimise statistical error, where $w_j = \max(d/2, p_j n)$ for a code distance d .

We use this splitting method to numerically interpolate between initial estimates obtained using Monte Carlo sampling at $p = 5\%$, and the analytical path counting results in the low error rate regime derived in Sec. 5.2. Our numerical results show good agreement with our analytical estimates for the logical error rates for both the restricted and unified decoders. In Fig. 8 we compare logical failure rates obtained with the splitting method to our analytical expressions.

5.4 Thresholds

The threshold is a critical physical error rate p_{th} below which a topological code is protected such that error correction succeeds with an exponentially suppressed probability as the code distance increases. Decoding becomes ineffective above this threshold, and the logical failure rate increases with the size of the code. We fit the logical fail-

ure rate data from Monte Carlo sampling near the threshold to a second order Taylor function given by

$$f = Ax^2 + Bx + C \quad (18)$$

where x is the re-scaled error rate $x = (p - p_{\text{th}})^{1/\nu}$ [28]. We find a fitted threshold of $p_{\text{th}} \sim 15.4(7)\%$ for the restricted decoder under depolarising errors using the data shown in Fig. 9, where we obtain a fitted critical exponent $\nu \approx 0.611$, and Taylor fit constants $A = 0.471$, $B = 0.540$ and $C = 0.153$. For the unified decoder, we find that the threshold estimate varies slightly with the chosen values of w_A and w_B . We considered multiple weights w_B with fixed $w_A = 1$ for edges of the matching graph in Fig. 4(c), and find a threshold of $p_{\text{th}} \sim 15.2(1)\%$ using $w_A = 1$ for $w_B = 0.5$, as shown in Fig. 9 where $\nu = 0.664$, $A = 0.606$, $B = 0.452$ and $C = 0.127$. We provide the data for different weight configurations in Appendix B, where the threshold is still comparable to that of the standard surface code, with a minor discrepancy.

6 Color code performance under bit-flip noise

In this section, we compare the performance of the restricted and unified decoders used to decode the color code undergoing independent and identically distributed bit-flip noise and perfect measurements, where a bit flip occurs on each qubit with probability p . We begin by counting the number of minimum-weight uncorrectable errors on the lattice for each decoder. We show that using the unified decoder achieves a higher success rate, providing an advantage in the low physical error-rate regime. We verify the estimates numerically using the splitting method as was done in Sec. 5.3 [63], as well as direct Monte Carlo sampling. We investigate the threshold performance of the code numerically assuming perfect measurements.

6.1 Path-counting

We start by counting the number of minimum-weight uncorrectable error configurations on each row of the color code $n_{\text{fail}}^{\text{res}}$ of distance D supporting $M = \frac{D}{2}$ square plaquettes. We define these new parameters to avoid confusion with the surface code parameters used in earlier sections.

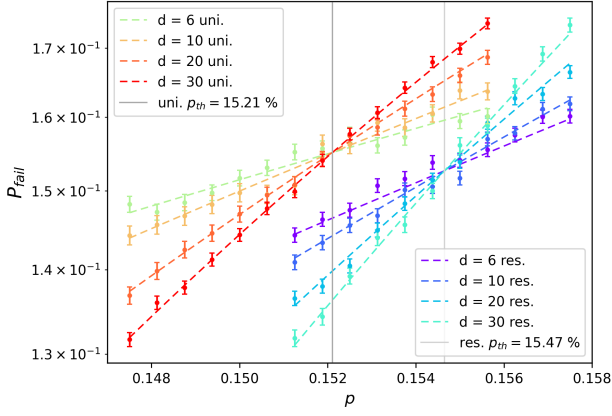


Figure 9: The logical failure rate P_{fail} (in logarithmic scale) as a function of physical error rate p for the restricted and unified surface codes of even distance d , under an i.i.d depolarising noise channel. The threshold p_{th} is indicated by the intersection. Edge weights were set to $w_A = 1$ and $w_B = 0.5$ in Fig. 4(c). Each dashed line indicates the fit to a Taylor expansion for a given system size, and the error bars show the standard deviation of the mean logical failure rate where each data point is collected using $> 6 \times 10^4$ Monte Carlo samples.

Since the lattice is square, there are M such rows. Using the restricted decoder, the number of errors that cannot be corrected is given by counting configurations consisting of errors lying on such rows. The possible paths an error can take depend on the configuration of X errors in an error string lying on each red square of a row, where each square can support a single-qubit X error, or a two-qubit error lying either on the two vertex qubits of a horizontal edge, or on diagonal vertices of the square. Placing any of these four combinations on $\lfloor M/2 \rfloor$ locations on a row accounts for a number of strings

$$n_{\text{fail}}^{\text{res}} = \sum_{k=0}^{\lfloor \frac{M}{2} \rfloor} \binom{M}{k} \binom{M-k}{M-2k} 4^k 4^{M-2k}, \quad (19)$$

where k is the number of red plaquettes that support a weight-2 error, and $M-2k$ the number of weight-1 errors. Hence, the total number of logical failures on the restricted lattice corresponds to the number of equivalent configurations supported on all M rows, namely

$$\begin{aligned} N_{\text{fail}}^{\text{res}} &= \frac{1}{2} n_{\text{fail}}^{\text{res}} M \\ &= \frac{M}{2} \sum_{k=0}^{\lfloor \frac{M}{2} \rfloor} \binom{M}{k} \binom{M-k}{k} 4^{M-k} \end{aligned} \quad (20)$$

where the prefactor of one half is due to the classical decoder misidentifying the right correction with a 50% probability.

We consider the MWPM problem on the unified lattice shown in Fig. 4(c). Unlike decoding on the restricted lattice, a diagonal error gives rise to a syndrome mapped onto all sub-lattices of the unified lattice, as depicted in Fig. 5. On each row supporting M square plaquettes, the number of weight- M error configurations that cannot be corrected is given by

$$n_{\text{fail}}^{\text{uni}} = \sum_{k=0}^{\lfloor \frac{M}{2} \rfloor} \binom{M}{k} \binom{M-k}{M-2k} 2^k 4^{M-2k}, \quad (21)$$

where k is the number of weight-2 errors that can reside on each square plaquette. As for the restricted lattice, the total number of logical failures when decoding on the unified lattice is

$$\begin{aligned} N_{\text{fail}}^{\text{uni}} &= \frac{1}{2} n_{\text{fail}}^{\text{uni}} M \\ &= \frac{M}{2} \sum_{k=0}^{\lfloor \frac{M}{2} \rfloor} \binom{M}{k} \binom{M-k}{k} 4^{M-\frac{3}{2}k}. \end{aligned} \quad (22)$$

We numerically verified these estimate by exhaustively generating all error strings of weight M on each M -square row of the color code, mapping the error configuration respectively to a restricted and unified lattice, and running the MWPM decoder.

6.2 Low error-rate performance

We implement the splitting method under the assumption of errors drawn from an i.i.d bit-flip noise on the color code, which can be written as

$$\pi_j(E) = p_j^{|E|} (1-p_j)^{n-|E|} \quad (23)$$

where n is the number of physical qubits in the color code, $|E|$ the weight of an error operator E . Once again, we adopt the sequence $p_{j+1} = p_j 2^{\pm 1/\sqrt{w_j}}$ to split ratios [63] to minimise statistical error, where $w_j = \max(d/2, p_j n)$ for a code distance d . We numerically interpolate between initial estimates obtained using Monte Carlo sampling at physical error rate $p = 5\%$, and the analytical path counting results in the low error rate regime shown in Sec. 6.1. The results are shown in Fig. 10 where a MWPM decoder was executed on the color code restricted and unified

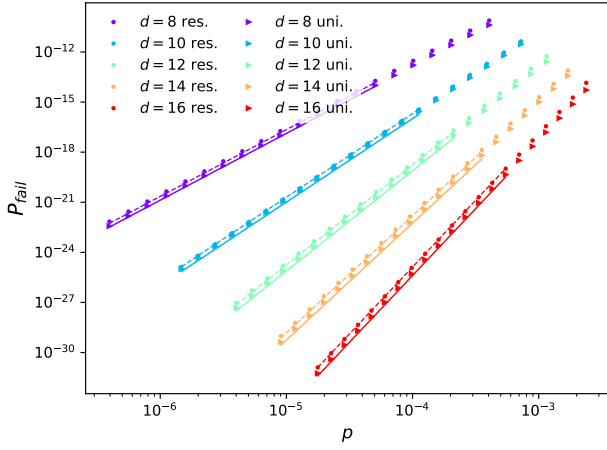


Figure 10: Logical failure rates at low physical error rates (in logarithmic scale) computed using the splitting method [63] with the restricted and unified decoders on the color code under i.i.d bit-flip noise. For the latter, edge weights were set to $w_A = 1.1, w_B = 1$. The dashed and solid lines respectively indicate the fits using Eq. 20 and 22.

lattices respectively, under bit-flip noise and perfect measurements. For the distances considered, the analytic estimates in Eqs. 20 and 22 accurately predict the dominant behaviour of the error chains in the low error rate regime, as indicated by the solid line fits. We note that in order to correct for all expected minimum-weight configurations for bit-flip errors on the color code there is a constraint on the edge weights, namely that $w_A > w_B$, which does not exist in the case of depolarising errors on the even-distance surface code.

6.3 Thresholds

The restricted decoder recovers the threshold $p_{\text{th}} \sim 10.2\%$ under independent and identically distributed bit-flip noise, see Fig. 11, where we obtain a fitted critical exponent $\nu = 0.628$, and Taylor fit constants $A = -0.0419$, $B = 1.136$ and $C = 0.165$ for Eq. 18. This is consistent with that of the surface code on a square lattice [28, 45]. In the case of the unified decoder, we considered multiple weights w_A with $w_B = 1$ for edges of the matching graph in Fig. 4(c), and find a peak in threshold at $p_{\text{th}} \sim 10.1(3)\%$ around $w_A = 2.1$ (Fig. 11). Surprisingly, the unified decoder demonstrates a marginally lower threshold than the restricted decoder. This pattern is consistent with the threshold comparison applied to the surface code in Sec. 5.4. We hypothesize that

in the near-threshold regime, the presence of additional defects on the unified lattice contributes to an increased number of incorrect minimum-weight paths that the decoder may choose, leading to this reduction.

7 Discussion

In conclusion, we demonstrated a method to reduce the logical error rate of the surface code under depolarising errors. This was done by first outlining a mapping from the surface to the color code. We then showed that this noise model on the surface code can be mapped onto a set of two-qubit bit-flip errors on the color code. Subsequently, we constructed a unified lattice in the form of a punctured torus, for use under a minimum-weight perfect matching decoder to correct high-weight errors. To evaluate the performance of our decoder we analytically and numerically investigated the logical failure rates at low physical error rates and verified their agreement. For even code distances, our unified decoder achieves a least-weight correction for all weight $d/2$ depolarising errors on the surface code. We also demonstrated an exponential suppression in the logical error rate for the color code under independent and identically distributed bit-flip noise, extending the work in Ref. [48] to the square-octagonal color code with boundaries.

It will be interesting to determine how the performance of our decoder compares to other practical decoders that are designed to account for depolarising noise, both in terms of error-correction performance and runtime. For instance, both iterative [45, 56] and belief-matching decoders [57] have been proposed to augment matching decoders with the ability to take advantage of correlations between the syndromes measured by the star and plaquette operators. To complete such a comparison, it will first be necessary to determine the performance of other decoding algorithms in the limit of low error rate. Of course, the matching decoder we have presented here is compatible with other decoding techniques such as belief-matching and two-stage decoding [65, 74], at the cost of added computational complexity, and it may be interesting to design versatile decoders for other codes by adapting these techniques. However, it is important to note that our decoder only

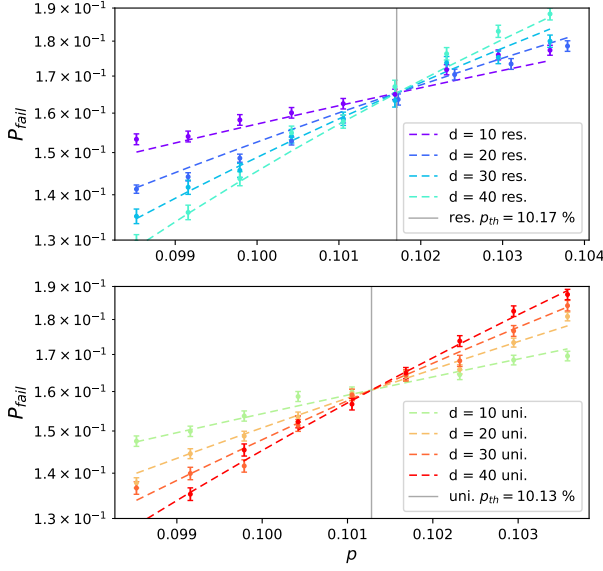


Figure 11: The logical failure rate P_{fail} (in logarithmic scale) as a function of physical error rate p for codes of different distance d , under an i.i.d bit-flip noise model. The threshold indicated by the intersection is found to be $p_{\text{th}} \sim 10.2\%$ for the restricted decoder (top). The unified decoder threshold (bottom) is $p_{\text{th}} \sim 10.1(3)\%$, using edge weights $w_A = 2.1$ and $w_B = 1$ with the matching graph in Fig. 4(c). Each dashed line indicates this fitting for a given system size, and the error bars show the standard deviation of the mean logical failure rate where each data point is collected using $> 6 \times 10^4$ Monte Carlo samples.

requires a single invocation of an MWPM decoder to address Pauli correlations. As a result, it benefits from the expected linear time complexity of MWPM [31], with only a constant factor increase of 4 in the size of the matching graph. One could also readily combine our decoder with fast decoders such as union-find [14, 75, 76] or weighted union-find [77] to improve its runtime. We note that we can also generalise our versatile matching decoder to deal with non-trivial syndrome correlations that occur in fault-tolerant error correction [28, 50, 17], and may be adapted for use in other variations of the surface code with boundaries such as Floquet codes [78, 54]. One might also explore generalising our decoding method to surface codes with alternative topologies, or perhaps using other color code symmetries. Both of these directions are discussed in Appendix F of Ref. [48].

Broadly speaking, we have shown that we can obtain a matching decoder for the surface code capable of correcting depolarising noise by map-

ping its syndrome onto an enlarged state space, namely, the syndrome of the color code. It will be interesting to explore this method further to determine the extent to which it generalises. For instance, it is compelling to investigate whether we obtain near-optimal performance using our mapped decoder for different lattice geometries [66]. Another important case to consider is the circuit noise model, where decoding is conducted on a $(2+1)$ -dimensional syndrome history. Trivially, we can adapt the matching graph obtained from the unified lattice for this setting by extending it along the time direction in the standard way [7, 41, 55]. Nevertheless, it will be very interesting if the novel decoding methods we have proposed can be generalised to offer an advantage in correcting the types of correlated errors the circuit noise model can introduce. Perhaps, for instance, the edge weights on the unified lattice can be chosen appropriately to account for correlated errors such as hook errors. Perhaps there is even some other alternative lattice that can be found to offer an advantage in dealing with this more complicated noise model. More generally, it is exciting to consider other decoding problems on extended Hilbert spaces. This may lead to practical, high-performance decoders for more sophisticated codes, as well as more exotic correlated noise models. We leave this exploration to future work.

Acknowledgments

We thank N. Delfosse for helpful conversations. A.B. and K.S. are grateful to the organizers of the IBM Quantum Error Correction Summer School where parts of this collaboration took place. A.B. acknowledges funding from the EPSRC Centre for Doctoral Training in Delivering Quantum Technologies at UCL, Grant No. EP/S021582/. B.J.B. received support from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 897158. B.J.B. changed affiliation to IBM Quantum during the preparation of the present manuscript.

Data Availability

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

reasonable request. The source code and data used in the figures in this manuscript can be accessed at <https://github.com/abenhemou/unimatch>.

References

- [1] Peter W Shor. “Scheme for reducing decoherence in quantum computer memory”. *Physical review A* **52**, R2493 (1995).
- [2] Peter W Shor. “Fault-tolerant quantum computation”. In Proceedings of 37th conference on foundations of computer science. Pages 56–65. IEEE (1996).
- [3] A Robert Calderbank and Peter W Shor. “Good quantum error-correcting codes exist”. *Physical Review A* **54**, 1098 (1996).
- [4] Barbara M Terhal. “Quantum error correction for quantum memories”. *Reviews of Modern Physics* **87**, 307 (2015).
- [5] Benjamin J Brown, Daniel Loss, Jiannis K Pachos, Chris N Self, and James R Wootton. “Quantum memories at finite temperature”. *Reviews of Modern Physics* **88**, 045005 (2016).
- [6] Earl T Campbell, Barbara M Terhal, and Christophe Vuillot. “Roads towards fault-tolerant universal quantum computation”. *Nature* **549**, 172–179 (2017).
- [7] Eric Dennis, Alexei Kitaev, Andrew Landahl, and John Preskill. “Topological quantum memory”. *Journal of Mathematical Physics* **43**, 4452–4505 (2002).
- [8] Guillaume Duclos-Cianci and David Poulin. “Fast decoders for topological quantum codes”. *Physical review letters* **104**, 050504 (2010).
- [9] James R Wootton and Daniel Loss. “High threshold error correction for the surface code”. *Physical review letters* **109**, 160503 (2012).
- [10] Hussain Anwar, Benjmain J. Brown, Earl. T. Campbell, and Dan E. Browne. “Fast decoders for qudit topological codes”. *New J. Phys.* **16**, 063038 (2014).
- [11] Adrian Hutter, James R Wootton, and Daniel Loss. “Efficient markov chain monte carlo algorithm for the surface code”. *Physical Review A* **89**, 022326 (2014).
- [12] Giacomo Torlai and Roger G. Melko. “Neural decoder for topological codes”. *Phys. Rev. Lett.* **119**, 030501 (2017).
- [13] Pavel Panteleev and Gleb Kalachev. “Degenerate quantum ldpc codes with good finite length performance”. *Quantum* **5**, 585 (2021).
- [14] Nicolas Delfosse and Naomi H. Nickerson. “Almost-linear time decoding algorithm for topological codes”. *Quantum* **5**, 595 (2021).
- [15] A Yu Kitaev. “Fault-tolerant quantum computation by anyons”. *Annals of physics* **303**, 2–30 (2003).
- [16] S. B. Bravyi and A. Yu. Kitaev. “Quantum codes on a lattice with boundary” (1998). [arXiv:quant-ph/9811052](https://arxiv.org/abs/quant-ph/9811052).
- [17] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. “Surface codes: Towards practical large-scale quantum computation”. *Phys. Rev. A* **86**, 032324 (2012).
- [18] H. Bombin and M. A. Martin-Delgado. “Topological quantum distillation”. *Phys. Rev. Lett.* **97**, 180501 (2006).
- [19] Hector Bombin and Miguel-Angel Martin-Delgado. “Topological computation without braiding”. *Physical review letters* **98**, 160502 (2007).
- [20] Héctor Bombín. “Gauge color codes: optimal transversal gates and gauge fixing in topological stabilizer codes”. *New Journal of Physics* **17**, 083002 (2015).
- [21] Aleksander Kubica and Michael E. Beverland. “Universal transversal gates with color codes: A simplified approach”. *Phys. Rev. A* **91**, 032330 (2015).
- [22] Laird Egan, Dripto M. Debroy, Crystal Noel, Andrew Risinger, Daiwei Zhu, Debopriyo Biswas, Michael Newman, Muyuan Li, Kenneth R. Brown, Marko Cetina, and Christopher Monroe. “Fault-tolerant control of an error-corrected qubit”. *Nature* **598**, 281–286 (2021).
- [23] C. Ryan-Anderson, J. G. Bohnet, K. Lee, D. Gresh, A. Hankin, J. P. Gaebler, D. Francois, A. Chernoguzov, D. Lucchetti, N. C.

- Brown, T. M. Gatterman, S. K. Halit, K. Gilmore, J. A. Gerber, B. Neyenhuis, D. Hayes, and R. P. Stutz. “Realization of real-time fault-tolerant quantum error correction”. *Phys. Rev. X* **11**, 041058 (2021).
- [24] Sebastian Krinner, Nathan Lacroix, Ants Remm, Agustin Di Paolo, Elie Genois, Catherine Leroux, Christoph Hellings, Stefania Lazar, Francois Swiadek, Johannes Herrmann, Graham J. Norris, Christian Kraglund Andersen, Markus Müller, Alexandre Blais, Christopher Eichler, and Andreas Wallraff. “Realizing repeated quantum error correction in a distance-three surface code”. *Nature* **605**, 669–674 (2022).
- [25] Neereja Sundaresan, Theodore J. Yoder, Youngseok Kim, Muyuan Li, Edward H. Chen, Grace Harper, Ted Thorbeck, Andrew W. Cross, Antonio D. Córcoles, and Maika Takita. “Demonstrating multi-round subsystem quantum error correction using matching and maximum likelihood decoders”. *Nature Communications* **14**, 2852 (2023).
- [26] Google Quantum AI. “Suppressing quantum errors by scaling a surface code logical qubit”. *Nature* **614**, 676–681 (2023).
- [27] Riddhi S. Gupta, Neereja Sundaresan, Thomas Alexander, Christopher J. Wood, Seth T. Merkel, Michael B. Healy, Marius Hillenbrand, Tomas Jochym-O’Connor, James R. Wootton, Theodore J. Yoder, Andrew W. Cross, Maika Takita, and Benjamin J. Brown. “Encoding a magic state with beyond break-even fidelity” (2023).
- [28] Chenyang Wang, Jim Harrington, and John Preskill. “Confinement-higgs transition in a disordered gauge theory and the accuracy threshold for quantum memory”. *Annals of Physics* **303**, 31–58 (2003).
- [29] Robert Raussendorf and Jim Harrington. “Fault-tolerant quantum computation with high threshold in two dimensions”. *Phys. Rev. Lett.* **98**, 190504 (2007).
- [30] Jack Edmonds. “Paths, trees, and flowers”. *Canadian Journal of mathematics* **17**, 449–467 (1965).
- [31] Oscar Higgott and Craig Gidney. “Sparse blossom: correcting a million errors per core second with minimum-weight matching” (2023).
- [32] Austin G. Fowler. “Coping with qubit leakage in topological codes”. *Phys. Rev. A* **88**, 042308 (2013).
- [33] Austin G. Fowler and John M. Martinis. “Quantifying the effects of local many-qubit errors and nonlocal two-qubit errors on the surface code”. *Phys. Rev. A* **89**, 032316 (2014).
- [34] Adrian Hutter and Daniel Loss. “Breakdown of surface-code error correction due to coupling to a bosonic bath”. *Phys. Rev. A* **89**, 042334 (2014).
- [35] Naomi H. Nickerson and Benjamin J. Brown. “Analysing correlated noise on the surface code using adaptive decoding algorithms”. *Quantum* **3**, 131 (2019).
- [36] David K Tuckett, Stephen D Bartlett, Steven T Flammia, and Benjamin J Brown. “Fault-tolerant thresholds for the surface code in excess of 5% under biased noise”. *Physical review letters* **124**, 130501 (2020).
- [37] J Pablo Bonilla Ataides, David K Tuckett, Stephen D Bartlett, Steven T Flammia, and Benjamin J Brown. “The xzzx surface code”. *Nature communications* **12**, 2172 (2021).
- [38] Armands Strikis, Simon C. Benjamin, and Benjamin J. Brown. “Quantum computing is scalable on a planar array of qubits with fabrication defects” (2021).
- [39] Adam Siegel, Armands Strikis, Thomas Flatters, and Simon Benjamin. “Adaptive surface code for quantum error correction in the presence of temporary or permanent defects” (2022).
- [40] Sophia Fuhui Lin, Joshua Vizlai, Kaitlin N. Smith, Gokul Subramanian Ravi, Charles Yuan, Frederic T. Chong, and Benjamin J. Brown. “Empirical overhead of the adapted surface code on defective qubit arrays” (2023).
- [41] Benjamin J Brown and Dominic J Williamson. “Parallelized quantum error correction with fracton topological codes”. *Physical Review Research* **2**, 013303 (2020).

- [42] Georgia M Nixon and Benjamin J Brown. “Correcting spanning errors with a fractal code”. *IEEE Transactions on Information Theory* **67**, 4504–4516 (2021).
- [43] Jonathan F. San Miguel, Dominic J. Williamson, and Benjamin J. Brown. “A cellular automaton decoder for a noise-bias tailored color code”. *Quantum* **7**, 940 (2023).
- [44] David S. Wang, Austin G. Fowler, Charles D. Hill, and Lloyd C. L. Hollenberg. “Graphical algorithms and threshold error rates for the 2d color code”. *Quantum Info. Comput.* **10**, 780–802 (2010).
- [45] Nicolas Delfosse and Jean-Pierre Tillich. “A decoding algorithm for css codes using the x/z correlations”. In 2014 IEEE International Symposium on Information Theory. Pages 1071–1075. IEEE (2014).
- [46] Christopher Chamberland, Aleksander Kubica, Theodore J Yoder, and Guanyu Zhu. “Triangular color codes on trivalent graphs with flag qubits”. *New Journal of Physics* **22**, 023019 (2020).
- [47] Michael E. Beverland, Aleksander Kubica, and Krysta M. Svore. “Cost of universality: A comparative study of the overhead of state distillation and code switching with color codes”. *PRX Quantum* **2**, 020341 (2021).
- [48] Kaavya Sahay and Benjamin J Brown. “Decoder for the triangular color code by matching on a möbius strip”. *PRX Quantum* **3**, 010310 (2022).
- [49] Aleksander Kubica and Nicolas Delfosse. “Efficient color code decoders in $d \geq 2$ dimensions from toric code decoders”. *Quantum* **7**, 929 (2023).
- [50] Robert Raussendorf, Sergey Bravyi, and Jim Harrington. “Long-range quantum entanglement in noisy cluster states”. *Phys. Rev. A* **71**, 062313 (2005).
- [51] Oscar Higgott and Nikolas P. Breuckmann. “Subsystem codes with high thresholds by gauge fixing and reduced qubit overhead”. *Phys. Rev. X* **11**, 031039 (2021).
- [52] Craig Gidney, Michael Newman, and Matt McEwen. “Benchmarking the Planar Honeycomb Code”. *Quantum* **6**, 813 (2022).
- [53] Adam Paetznick, Christina Knapp, Nicolas Delfosse, Bela Bauer, Jeongwan Haah, Matthew B. Hastings, and Marcus P. da Silva. “Performance of planar floquet codes with majorana-based qubits”. *PRX Quantum* **4**, 010310 (2023).
- [54] Markus S. Kesselring, Julio C. Magdalena de la Fuente, Felix Thomsen, Jens Eisert, Stephen D. Bartlett, and Benjamin J. Brown. “Anyon condensation and the color code” (2022).
- [55] Benjamin J Brown. “Conservation laws and quantum error correction: towards a generalised matching decoder” (2022).
- [56] Austin G. Fowler. “Optimal complexity correction of correlated errors in the surface code” (2013).
- [57] Oscar Higgott, Thomas C Bohdanowicz, Aleksander Kubica, Steven T Flammia, and Earl T Campbell. “Improved decoding of circuit noise and fragile boundaries of tailored surface codes”. *Physical Review X* **13**, 031007 (2023).
- [58] Hector Bombin, Guillaume Duclos-Cianci, and David Poulin. “Universal topological phase of two-dimensional stabilizer codes”. *New Journal of Physics* **14**, 073048 (2012).
- [59] Aleksander Kubica, Beni Yoshida, and Fernando Pastawski. “Unfolding the color code”. *New Journal of Physics* **17**, 083026 (2015).
- [60] Arjun Bhagoji and Pradeep Sarvepalli. “Equivalence of 2d color codes (without translational symmetry) to surface codes”. In 2015 IEEE International Symposium on Information Theory (ISIT). Page 1109. (2015).
- [61] Ben Criger and Barbara Terhal. “Noise thresholds for the $[[4, 2, 2]]$ -concatenated toric code” (2016).
- [62] Charles H Bennett. “Efficient estimation of free energy differences from monte carlo data”. *Journal of Computational Physics* **22**, 245–268 (1976).
- [63] Sergey Bravyi and Alexander Vargo. “Simulation of rare events in quantum error correction”. *Physical Review A* **88** (2013).
- [64] Thomas M Stace and Sean D Barrett. “Error correction and degeneracy in surface

- codes suffering loss”. *Physical Review A* **81**, 022317 (2010).
- [65] Ben Criger and Imran Ashraf. “Multi-path summation for decoding 2d topological codes”. *Quantum* **2**, 102 (2018).
- [66] Michael E Beverland, Benjamin J Brown, Michael J Kastoryano, and Quentin Marol-leau. “The role of entropy in topological quantum error correction”. *Journal of Statistical Mechanics: Theory and Experiment* **2019**, 073404 (2019).
- [67] Daniel Gottesman. “Theory of fault-tolerant quantum computation”. *Physical Review A* **57**, 127–137 (1998).
- [68] Craig Gidney and Cody Jones. “New circuits and an open source decoder for the color code” (2023). [arXiv:2312.08813](#).
- [69] Basudha Srivastava, Anton Frisk Kockum, and Mats Granath. “The xyz^2 hexagonal stabilizer code”. *Quantum* **6**, 698 (2022).
- [70] Eric Huang, Arthur Pesah, Christopher T. Chubb, Michael Vasmer, and Arpit Dua. “Tailoring three-dimensional topological codes for biased noise” (2022). [arXiv:2211.02116](#).
- [71] Beni Yoshida. “Classification of quantum phases and topology of logical operators in an exactly solved model of quantum codes”. *Annals of Physics* **326**, 15–95 (2011).
- [72] Courtney G Brell, Steven T Flammia, Stephen D Bartlett, and Andrew C Doherty. “Toric codes and quantum doubles from two-body hamiltonians”. *New Journal of Physics* **13**, 053039 (2011).
- [73] Andrew M Steane. “Simple quantum error-correcting codes”. *Physical Review A* **54**, 4741 (1996).
- [74] Joschka Roffe, Lawrence Z Cohen, Ar-manda O Quintavalle, Daryus Chandra, and Earl T Campbell. “Bias-tailored quantum ldpc codes”. *Quantum* **7**, 1005 (2023).
- [75] Tim Chan and Simon C. Benjamin. “Strictly local union-find” (2023).
- [76] Sam J. Griffiths and Dan E. Browne. “Union-find quantum decoding without union-find” (2023).
- [77] Shilin Huang, Michael Newman, and Kenneth R. Brown. “Fault-tolerant weighted union-find decoding on the toric code”. *Phys. Rev. A* **102**, 012419 (2020).
- [78] Jeongwan Haah and Matthew B. Hastings. “Boundaries for the Honeycomb Code”. *Quantum* **6**, 693 (2022).

A Pseudocode for unified decoding on the surface code

Here, we outline the Unified decoder presented in the main text. The procedure requires a set of syndrome measurements of the color code stabilizers $\sigma_{c.c.}$, a subroutine which restricts $\sigma_{c.c.}$ to syndromes on restricted lattices, and a subroutine which builds the unified lattice as prescribed in Fig. 4. A minimum-weight perfect matching decoder `mwpmdcoder` is then used to decode the syndrome and provide a correction \bar{C} . The correction operator returns the code to the codespace, and its success is given by the condition that the decoding algorithm has determined that the parity of the edges in the matching solution crossing a chosen logical representative \bar{Z} is equal to the parity of Pauli errors which occurred on qubits $q \in \bar{Z}$, as mentioned in Sec. 3.2.

Algorithm 1: Unified Decoder

```

1 Require:
   Restricted lattices  $\mathcal{R}_{\mathbf{u}} = \{\{S_f\}_{\text{col}(f) \neq \mathbf{u}}\}$  on a manifold  $G_{\mathcal{R}_{\mathbf{u}}}$ 
   Unified lattice constructor  $\mathcal{U}(\cdot)$ 
   A MWPM decoder mwpmdcoder
2 Input: A color code syndrome  $\sigma_{c.c.}$  on the color code manifold  $G_{c.c.}$ 
3 Output: A correction operator  $\bar{C}$ 
   /* Map syndromes to restricted lattices */
4 foreach  $\mathbf{u} \in \{r, g, b\}$  do
5     foreach  $\sigma_i \in \sigma_{c.c.}$  do
6         if  $\text{color}(\sigma_i) = \mathbf{u}$  then
7              $\mathcal{R}_{\mathbf{u}}(\sigma_i)$  on  $G_{\mathcal{R}_{\mathbf{u}}} \leftarrow \sigma_i$  on  $G_{c.c.}$ 
8         end
9     end
10 end
   /* Construct unified lattice, and map syndrome on the unified lattice */
11  $(G_{\mathcal{U}}, \sigma_{i,\mathcal{U}}) \leftarrow \mathcal{U}(G_{\mathcal{R}_r}, G_{\mathcal{R}_g}, G_{\mathcal{R}_b}, \mathcal{R}_r(\sigma_i), \mathcal{R}_g(\sigma_i), \mathcal{R}_b(\sigma_i))$ 
   /* Decode */
12  $\bar{C} = \text{mwpmdcoder}(G_{\mathcal{U}}, \sigma_{i,\mathcal{U}})$ 
   /* Apply Pauli-Z corrections to qubits along the shortest paths determined by
      the decoder */
13 Apply  $\prod_{q \in \bar{C}} Z_q$ 

```

B Thresholds for the even-distance surface code using varied weights

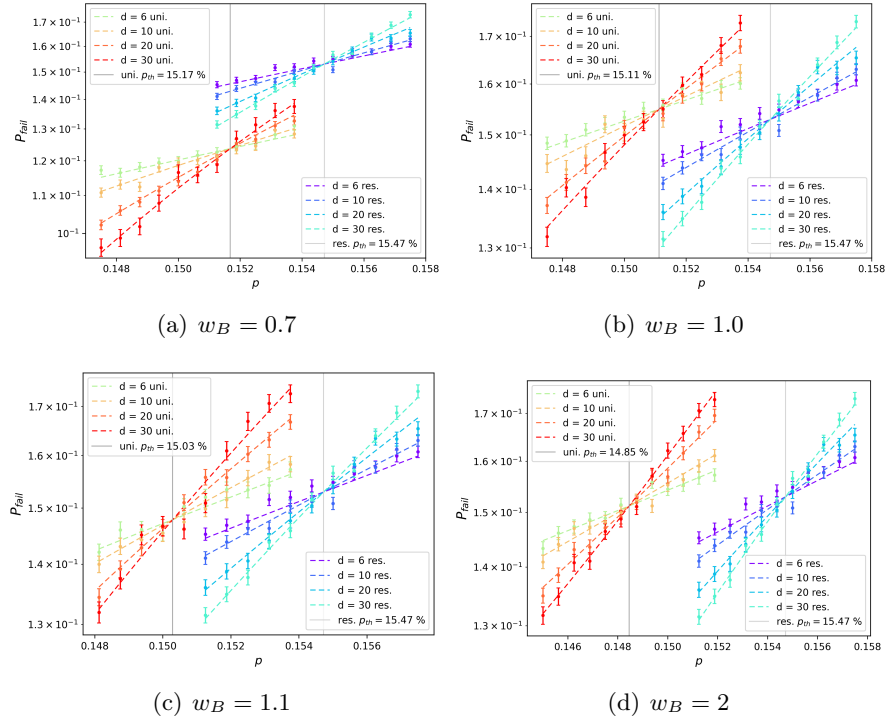


Figure 12: The logical failure rate P_{fail} (in logarithmic scale) as a function of physical error rate p for surface codes of different even distance d , under i.i.d depolarising noise. The restricted decoder threshold is shown against the unified decoder configured with weights $w_A = 1$, and (a) $w_B = 0.7$, (b) $w_B = 1$ and (c) $w_B = 1.1$, (d) $w_B = 2$ using the matching graph in Fig. 4(c). Each dashed line indicates the fit to a Taylor expansion for each system size, and the error bars show the standard deviation of the mean logical failure rate where each data point is collected using $> 6 \times 10^4$ Monte Carlo samples.