Color code decoder with improved scaling for correcting circuit-level noise

Seok-Hyung Lee, Andrew Li, and Stephen D. Bartlett

Centre for Engineered Quantum Systems, School of Physics, The University of Sydney, Sydney, NSW 2006, Australia

Two-dimensional color codes are a promising candidate for fault-tolerant quantum computing, as they have high encoding rates, transversal implementation of logical Clifford gates, and resource-efficient magic state preparation schemes. However, decoding color codes presents a significant challenge due to their structure, where elementary errors violate three checks instead of just two (a key feature in surface code decoding), and the complexity of extracting syndrome is greater. We introduce an efficient color-code decoder that tackles these issues by combining two matching decoders for each color, generalized to handle circuit-level noise by employing detector error models. We provide comprehensive analyses of the decoder, covering its threshold and sub-threshold scaling both for bit-flip noise with ideal measurements and for circuit-level noise. Our simulations reveal that this decoding strategy nearly reaches the best possible scaling of logical failure ($p_{\text{fail}} \sim p^{d/2}$) for both noise models, where p is the noise strength, in the regime of interest for fault-tolerant quantum computing. While its noise thresholds are comparable with other matching-based decoders for color codes (8.2% for bit-flip noise and 0.46% for circuit-level noise), the scaling of logical failure rates below threshold significantly outperforms the best matching-based decoders.

1 Introduction

Two-dimensional (2D) color codes [1, 2] are a family of stabilizer quantum error-correcting codes that can be realized with local interactions on a 2D plane, and provide a promising pathway to implementing fault-tolerant quantum computation. Compared to surface codes [3, 4], color codes have several noteworthy advantages: (i) They have higher encoding rates for the same code distance [5], (ii) all the Clifford gates can be implemented transversally [1], and (iii) an arbitrary pair of commuting logical Pauli product operators can be measured in parallel via lattice surgery [6]. Notably, the transversality of Clifford gates enables highly resource-efficient magic state preparation schemes without distillation [7–9], which can be concatenated with distillation to further improve the quality of the output magic state [10, 11].

Recent experiments have demonstrated significant advancements in the implementation of color codes. The distance-3 Steane code, a well-studied small instance of a color code, has been used to perform various fault-tolerant operations (such as transversal logic gates, magic state preparation, and lattice surgery) on trapped-ion [12–14], neutral atom [15, 16], and superconducting [17] platforms. Furthermore, Refs. [16, 17] also implement the distance-5 color code, with Ref. [17] demonstrating its improved logical error suppression compared to the Steane code.

For these desirable features to be exploited for fault-tolerant quantum computing in practice, we need better decoders for color codes. Surface codes benefit from the many advantages of a decoding approach based on 'matching', which is a standard method to handle errors in codes that can only have *edge-like* errors (namely, elementary errors violate at most two checks). Matching-based decoders can operate both efficiently and near-optimally, and are readily adapted to handle noisy syndrome extraction circuits. In color codes, an elementary error, which is a single-qubit X or Z error, is generally involved in three checks (or stabilizer generators),

Seok-Hyung Lee: seokhyung.lee@sydney.edu.au
Stephen D. Bartlett: stephen.bartlett@sydney.edu.au

and so a matching decoder cannot directly be used. Moreover, considering realistic circuit-level noise makes decoding more difficult because the color code syndrome extraction circuits are more complex than for the surface code. As a result of these deficiencies, existing decoders for the color code do not perform as well as expected either in terms of error thresholds or for sub-threshold scaling of the logical failure rate.

Several approaches to decode errors in color codes have been proposed. The most widely studied methods are the projection decoder and its variants [18–22], which apply minimum-weight perfect matching (MWPM) on two or three restricted lattices (which are specific sub-lattices of the dual lattice of the color code) and then deduce the final correction by a procedure called 'lifting'. These approaches can achieve thresholds of around 8.7% for bit-flip noise [18] and around 0.47% for circuit-level noise [22]. However, they have a fundamental limitation that the logical failure rate p_{fail} scales like $p^{d/3}$ below threshold [20, 22, 23], not $p^{d/2}$, where p is a physical noise strength and d is the code distance. This is because there exist errors with weights $O(d/3)$ that are uncorrectable via the decoder (noting that, in principle, an optimal decoder can correct any error with weight up to $d/2$). Such a drawback may significantly hinder resource efficiency of quantum computing, necessitating a larger code distance to maintain the same p_{fail}, compared to a scenario using a decoder with the expected optimal scaling of $p^{d/2}$. Roughly estimating based on the ansatz $p_{\text{fail}} = \alpha(p/p_{\text{th}})^{\beta d+\eta}$ for the threshold p_{th} and parameters α, β, and η, using a decoder with $\beta = 1/3$ demands about 9/4 times more qubits than an optimal decoder with $\beta = 1/2$ to achieve a similar p_{fail}, assuming p_{th}, α, and η remain similar in both cases.

The Möbius MWPM decoder [23] is another matching-based decoder implemented by applying the MWPM algorithm on a manifold built by connecting the three restricted lattices, which has the topology of a Möbius strip. It achieves a higher threshold of 9.0% under bit-flip noise and, more importantly, a better scaling of $p_{\text{fail}} \sim p^{3d/7}$. The decoder has subsequently been improved to accommodate circuit-level noise and general color-code lattices [24].

Besides these matching-based decoders, there are tensor network decoders (achieving a threshold of 10.9% under bit-flip noise) [25], simulated annealing (10.4%) [26], MaxSAT problem (10.1%) [27], trellis (10.1%) [28], neural network (10.0%) [29], union-find (8.4%) [30], and renormalization group (7.8%) [31]. All of these reported thresholds are for the 6-6-6 (or hexagonal) color-code lattice except that of the decoder based on simulated annealing [26], which is for the 4-8-8 lattice. However, it is currently unclear how these decoders can be adapted to circuit-level noise and how their performance will be. We note that the tensor network decoder can be adapted to treat noisy syndrome measurements in a phenomenological or circuit-level noise model, but it is highly inefficient due to the difficulty of 3D tensor network contraction [32].

In this work, we propose a matching-based color-code decoder, which we call the *concatenated MWPM decoder*, that demonstrates exceptional sub-threshold scaling of the logical failure rate in regimes of interest for fault-tolerant quantum computing. Our decoder functions by 'concatenation' of two MWPM decoders per color, for a total of six matchings. Roughly speaking, this decoder is based on the idea that decoding syndrome on a specific (say, red) restricted lattice returns a prediction of red edges with odd-parity errors, which can be combined with the syndrome data of red checks (specifying red faces with odd-parity errors) and decoded again to predict errors. This process is repeated for each of the three colors and the most probable prediction is selected as the final prediction. We demonstrate that this decoder can be generalized to handle circuit-level noise by using the concept of *detector error model*. This approach to color-code decoding was inspired by a decoding approach for measurement-based quantum computing [33], and here we develop and generalize it further for gate-based quantum computing.

We analyze the performance of the concatenated MWPM decoder against bit-flip and circuit-level noise models by evaluating its noise thresholds and investigating the sub-threshold scaling of logical failure rates. Notably, the logical failure rates of the concatenated MWPM decoder below threshold is shown to be well-described by the scaling $p_{\text{fail}} \sim p^{d/2}$ for both of the noise models within the range of our simulations ($d \leq 31$ for bit-flip noise and $d \leq 21$ for

circuit-level noise). Thanks to this improvement, although it has comparable or slightly lower thresholds (8.2% for bit-flip noise and 0.46% for circuit-level noise), its sub-threshold performance significantly surpasses the projection decoder. Compared to the Möbius decoder, our decoder has a similar scaling factor against d but achieves approximately 3–7 times lower logical failure rates for circuit-level noise when $10^{-4} \lessapprox p \lessapprox 5 \times 10^{-4}$.

A python module for simulating color code circuits and running the concatenated MWPM decoder is publicly available on Github [34].

This paper is structured as follows: In Sec. 2, we provide a brief overview of color codes and their decoding problem, and describe our methodology of analysis including the settings, noise models, and criteria for evaluating decoder performance. In Sec. 3, we introduce the 2D variant of the concatenated MWPM decoder that works when syndrome measurements are perfect and analyze it numerically for bit-flip noise. In Sec. 4, we generalize the decoder by using detector error models to accommodate circuit-level noise including faulty syndrome measurements and present the outcomes of its numerical analyses as well. We conclude with final remarks in Sec. 5.

2 Preliminaries

2.1 Color codes

A 2D color-code lattice indicates a lattice satisfying the following two conditions:

- *3-valent*: Each vertex is connected with three edges.
- *3-colorable*: One of three colors can be assigned to each face in a way that adjacent faces do not have the same color.

Following the usual convention, we use red (r), green (g), and blue (b) as these three colors assigned to faces. Note that each edge is also colorable by the color of the faces that it connects. There are various types of color-code lattices, among which hexagonal (or 6-6-6) lattices are used to test the concatenated MWPM decoder in this work. Nonetheless, since our decoder is described in a form that does not depend on the lattice type, it can also be applied to other color-code lattices such as 4-8-8 lattices [35].

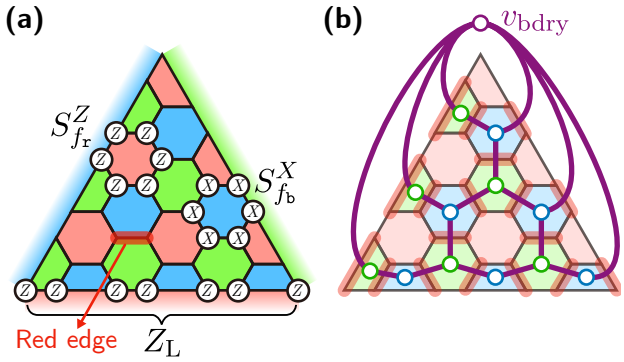

Figure 1: **Example of a triangular color code and its restricted lattice. (a)** The triangular color code with code distance $d = 7$ based on a hexagonal lattice with three boundaries of different colors is displayed. Each face is associated with a pair of Z-type and X-type checks, exemplified by $S^Z_{f_r}$ on a red face and $S^X_{f_b}$ on a blue face, respectively. Logical operators X_L and Z_L can be supported on one of the three boundaries. Each edge is colored by the color of the faces connected by it, as exemplified by the red edge highlighted. **(b)** The red-restricted lattice of the code is illustrated, where vertices and edges are shown as circles and purple lines, respectively. Its vertices consist of blue faces (blue circles), green faces (green circles), and an additional boundary vertex v_{bdry} (purple circle). Each of its edges corresponds to a red edge of the original lattice, which are highlighted as thick red lines.

Given a 2D color-code lattice, a 2D color code [1] is defined on qubits placed at the vertices of the lattice. The code space of the color code is stabilized by two types of *checks* S^X_f and S^Z_f for each face f, which are defined as

$$S^X_f := \prod_{v \in f} X_v, \qquad S^Z_f := \prod_{v \in f} Z_v, \tag{1}$$

where X_v and Z_v are the Pauli-X and Z operators on the qubit placed at v and the notation '$v \in f$' means that the vertex v is included in f. In other words, the code space is the common $+1$ eigenspace of these operators. We categorize checks according to their Pauli types and the colors of the corresponding faces; for example, if f is a red face, S^X_f (S^Z_f) is a red X-type (Z-type) check.

To define a single logical qubit with a color code, we can use a triangular patch possessing three boundaries of different colors, as shown in Fig. 1(a) for the code distance of $d = 7$ with several examples of its checks and logical-Z operator. Here, we say that a boundary has a specific

color (e.g., red) if and only if it is adjacent to only faces of other two colors (e.g., green and blue). In addition, an edge connecting a boundary and a face is regarded to have the same color as them. The logical Pauli-X (Z) operator, denoted as $\overline{X}$ ($\overline{Z}$), is defined as the product of X (Z) operators on qubits placed along one of the three boundaries. The weight of each of these logical Pauli operators is the code distance of the code.

2.2 Decoding problem

For detecting and correcting errors, the checks of Eq. (1) are measured, repeatedly. These measurements consist of multiple *rounds*, each round being a subroutine to measure all the commuting checks in parallel without duplication (see Sec. 4.1 for the circuit implementing this). Each measurement outcome of a check is referred to as a *check outcome* and, if it is -1, we say that the check is *violated*. The collection of check outcomes of a single round is called a *syndrome*.

Decoding is the process of predicting errors from syndromes of a single or multiple rounds. If the residual errors after decoding incurs a logical error, we say that the decoding fails. Ideal decoding using maximum likelihood is generally known to be #P-complete [36], thus alternative decoders that are less precise but more efficient are necessary for practical quantum computing. For Calderbank-Shor-Steane (CSS) codes where X-type and Z-type checks are separately defined, these two types of checks can be decoded independently by regarding Y errors as combinations of X and Z errors.

Assuming that syndrome measurements are perfect, meaning each check outcome is obtained without error, the minimum-weight perfect matching (MWPM) algorithm [37] can be used directly for decoding Pauli errors on data qubits when each of elementary Pauli errors (which typically consist of the Pauli-X and Z operators of all data qubits) anticommutes with at most two checks of the code. This is the situation for surface codes [3]. In such a case, we consider a *matching graph* $G_{\mathrm{mat}} = (V, E)$ whose vertices V consist of checks and an additional 'boundary vertex' v_{bdry}. For each elementary error P, two checks anticommuting with P (or v_{bdry} and a single check anticommuting with P) are connected within G_{mat}. A weight is assigned to each edge either uniformly or as $\log[(1-q)/q]$, where q is the probability of the corresponding error. Denoting the set of violated checks as σ, the MWPM algorithm identifies a minimal-total-weight set of edges, denoted as $\mathrm{MWPM}(\sigma; G_{\mathrm{mat}}, v_{\mathrm{bdry}})$, that meet each violated check an odd number of times and each unviolated check an even number of times. In other words, $E_{\mathrm{MWPM}} \coloneqq \mathrm{MWPM}(\sigma; G_{\mathrm{mat}}, v_{\mathrm{bdry}}) \subset E$ satisfies

$$\begin{aligned} &\forall v \in \sigma,\ |\{e \in E_{\mathrm{MWPM}} \mid v \in e\}| \equiv 1 \pmod 2, \\ &\forall v \in V \setminus (\sigma \cup \{v_{\mathrm{bdry}}\}), \\ &\qquad |\{e \in E_{\mathrm{MWPM}} \mid v \in e\}| \equiv 0 \pmod 2 \end{aligned}$$

and have the smallest sum of weights. The final prediction is then the set of elementary errors corresponding to the edges in E_{MWPM}.

However, the above method is not applicable to color codes since each of the Pauli-X and Z operators of their physical qubits can anticommute with three checks. To obviate this problem, we can instead consider three *restricted lattices* visualized in Fig. 1(b), which are derived from the color-code lattice $\mathcal{L}_{\mathrm{2D}}$ as follows: The red-restricted lattice $\mathcal{L}^*_{\neg \mathtt{r}}$ contains the green and blue faces and an additional 'boundary vertex' v_{bdry} as its vertices and, for each red edge $e^{(\mathtt{r})}$ of $\mathcal{L}_{\mathrm{2D}}$, the green and blue faces divided by $e^{(\mathtt{r})}$ (or v_{bdry} and f if $e^{(\mathtt{r})}$ belongs to only one face f) are connected within $\mathcal{L}^*_{\neg \mathtt{r}}$ by an edge.[1] The green- and blue-restricted lattices $\mathcal{L}^*_{\neg \mathtt{g}}$, $\mathcal{L}^*_{\neg \mathtt{b}}$ are defined analogously. Importantly, any single-qubit X and Z errors respectively affect checks on at most two vertices in each restricted lattice. Therefore, we can apply MWPM on (all or some of) these three restricted lattices individually and combine these outcomes through a specific method, which is the basic idea of the projection decoder [18]. Alternatively, MWPM can be performed on a manifold built by connecting the three restricted lattices appropriately as in the Möbius MWPM decoder [23]. In this work, we take another approach to perform MWPM twice in a concatenated way, where the first one is applied on a restricted lat-

[1] In the literature, the restricted lattice is more often defined to have two boundary vertices (connected with each other), which respectively correspond to two among the three boundaries except the boundary of the restricted color. Although it may be mathematically more natural, it is not necessary for MWPM since the edge between the two boundary vertices is regarded to have zero weight. See Appendix A.3 for more details.

tice but the second one is applied on another useful lattice derived from $\mathcal{L}_{2\mathrm{D}}$ in a different way.

We note that we have assumed perfect syndrome extraction in the above discussions. If syndrome extraction is noisy, we can generally employ detector error models [38], which will be elaborated on in Sec. 4.

2.3 Noise models and settings

We consider one of the two types of noise models: bit-flip and circuit-level noise models. In the bit-flip noise model of strength p, every data qubit undergoes an X error with probability p at the start of each round. There are no errors in other steps including initialization and measurement. A more sophisticated noise model is the circuit-level noise model of strength p, defined as follows:

- Every measurement outcome is flipped with probability p.
- Every preparation of a qubit produces an orthogonal state with probability p.
- Every single- or two-qubit unitary gate (including the idle gate I) is followed by a single- or two-qubit depolarizing noise channel of strength p. We here regard that, for every time slice of the circuit, idle gates I are acted on all the qubits that are not involved in any non-trivial unitary gates or measurements.

Here, the single- and two-qubit depolarizing channels of strength p are respectively defined as

$$\mathcal{E}_p^{(1)}:\ \rho^{(1)}\mapsto(1-p)\rho^{(1)}+\frac{p}{3}\sum_{P\in\{X,Y,Z\}}P\rho^{(1)}P,$$
$$\mathcal{E}_p^{(2)}:\ \rho^{(2)}\mapsto(1-p)\rho^{(2)}+\frac{p}{15}\times\sum_{\substack{P_1,P_2\in\{I,X,Y,Z\}\\P_1\otimes P_2\neq I\otimes I}}(P_1\otimes P_2)\rho^{(2)}(P_1\otimes P_2),$$

where $\rho^{(1)}$ and $\rho^{(2)}$ are arbitrary single- and two-qubit density matrices, respectively.

To evaluate the performance of our decoder, we simulate T rounds of the logical idling gate acting on the triangular color code of code distance d, which is preceded by logical initialization (to the $+1$ eigenstate $|\overline{0}\rangle$ of $\overline{Z}$) and followed by a measurement in the $\overline{Z}$ basis. The logical initialization and measurement are done by initializing all the data qubits to $|0\rangle$ and measuring them in the Z basis, respectively. From this scenario, we can identify whether the $\overline{Z}$ observable fails (i.e., a $\overline{X}$ or $\overline{Y}$ error occurs) during the idling gate. While it is sufficient for bit-flip noise models, the $\overline{X}$ observable also can fail under circuit-level noise models. To cover this, we simulate again by swapping the X- and Z-type parts from the CNOT schedule (see Sec. 4.3 for more details). We estimate the logical failure rate p_{fail} as $p_{\mathrm{fail}}=p_{\mathrm{fail}}^{(X)}+p_{\mathrm{fail}}^{(Z)}$, where $p_{\mathrm{fail}}^{(X)}$ and $p_{\mathrm{fail}}^{(Z)}$ are respectively the failure rates of the $\overline{X}$ and $\overline{Z}$ observables. Since a $\overline{Y}$ error causes both observables to fail, this provides a conservative estimate that upper-bounds the actual failure rate.

2.4 Assessment of decoder performance

We assess the performance of the decoder from two perspectives: (i) what the noise threshold $p^{\times}$ of the decoder is and (ii) how the logical failure rate behaves when the noise strength p is sufficiently lower than the threshold (namely, $p\lessapprox p^{\times}/2$).

For a given value of T, the (cross) noise threshold $p^{\times}(T)$ is defined by the noise strength at which the curves of the logical failure rates p_{fail} for different code distances (which are sufficiently large) cross each other. We also define the *long-term cross threshold* as $p^{\times,\mathrm{LT}}:=\lim_{T\to\infty}p^{\times}(T)$ and predict it by fitting data into the ansatz

$$p^{\times}(T)=p^{\times,\mathrm{LT}}\left[1-\left(1-\frac{p^{\times}(1)}{p^{\times,\mathrm{LT}}}\right)T^{-\gamma}\right],\qquad(2)$$

following the method used in Ref. [20].

To investigate the sub-threshold scaling, we fit data (with sufficiently low p) into the ansatz[2]

$$\frac{p_{\mathrm{fail}}}{T}=\alpha\left(\frac{p}{p^{\star}}\right)^{\beta(d-d_0)+\eta}\qquad(3)$$

[2]This ansatz, which is widely employed in the literature [23, 39–43], can be justified as follows: For sufficiently low p, p_{fail} can be approximated by leaving only its lowest-order term with respect to p (with no constant term, since $p_{\mathrm{fail}}=0$ when $p=0$). Furthermore, below the threshold, p_{fail} is expected to decrease exponentially with d, as proven for toric and surface codes [4, 44–46] and supported numerically for color codes [20, 23]. Hence, for sufficiently low p, it is natural to consider an ansatz where $\log p_{\mathrm{fail}}$ is bilinear in $\log p$ and d, which leads to the form of Eq. (3).

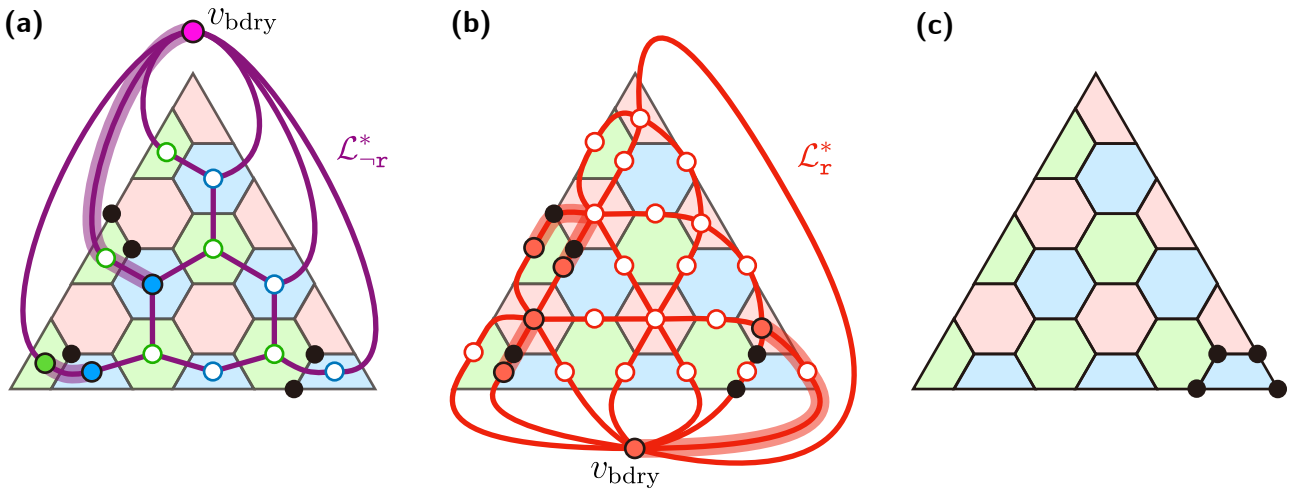

Figure 2: **Example of executing the concatenated MWPM decoder on the triangular color code of distance 7. (a)** First-round MWPM is performed on the red-restricted lattice $\mathcal{L}^*_{\neg \mathtt{r}}$, where the vertices are the green and blue faces of the original lattice $\mathcal{L}_{\mathrm{2D}}$ (marked as green and blue empty/filled circles) and a boundary vertex v_{bdry} (marked as a purple filled circle) connected by purple solid lines. The green/blue faces with violated checks are highlighted as green/blue filled circles. The obtained matching is drawn as purple thick lines, which correspond to a set of red edges $E^{(\mathtt{r})}_{\mathrm{pred}}$. **(b)** Second-round MWPM is performed on the red-only lattice $\mathcal{L}^*_{\mathtt{r}}$, where the vertices are the red faces and edges of $\mathcal{L}_{\mathrm{2D}}$ and a boundary vertex v_{bdry} (marked as red empty/filled circles) connected by red lines. The red edges in $E^{(\mathtt{r})}_{\mathrm{pred}}$ and the red faces with violated checks are highlighted as red filled circles. The obtained matching is drawn as red thick lines, which corresponds to a set of vertices $V^{(\mathtt{r})}_{\mathrm{pred}}$ predicted to have errors. **(c)** Residual errors after correcting $V^{(\mathtt{r})}_{\mathrm{pred}}$ are marked as black dots, which is equivalent to a stabilizer. The above process is repeated two more times while varying colors and the smallest one among $V^{(\mathtt{r})}_{\mathrm{pred}}$, $V^{(\mathtt{g})}_{\mathrm{pred}}$, and $V^{(\mathtt{b})}_{\mathrm{pred}}$ is selected as the outcome.

with five parameters $p^\star$, α, β, η and d_0, where d_0 is a number determined appropriately to minimize the uncertainties of α and η. Equation (3) can be equivalently written as

$$\log\left(\frac{p_{\mathrm{fail}}}{T}\right) = G(d)\log p + C(d), \tag{4}$$

where

$$G(d) = \beta(d-d_0)+\eta,$$
$$C(d) = -(\beta \log p^\star)(d-d_0) + (\log\alpha - \eta\log p^\star).$$

Hence, we can determine these parameters by two steps of linear regressions: first computing $G(d)$ and $C(d)$ by fitting $\log(p_{\mathrm{fail}}/T)$ against $\log p$ for each d and then fitting $G(d)$ and $C(d)$ against d, respectively. We select d_0 such that the uncertainties of the constant terms (i.e., η and $\log\alpha - \eta\log p^\star$) of $G(d)$ and $C(d)$ are minimized. (If such values of d_0 differ for $G(d)$ and $C(d)$, select their average.)

Since p_{fail}/T is expected to be constant when $p = p^\star$, we regard $p^\star$ as a 'noise threshold' as well. To distinguish the two types of thresholds $p^\times$ and $p^\star$, we refer to them as the *cross threshold* and *scaling threshold*, respectively. Note that $p^\times$ and $p^\star$ are generally not equal, as the ansatz in Eq. (3) is an asymptotical expression valid for sufficiently small p, where the sub-leading order terms with respect to p are negligible. Additionally, we specify the noise model (bit-flip or circuit-level) on the subscripts of the thresholds such as $p^\times_{\mathrm{bitflip}}$, $p^\times_{\mathrm{circuit}}$, $p^\star_{\mathrm{bitflip}}$, and $p^\star_{\mathrm{circuit}}$.

3 Concatenated MWPM decoder without faulty measurements

In this section, we describe the 2D variant of the concatenated MWPM decoder that is applicable only when every syndrome measurement is perfect. Let us consider the 2D color code on a lattice $\mathcal{L}_{\mathrm{2D}}$, which may have boundaries. The decoder to predict X errors in a single round can be briefly depicted as follows (see Fig. 2 for an example):

1. **(First-round MWPM)** Input violated blue and green Z-type checks to the MWPM algorithm on the red-restricted lattice $\mathcal{L}^*_{\neg \mathtt{r}}$, which returns a set of edges of $\mathcal{L}^*_{\neg \mathtt{r}}$. This

set corresponds to a set $E^{(\mathtt{r})}_{\mathrm{pred}}$ of red edges of $\mathcal{L}_{\mathrm{2D}}$, each of which is predicted to contain one X error. See Fig. 2(a).

2. **(Second-round MWPM)** Input $E^{(\mathtt{r})}_{\mathrm{pred}}$ and violated red Z-type checks to the MWPM algorithm on the 'red-only lattice' $\mathcal{L}^*_{\mathtt{r}}$, which is constructed according to the connection structure of red edges and faces in $\mathcal{L}_{\mathrm{2D}}$. The algorithm returns a set of edges of $\mathcal{L}^*_{\mathtt{r}}$, which correspond to a set of vertices $V^{(\mathtt{r})}_{\mathrm{pred}}$ of $\mathcal{L}_{\mathrm{2D}}$ that are predicted to have errors. See Fig. 2(b).

3. Repeat the above two steps (together referred to as the *red sub-decoding procedure*) while varying the color to green and blue, obtaining $V^{(\mathtt{g})}_{\mathrm{pred}}$ and $V^{(\mathtt{b})}_{\mathrm{pred}}$. Select the smallest one V_{pred} among $V^{(\mathtt{r})}_{\mathrm{pred}}$, $V^{(\mathtt{g})}_{\mathrm{pred}}$, and $V^{(\mathtt{b})}_{\mathrm{pred}}$ as the final outcome.

Figure 2(c) presents a set of residual errors after correction, which is equivalent to a stabilizer thus does not cause a logical failure. Z errors can be predicted by decoding X-type check outcomes in an analogous way.

To formally describe the above procedure, let us first define some notations. For a lattice $\mathcal{L}$, the sets of its vertices, edges, and faces are respectively denoted as $\Delta_0(\mathcal{L})$, $\Delta_1(\mathcal{L})$, and $\Delta_2(\mathcal{L})$. For each color $\mathtt{c} \in \{\mathtt{r},\mathtt{g},\mathtt{b}\}$, we denote the sets of $\mathtt{c}$-colored edges and faces in $\mathcal{L}_{\mathrm{2D}}$ as $\Delta^{(\mathtt{c})}_1(\mathcal{L}_{\mathrm{2D}})$ and $\Delta^{(\mathtt{c})}_2(\mathcal{L}_{\mathrm{2D}})$. The set of faces with violated Z-type checks is denoted as $\sigma_Z \in \Delta_2(\mathcal{L}_{\mathrm{2D}})$. We partition σ_Z into $\sigma_Z = \sigma^{(\mathtt{r})}_Z \cup \sigma^{(\mathtt{g})}_Z \cup \sigma^{(\mathtt{b})}_Z$ such that $\sigma^{(\mathtt{c})}_Z \subseteq \Delta^{(\mathtt{c})}_2(\mathcal{L}_{\mathrm{2D}})$ for each $\mathtt{c} \in \{\mathtt{r},\mathtt{g},\mathtt{b}\}$. The restricted and monochromatic lattices are then formally defined as follows:

Definition 1 (**Restricted lattices**)**.** The *red-restricted lattice* $\mathcal{L}^*_{\neg\mathtt{r}}$ is defined to have vertices of $\Delta_0(\mathcal{L}^*_{\neg\mathtt{r}}) = \Delta^{(\mathtt{g})}_2(\mathcal{L}_{\mathrm{2D}}) \cup \Delta^{(\mathtt{b})}_2(\mathcal{L}_{\mathrm{2D}}) \cup \{v_{\mathrm{bdry}}\}$, where v_{bdry} is an additional boundary vertex. For each red edge $e \in \Delta^{(\mathtt{r})}_1(\mathcal{L}_{\mathrm{2D}})$, we create an edge denoted as $\epsilon_{\neg\mathtt{r}}(e)$ within $\mathcal{L}^*_{\neg\mathtt{r}}$ as follows: If e belongs to two faces $f_{\mathtt{g}} \in \Delta^{(\mathtt{g})}_2(\mathcal{L}_{\mathrm{2D}})$ and $f_{\mathtt{b}} \in \Delta^{(\mathtt{b})}_2(\mathcal{L}_{\mathrm{2D}})$, $\epsilon_{\neg\mathtt{r}}(e)$ connects $f_{\mathtt{g}}$ and $f_{\mathtt{b}}$. If e belongs to only one face $f \in \Delta^{(\mathtt{g})}_2(\mathcal{L}_{\mathrm{2D}}) \cup \Delta^{(\mathtt{b})}_2(\mathcal{L}_{\mathrm{2D}})$, $\epsilon_{\neg\mathtt{r}}(e)$ connects f and v_{bdry}. Note that $\epsilon_{\neg\mathtt{r}}$ is a bijection between $\Delta^{(\mathtt{r})}_1(\mathcal{L}_{\mathrm{2D}})$ and $\Delta_1(\mathcal{L}^*_{\neg\mathtt{r}})$. For $E \subseteq \Delta^{(\mathtt{r})}_1(\mathcal{L}_{\mathrm{2D}})$, we denote $\epsilon_{\neg\mathtt{r}}(E) \coloneqq \{\epsilon_{\neg\mathtt{r}}(e) \mid e \in E\}$. The green- and blue-restricted lattices $\mathcal{L}^*_{\neg\mathtt{g}}$, $\mathcal{L}^*_{\neg\mathtt{b}}$ (with bijections $\epsilon_{\neg\mathtt{g}}$ and $\epsilon_{\neg\mathtt{b}}$) are defined similarly.

Definition 2 (**Monochromatic lattices**)**.** The *red-only lattice* $\mathcal{L}^*_{\mathtt{r}}$ is defined to have vertices of $\Delta_0(\mathcal{L}^*_{\mathtt{r}}) = \Delta^{(\mathtt{r})}_1(\mathcal{L}_{\mathrm{2D}}) \cup \Delta^{(\mathtt{r})}_2(\mathcal{L}_{\mathrm{2D}}) \cup \{v_{\mathrm{bdry}}\}$, where v_{bdry} is an additional boundary vertex. For each vertex $v \in \Delta_0(\mathcal{L}_{\mathrm{2D}})$, we connect an edge denoted as $\epsilon_{\mathtt{r}}(v)$ within $\mathcal{L}^*_{\mathtt{r}}$ as follows: If v is commonly included in a red edge $e \in \Delta^{(\mathtt{r})}_1(\mathcal{L}_{\mathrm{2D}})$ and a red face $f \in \Delta^{(\mathtt{r})}_2(\mathcal{L}_{\mathrm{2D}})$, $\epsilon_{\mathtt{r}}(v)$ connects e and f. If v is only included either in a red face f or in a red edge e, $\epsilon_{\mathtt{r}}(v)$ connects f or e with v_{bdry}. Note that $\epsilon_{\mathtt{r}}$ is a bijection between $\Delta_0(\mathcal{L}_{\mathrm{2D}})$ and $\Delta_1(\mathcal{L}^*_{\mathtt{r}})$. For $V \subseteq \Delta_0(\mathcal{L}_{\mathrm{2D}})$, we denote $\epsilon_{\mathtt{r}}(V) \coloneqq \{\epsilon_{\mathtt{r}}(v) \mid v \in V\}$. The green- and blue-only lattices $\mathcal{L}^*_{\mathtt{g}}$, $\mathcal{L}^*_{\mathtt{b}}$ (with bijections $\epsilon_{\mathtt{g}}$ and $\epsilon_{\mathtt{b}}$) are defined similarly.

Using the above definitions, we formally describe the decoding process as follows:

1. For each $\mathtt{c} \in \{\mathtt{r},\mathtt{g},\mathtt{b}\}$, execute the $\mathtt{c}$-colored sub-decoding procedure as follows:
   (a) Perform the MWPM algorithm to identify a set $\widetilde{E}^{(\mathtt{c})}_{\mathrm{pred}} \coloneqq \mathrm{MWPM}\left(\sigma^{(\mathtt{c}_1)}_Z \cup \sigma^{(\mathtt{c}_2)}_Z; \mathcal{L}^*_{\neg\mathtt{c}}, v_{\mathrm{bdry}}\right) \subseteq \Delta_1(\mathcal{L}^*_{\neg\mathtt{c}})$, where $\mathtt{c}_1$ and $\mathtt{c}_2$ are the other two colors besides $\mathtt{c}$. Define $E^{(\mathtt{c})}_{\mathrm{pred}} \coloneqq \epsilon^{-1}_{\neg\mathtt{c}}\left(\widetilde{E}^{(\mathtt{c})}_{\mathrm{pred}}\right) \in \Delta^{(\mathtt{c})}_1(\mathcal{L}_{\mathrm{2D}})$.
   (b) Perform the MWPM algorithm to identify a set $\widetilde{V}^{(\mathtt{c})}_{\mathrm{pred}} \coloneqq \mathrm{MWPM}\left(\sigma^{(\mathtt{c})}_Z \cup E^{(\mathtt{c})}_{\mathrm{pred}}; \mathcal{L}^*_{\mathtt{c}}, v_{\mathrm{bdry}}\right) \subseteq \Delta_1(\mathcal{L}^*_{\mathtt{c}})$ and define $V^{(\mathtt{c})}_{\mathrm{pred}} \coloneqq \epsilon^{-1}_{\mathtt{c}}\left(\widetilde{V}^{(\mathtt{c})}_{\mathrm{pred}}\right) \in \Delta_0(\mathcal{L}_{\mathrm{2D}})$
2. Return the smallest one V_{pred} among $V^{(\mathtt{r})}_{\mathrm{pred}}$, $V^{(\mathtt{g})}_{\mathrm{pred}}$, and $V^{(\mathtt{b})}_{\mathrm{pred}}$.

In Appendix A, we formally prove that the above process always returns a valid error prediction consistent with the syndrome. Additionally, we confirm that any outcome obtained from the projection decoder is a valid matching for the concatenated MWPM decoder, which implies that the former cannot outperform the latter.

In Fig. 14 of Appendix D, we numerically verify that our color-selecting strategy of executing

the sub-decoding procedures for all the three colors and selecting the best one is indeed necessary to maximize the performance of the decoder. If only one or two colors are considered in this process, the failure rate of the decoder significantly increases.

3.1 Performance analysis

To assess the performance of the decoder, we consider the setting in Sec. 2.3 under the bit-flip noise model of strength p. Since the bit-flip noise is independently applied to each round, the cross threshold is invariant under T, thus we simply denote $p^{\times}_{\mathrm{bitflip}} \coloneqq p^{\times}_{\mathrm{bitflip}}(T) = p^{\times,\mathrm{LT}}_{\mathrm{bitflip}}$ and consider only the case of $T = 1$. We employ the *PyMatching* library [47] to run the MWPM algorithm.

In Fig. 3(a) and (b), we present the logical failure probabilities p_{fail} computed for various code distances d when p is near the threshold and when p is sufficiently lower than the threshold, respectively. From Fig. 3(a), we can clearly observe that the cross threshold is $p^{\times}_{\mathrm{bitflip}} \approx 8.2\%$. In Fig. 3(b), the data points are linearly fitted for each d in the logarithmic scale, where the slopes $G(d)$ and constant terms $C(d)$ are plotted over d in Fig. 3(c). The regression parameters of $G(d)$ and $C(d)$ are estimated as

$$G(d) = (0.488 \pm 0.006)(d - 17) + (8.53 \pm 0.05),$$
$$C(d) = (1.30 \pm 0.02)(d - 17) + (20.6 \pm 0.2),$$

by selecting $d_0 = 17$, where the error terms are the 99% confidence intervals (CIs) estimated based on Students' t-distribution by using the python module `statsmodels` [48]. The corresponding ansatz parameters in Eq. (3) are estimated as

$$p^{\star}_{\mathrm{bitflip}} \approx 0.069, \quad \alpha \approx 0.12, \quad \beta \approx 0.49, \quad \eta \approx 8.5, \quad d_0 = 17. \tag{5}$$

Although the obtained cross threshold $p^{\times}_{\mathrm{bitflip}} = 8.2\%$ is lower than 8.7% of the projection decoder [18] and 9.0% of Möbius decoder [23], the concatenated MWPM decoder has a significant advantage in terms of the scaling of the failure rate over d and p. Namely, $\beta \approx 1/2$ for our decoder within our simulation range of $d \leq 31$, while $\beta \approx 1/3$ for the projection decoder [20] and $\beta \approx 3/7$ for the Möbius decoder [23]. The impact of this improvement is numerically presented in Fig. 4, where logical failure rates p_{fail} are estimated using the ansatz

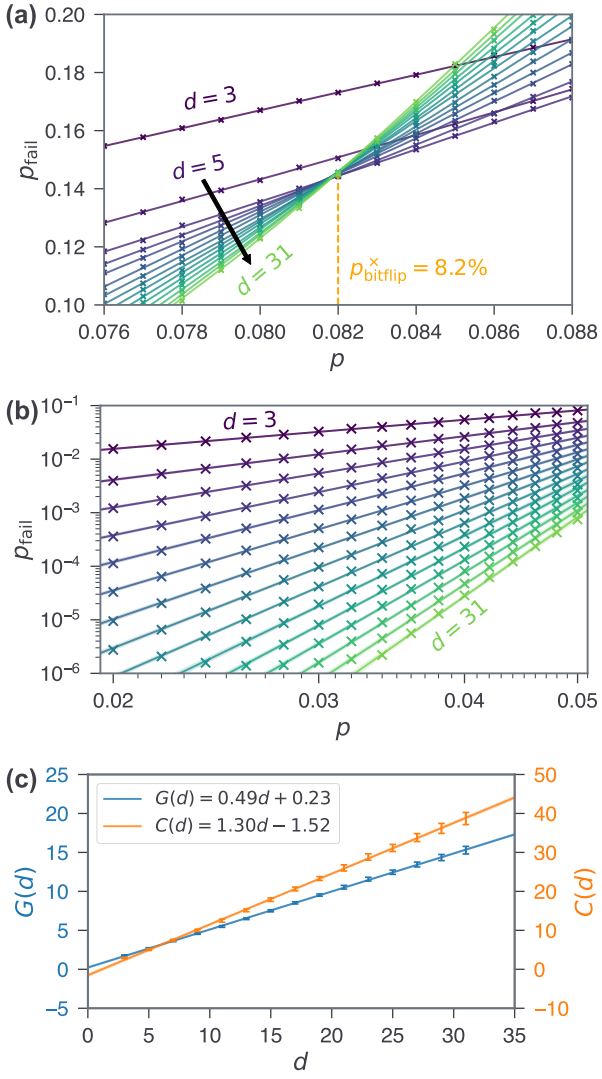

Figure 3: **Numerical analysis of the concatenated MWPM decoder under bit-flip noise.** We consider a single round of QEC on a triangular patch with code distance d under the bit-flip noise model of strength p. The logical failure rates p_{fail} are plotted over p **(a)** near the threshold and **(b)** in a sub-threshold region for various code distances ($d = 3, 5, 7, \cdots, 31$). In **(a)**, the solid lines are the LOWESS regressions with fraction $2/3$. The obtained cross threshold is $p^{\times}_{\mathrm{bitflip}} = 8.2\%$. In **(b)**, the solid lines are the linear regressions in the logarithmic scale, each of which is in the form of Eq. (4). The 99% confidence intervals (CIs) of the regression estimates are depicted as shaded regions around the solid lines. The number of samples for obtaining each data point is selected so that the 99% CI of p_{fail} is $\pm 10^{-3}$ for **(a)** and $\pm 0.05 p_{\mathrm{fail}}$ for **(b)**. In **(c)**, the 99% CIs of $G(d)$ and $C(d)$ in Eq. (4) are plotted over d with their linear regressions. From the regression parameters, we obtain the parameters of the ansatz of Eq. (3) as presented in Eq. (5).

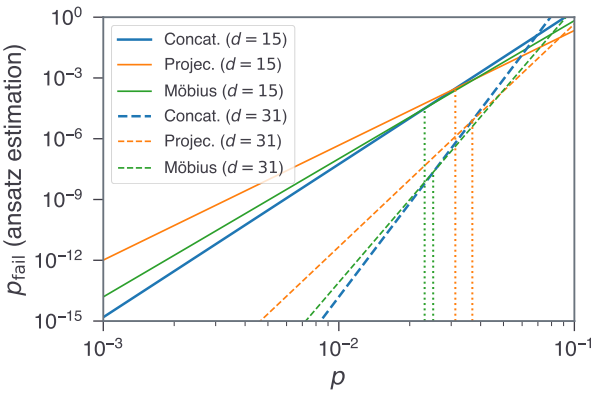

Figure 4: **Comparison of three matching-based decoders under bit-flip noise.** Logical failure rates p_{fail} estimated from the ansatz of Eq. (3) are plotted over p for two code distances $d \in \{15, 31\}$ and three decoders: the concatenated (Concat.), projection (Proj.) [18, 20], and Möbius MWPM decoders [23]. The parameter values used for the estimation are those in Eq. (5) for the concatenated MWPM decoder, $(p^{\star}_{\text{bitflip}}, \alpha, \beta, \eta, d_0) = (0.087, 0.1, 1/3, 2/3, 0)$ for the projection decoder (estimated optimistically by assuming $p^{\star}_{\text{bitflip}} = p^{\times}_{\text{bitflip}}$), and $(p^{\star}_{\text{bitflip}}, \alpha, \beta, \eta, d_0) = (0.0801, 0.148, 0.422, 0.488, 0)$ for the Möbius decoder (reported in Ref. [23]). The intersections of the curves of the concatenated MWPM decoder and the other two decoders are 3.12% ($d = 15$) and 3.69% ($d = 31$) for the projection decoder and 2.31% ($d = 15$) and 2.52% ($d = 31$) for the Möbius decoder.

of Eq. (3) for these three decoders and two code distances $d \in \{15, 31\}$. The values of the parameters $(p^{\star}_{\text{bitflip}}, \alpha, \beta, \eta, d_0)$ that we use for the estimation are $(0.087, 0.1, 1/3, 2/3, 0)$ for the projection decoder (which are optimistically guessed by assuming $p^{\star}_{\text{bitflip}} = p^{\times}_{\text{bitflip}}$) and $(0.0801, 0.148, 0.422, 0.488, 0)$ for the Möbius decoder (which are reported in Ref. [23]). We observe that the concatenated MWPM decoder outperforms the other two decoders when $p \lessapprox 2\% \approx p^{\times}_{\text{bitflip}}/4$.

We note that the value of β is closely related to the least-weight uncorrectable errors; namely, a decoder may be able to correct any error of weight smaller than $O(\beta d)$ for a color code with code distance d. The projection decoder cannot correct specific types of errors with weights $O(d/3)$ [20, 23], which are correctable via the concatenated MWPM decoder as shown in Appendix C.1. Likewise, we can find weight-$O(3d/7)$ uncorrectable errors for the Möbius decoder [23].

One may be led to guess that the concatenated MWPM decoder can correct any error of weight smaller than $O(d/2)$. Surprisingly, however, there exist uncorrectable errors with weights $O(3d/7)$ just as with the Möbius decoder, as illustrated in Appendix C.2. This raises the question of how the numerical analysis appears to scale as $\beta \approx 1/2$. We speculate that small-weight uncorrectable errors with weights $< O(d/2)$ may exist only when d is sufficiently large. As evidence, weight-$O(3d/7)$ uncorrectable errors of the type described in Appendix C.2 exist only when $d \geq 25$. Hence, we conjecture that β may approach a limit of $3/7$ for sufficiently large values of d, although its confirmation would demand extensive computational resources. Nonetheless, the concatenated MWPM decoder is still beneficial compared to the Möbius decoder, where β is computed to be about $3/7$ even for small values of $d < 25$. Note that Ref. [23] also suggests a modification of the Möbius decoder that manually compares inequivalent recovery operators, which is proved to correct every error with weight less than $d/2$ when $d \leq 13$. This improvement has not been considered in Fig. 4.

An additional numerical analysis for bit-flip noise is presented in Fig. 14 of Appendix D, showing that our strategy of comparing the outcomes from the sub-decoding procedures of all the three colors is indeed effective compared to performing only one or two sub-decoding procedures.

4 Generalized concatenated MWPM decoder for circuit-level noise

The bit-flip noise model considered in the previous section is adequate for assessing basic performance features of decoders, but it does not represent realistic noise that is relevant to fault-tolerant quantum computing. In practice, syndrome measurements are not perfect. We thus need to modify our decoder appropriately to accommodate circuit-level noise. We adapt our concatenated MWPM decoder to a circuit-level noise model defined in Sec. 2.3 by using the concept of *detector error model* (DEM) [38], which is a list of independent error mechanisms. We generalize the three steps of the concatenated MWPM decoder in Sec. 3 using DEMs and employ the *Stim* library [38] to implement and analyze the decoder.

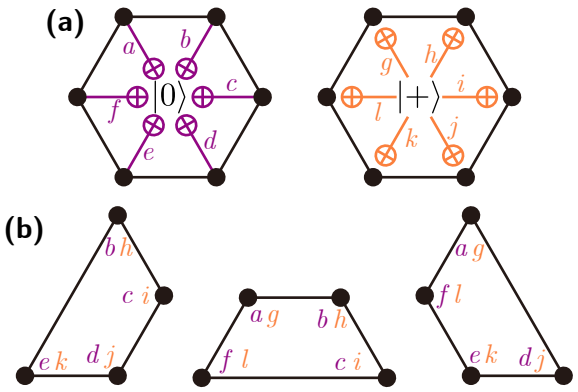

Figure 5: **Syndrome extraction circuit for color codes. (a)** Circuits for measuring checks on hexagonal faces, which are composed of multiple CNOT gates. The left (right) circuit followed by a Z (X) measurement on the center ancillary qubit measures the Z-type (X-type) checks. Twelve variables $a, b, \cdots, l$ are positive integers specifying the time slices that the corresponding CNOT gates are applied. **(b)** Faces located along the boundaries and corresponding time-slice variables of the CNOT gates, colored in purple (orange) for the measurements of Z-type (X-type) checks. The syndrome extraction circuits are in the same form as **(a)**, and are therefore omitted from the figure for simplicity.

4.1 Circuit and detectors

We first need to clarify the circuit implementing a color code, especially its syndrome extraction. We suppose that each face of the lattice contains two ancillary qubits (referred to as Z-type and X-type ancillary qubits), which are respectively used to extract the measurement outcomes of the Z-type and X-type checks on the face. We consider the scenario of T rounds of the logical idling gate with logical initialization and measurement, as described in Sec. 2.3.

In each round of syndrome extraction, Z-type (X-type) ancillary qubits are first initialized to $|0\rangle$ ($|+\rangle$), followed by CNOT gates between the ancillary and data qubits, concluding with the Z-basis (X-basis) measurement of the ancilla. The arrangement of these CNOT gates are presented in Fig. 5(a), where the left (right) circuit followed by a Z (X) measurement on the center Z-type (X-type) ancillary qubit measures the Z-type (X-type) check on the face. We have a degree of freedom on choosing the CNOT schedule (i.e., the order of applying these CNOT gates), which will be discussed later in Sec. 4.3.

We group the operations in the circuit in a way that each group (called a *time slice*) is composed of consecutive operations applied on distinct qubits that can be performed simultaneously. As stated in Sec. 2.3, we regard that, in each time slice, idle gates I are applied on all the qubits that are not involved in any non-trivial operations.

Let $m_{f,Z}^{(t)}, m_{f,X}^{(t)} \in \{\pm 1\}$ denote the measurement outcomes of the Z-type and X-type ancillary qubits, respectively, of a face f in the t-th round, where $1 \le t \le T$. We also define $\widetilde{m}_v \in \{\pm 1\}$ as the final measurement outcome of the data qubit located at a vertex v. For every face f and every pair of integers $t \in \{1, 2, \cdots, T+1\}$ and $s \in \{2, 3, \cdots, T\}$, the values

$$D_{f,Z}^{(t)} := \begin{cases} m_{f,Z}^{(1)} & \text{if } t = 1, \\ m_{f,Z}^{(t-1)} m_{f,Z}^{(t)} & \text{if } 2 \le t \le T, \\ m_{f,Z}^{(T)} \prod_{v \in f} \widetilde{m}_v & \text{if } t = T+1, \end{cases} \tag{6a}$$

$$D_{f,X}^{(s)} := m_{f,X}^{(s-1)} m_{f,X}^{(s)}, \tag{6b}$$

must be $+1$ when there are no errors. We refer to these values as *detectors* and classify them by their Pauli types and the colors of the faces; for example, $D_{f,Z}^{(t)}$ is a red Z-type detector if f is a red face. If a detector is -1, we say it is violated.

Lastly, the final measurement of the logical-Z operator always has an outcome of $+1$ when there are no errors. Thus, if $V_{\mathrm{bdry}}^{(\mathrm{r})}$ is defined by the set of the vertices placed along the red boundary, the value

$$L_Z = \prod_{v \in V_{\mathrm{bdry}}^{(\mathrm{r})}} \widetilde{m}_v \tag{7}$$

must be $+1$ when there are no errors. We refer to L_Z as the *logical observable* of our scenario. After decoding, a correction $L_Z^{(\mathrm{corr})}$ is obtained and the decoding succeeds if and only if $L_Z = L_Z^{(\mathrm{corr})}$. We describe L_Z as 'Z-type' because it involves only Z-basis measurements and shares common factors ($\{\widetilde{m}_v\}$) exclusively with Z-type detectors; see Eqs. (6) and (7). Note that, for an alternative scenario where the logical qubit is initialized and measured in the $\overline{X}$ basis, we have one X-type logical observable instead.

4.2 Generalization of the concatenated MWPM decoder

A *detector error model* (DEM) is defined by a set of independent error mechanisms. Each error

mechanism, which is formally described as a 3-tuple $(q, \mathcal{D}, \mathcal{O})$, specifies the probability (q) that the error occurs and the set of detectors $(\mathcal{D})$ and logical observables $(\mathcal{O})$ flipped by the error. The elements of $\mathcal{D} \cup \mathcal{O}$ are referred to as the *targets* of the error mechanism. We say that an error mechanism $(q, \mathcal{D}, \mathcal{O})$ is *edge-like* if and only if $|\mathcal{D}| \leq 2$.

The *Stim* library [38] can be used to extract a DEM from a noisy Clifford circuit (which is composed of Clifford gates, Pauli initializations/measurements, and Pauli noise channels) provided that detectors and logical observables are annotated appropriately. It works as follows: If the circuit has a single-Pauli error channel $\mathcal{E}_q^{(P)}$, which applies a specific Pauli product operator P with probability q, we can get its effect by commuting P to the end of the circuit and checking the detectors and logical observables flipped by it. If the circuit has a depolarizing channel $\mathcal{E}_p^{(1)}$ or $\mathcal{E}_p^{(2)}$, we can convert it into a sequence of single-Pauli error channels as

$$\mathcal{E}_p^{(1)} = \mathcal{E}_{q_1}^{(X)} \circ \mathcal{E}_{q_1}^{(Y)} \circ \mathcal{E}_{q_1}^{(Z)},$$
$$\mathcal{E}_p^{(2)} = \mathcal{E}_{q_2}^{(X\otimes I)} \circ \mathcal{E}_{q_2}^{(X\otimes X)} \circ \mathcal{E}_{q_2}^{(X\otimes Y)} \circ \cdots \circ \mathcal{E}_{q_2}^{(Z\otimes Z)},$$

where $q_1 := (1 - \sqrt{1 - 4p/3})/2$ and $q_2 := [1 - (1 - 16p/15)^{1/8}]/2$. We can then take account of these single-Pauli error channels individually. Note that such exact decomposition might be not possible for general Pauli noise channels, but it is not important in our discussions.

In the 2D variant of the decoder presented in Sec. 3, we consider two sub-lattices ($\mathtt{c}$-restricted lattice $\mathcal{L}^*_{\neg\mathtt{c}}$ and $\mathtt{c}$-only lattice $\mathcal{L}^*_{\mathtt{c}}$) for each sub-decoding procedure of color $\mathtt{c}$. Analogously, for each color $\mathtt{c}$, we deform and decompose the DEM $\mathcal{M}$ obtained from a color-code circuit into the *$\mathtt{c}$-restricted DEM* $\mathcal{M}_{\neg\mathtt{c}}$ and *$\mathtt{c}$-only DEM* $\mathcal{M}_{\mathtt{c}}$.

The step-by-step instructions to construct $\mathcal{M}_{\neg\mathtt{c}}$ and $\mathcal{M}_{\mathtt{c}}$ from $\mathcal{M}$ are presented in Algorithm 1. These can be outlined for $\mathtt{c} = \mathtt{r}$ as follows: First (in lines 1–10), we define $\mathcal{M}_{ZX}$ from $\mathcal{M}$ by separating each error mechanism that affects both Z- and X-type detectors into two independent mechanisms, ensuring every error mechanism affects only one detector type. If the logical observable L_Z is a target of an error mechanism to be separated, L_Z is included only in the Z-type part after the separation. Note that this process is equivalent to ignoring correlations between X and Z errors originated from Y errors or two-qubit errors such as $X \otimes Z$. We then compress $\mathcal{M}_{ZX}$; namely, we merge error mechanisms of $\mathcal{M}_{ZX}$ that have the same set of targets while updating the probabilities properly. After that (in lines 11–18), $\mathcal{M}_{\neg\mathtt{r}}$ is constructed by removing red detectors and logical observables from each error mechanism of $\mathcal{M}_{ZX}$ and compressing it. Importantly, we introduce a new *virtual detector* D_e for each error mechanism e in $\mathcal{M}_{\neg\mathtt{r}}$. Lastly (in lines 19–28), we build $\mathcal{M}_{\mathtt{r}}$ by replacing the green and blue detectors of each error mechanism of $\mathcal{M}_{ZX}$ with the corresponding virtual detector. As a consequence, only red detectors, virtual detectors, and logical observables are involved in $\mathcal{M}_{\mathtt{r}}$.

In addition to the above description, Algorithm 1 also contains processes to leave only edge-like mechanisms in the decomposed DEMs (see the conditions in lines 14, 22, and 24). By doing so, each of the DEMs can be expressed as a weighted graph G whose vertices comprise detectors and an additional boundary vertex v_{bdry}. Namely, each error mechanism $(q, \mathcal{D}, \mathcal{O})$ corresponds to an edge of G with weight $\log[(1-q)/q]$, which connects the two detectors in $\mathcal{D}$ (if $|\mathcal{D}| = 2$) or the only detector in $\mathcal{D}$ and v_{bdry} (if $|\mathcal{D}| = 1$). This graph can be used to perform the MWPM algorithm that predicts one of the most probable combinations of error mechanisms consistent with given violated detectors. If a logical observable is flipped by an odd number of these error mechanisms, its correction is -1; otherwise, it is $+1$.

With the above ingredients, we can finally describe how the concatenated MWPM decoder works. Denoting the set of violated $\mathtt{c}$-colored detectors as $\sigma^{(\mathtt{c})}$ for each $\mathtt{c} \in \{\mathtt{r}, \mathtt{g}, \mathtt{b}\}$, the decoder works as follows:

1. Obtain the red-restricted DEM $\mathcal{M}_{\neg\mathtt{r}}$, red-only DEM $\mathcal{M}_{\mathtt{r}}$, and set of virtual detectors $\{D_e\}_{e\in\mathcal{M}_{\neg\mathtt{r}}}$ from the DEM of the original circuit through Algorithm 1.

2. Perform the MWPM algorithm on $\mathcal{M}_{\neg\mathtt{r}}$ with the input $\sigma^{(\mathtt{g})} \cup \sigma^{(\mathtt{b})}$ and obtain a least-weight error set $E_{\mathtt{r}} \subseteq \mathcal{M}_{\neg\mathtt{r}}$.

3. Perform the MWPM algorithm on $\mathcal{M}_{\mathtt{r}}$ with the input $\sigma^{(\mathtt{r})} \cup \{D_e \mid e \in E_{\mathtt{r}}\}$ and obtain a correction $L_Z^{(\mathrm{corr},\mathtt{r})} \in \{\pm 1\}$ and the corresponding total weight $w_{\mathtt{r}}$ of predicted errors.

Algorithm 1: Decomposition of a color-code detector error model (DEM)

Input: A DEM $\mathcal{M} = \{(q_i, \mathcal{D}_i, \mathcal{O}_i)\}_i$
A color $\mathtt{c} \in \{\mathtt{r}, \mathtt{g}, \mathtt{b}\}$
Output: A $\mathtt{c}$-restricted DEM $\mathcal{M}_{\neg\mathtt{c}}$
A $\mathtt{c}$-only DEM $\mathcal{M}_{\mathtt{c}}$
A set of new 'virtual' detectors $\{D_e\}_{e\in\mathcal{M}_{\neg\mathtt{c}}}$

/* Separating Z- and X-type detectors */
$\mathcal{M}_{ZX} \leftarrow \emptyset$;
foreach $(q, \mathcal{D}, \mathcal{O}) \in \mathcal{M}$ **do**
  foreach $P \in \{Z, X\}$ **do**
    $\mathcal{D}_P \leftarrow \{P$-type detectors in $\mathcal{D}\}$;
    $\mathcal{O}_P \leftarrow \{P$-type logical observables in $\mathcal{D}\}$; // Our scenario has only one Z-type logical observable, but this is to ensure generalizability.
    if $\mathcal{D}_P \neq \emptyset$ **then**
      Add $(q, \mathcal{D}_P, \mathcal{O}_P)$ to $\mathcal{M}_{ZX}$;
  end
end
Compress $\mathcal{M}_{ZX}$ (i.e., successively merge a pair of error mechanisms of $\mathcal{M}_{ZX}$ that have the same targets while updating the probability as $q = q_1 + q_2 - 2q_1q_2$, where q_1 and q_2 are respectively the probabilities of the two error mechanisms);
/* Constructing $\mathcal{M}_{\neg\mathtt{c}}$ */
$\mathcal{M}_{\neg\mathtt{c}} \leftarrow \emptyset$;
foreach $(q, \mathcal{D}, \mathcal{O}) \in \mathcal{M}_{ZX}$ **do**
  $\mathcal{D}_{\neg\mathtt{c}} \leftarrow \{$Detectors in $\mathcal{D}$ that are not $\mathtt{c}$-colored$\}$;
  if $0 < |\mathcal{D}_{\neg\mathtt{c}}| \leq 2$ **then**
    Add $(q, \mathcal{D}_{\neg\mathtt{c}}, \emptyset)$ to $\mathcal{M}_{\neg\mathtt{c}}$;
end
Compress $\mathcal{M}_{\neg\mathtt{c}}$;
$\{D_e\}_{e\in\mathcal{M}_{\neg\mathtt{c}}} \leftarrow$ A set of new detectors;
/* Constructing $\mathcal{M}_{\mathtt{c}}$ */
$\mathcal{M}_{\mathtt{c}} \leftarrow \emptyset$;
foreach $(q, \mathcal{D}, \mathcal{O}) \in \mathcal{M}_{ZX}$ **do**
  $\mathcal{D}_{\neg\mathtt{c}} \leftarrow \{$Detectors in $\mathcal{D}$ that are not $\mathtt{c}$-colored$\}$;
  if $\mathcal{D}_{\neg\mathtt{c}} = \emptyset$ *and* $|\mathcal{D}| \leq 2$ **then**
    Add $(q, \mathcal{D}, \mathcal{O})$ to $\mathcal{M}_{\mathtt{c}}$;
  else if $0 < |\mathcal{D}_{\neg\mathtt{c}}| \leq 2$ *and* $|\mathcal{D} \setminus \mathcal{D}_{\neg\mathtt{c}}| \leq 1$ **then**
    $e \leftarrow$ An element of $\mathcal{M}_{\neg\mathtt{c}}$ with detectors $\mathcal{D}_{\neg\mathtt{c}}$ (which uniquely exists);
    Add $(q, \mathcal{D} \setminus \mathcal{D}_{\neg\mathtt{c}} \cup \{D_e\}, \mathcal{O})$ to $\mathcal{M}_{\mathtt{c}}$;
  end
end

4. Repeat the above steps for the other two colors and obtain $L_Z^{(\mathrm{corr},\mathtt{g})}$ and $L_Z^{(\mathrm{corr},\mathtt{b})}$ with the corresponding total weights $w_{\mathtt{g}}$ and $w_{\mathtt{b}}$.

5. Return $L_Z^{(\mathrm{corr},\mathtt{c})}$ where $\mathtt{c} := \mathrm{argmin}_{\mathtt{c}'\in\{\mathtt{r},\mathtt{g},\mathtt{b}\}} w_{\mathtt{c}'}$.

4.3 Optimization of the CNOT schedule

The time order of CNOT gates for syndrome extraction, called the CNOT schedule, needs to be optimized carefully before analyzing the performance of the decoder. We use a similar method as Ref. [20] to determine it. A brief review of this is as follows: The CNOT schedule can be specified by a tuple of twelve positive integers

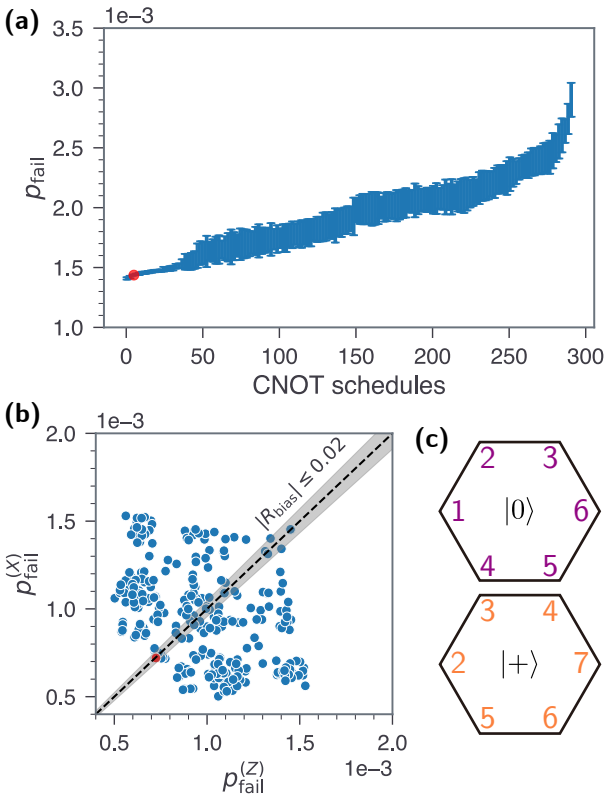

Figure 6: **Performance comparison of 292 length-7 CNOT schedules. (a)** The failure rate p_{fail} of $T = 7$ rounds of the logical idling gate when using the concatenated MWPM decoder is evaluated for code distance $d = 7$ under the circuit-level noise model of strength $p = 10^{-3}$ for each of these CNOT schedules (sorted by the values of p_{fail} in ascending order). The error bars correspond to the 99% CIs. The selected optimal schedule is $[2, 3, 6, 5, 4, 1; 3, 4, 7, 6, 5, 2]$, which is marked as a red dot. **(b)** The Z- and X-failure rates $p^{(Z)}_{\text{fail}}$, $p^{(X)}_{\text{fail}}$ for these CNOT schedules are visualized as a scatter plot. The dashed line plots $p^{(X)}_{\text{fail}} = p^{(Z)}_{\text{fail}}$ and the gray area indicates the region of $|R_{\text{bias}}| \leq 0.02$. The selected optimal schedule is marked as a red dot. **(c)** Selected optimal schedule following the same graphical representation as in Fig. 5(a).

$\mathcal{A} = [a, b, c, d, e, f; g, h, i, j, k, l]$, which contains all the integers from 1 to $\max \mathcal{A}$ (called the length of the schedule). Each integer indicates the time slice at which the corresponding CNOT gate presented in Fig. 5 is applied; that is, we first apply the CNOT gates with integer 1, then apply those with integer 2, and so on. Several conditions need to be imposed on $\mathcal{A}$ since each qubit can be involved in at most one CNOT gate per time slice and Z-type and X-type syndrome measurements should not interfere each other; see Sec. II C of Ref. [20] for their explicit descriptions. No CNOT schedules with length less than 7 satisfy these conditions. However, there are 876 valid length-7 schedules and we can leave only 292 among them by removing every schedule equivalent to another schedule up to a symmetry.[3]

We note that our simulating scenario described in Sec. 2.3 is only for computing the $\overline{Z}$-failure rate $p^{(Z)}_{\text{fail}}$ (i.e., failure rate of the Z observable). Nonetheless, we can utilize the fact that the $\overline{X}$-failure rate $p^{(X)}_{\text{fail}}$ is equal to the $\overline{Z}$-failure rate when using the CNOT schedule with the "Z-part" (i.e., first six integers) and "X-part" (i.e., last six integers) reversed from the original one. For example, the $\overline{X}$-failure rate for the schedule $[2, 3, 6, 5, 4, 1; 3, 4, 7, 6, 5, 2]$ can be computed from our scenario by instead using the schedule $[3, 4, 7, 6, 5, 2; 2, 3, 6, 5, 4, 1]$. For a given CNOT schedule, we compute the logical failure rate as $p_{\text{fail}} = p^{(X)}_{\text{fail}} + p^{(Z)}_{\text{fail}}$ and quantify the bias between them as

$$R_{\text{bias}} := \log_{10} \frac{p^{(Z)}_{\text{fail}}}{p^{(X)}_{\text{fail}}}, \tag{8}$$

which is zero when there is no bias.

To find the optimal CNOT schedule, we evaluate p_{fail} by using the concatenated MWPM decoder when $d = T = 7$ for the above 292 valid length-7 CNOT schedules under the circuit-level noise model of strength $p = 10^{-3}$. We select $p = 10^{-3}$ since we mainly concern the sub-threshold performance of our decoder when $p \leq 10^{-3}$. The evaluated values of p_{fail} are plotted in ascending order in Fig. 6(a). In addition, Fig. 6(b) shows the distribution of $p^{(Z)}_{\text{fail}}$ and $p^{(X)}_{\text{fail}}$ for these schedules. We select the optimal CNOT schedule as the one that gives the smallest p_{fail} among the schedules having $|R_{\text{bias}}| \leq 0.02 \approx \log_{10} 1.05$, which is visualized as a gray area in Fig. 6(b). The selected CNOT schedule is

$$\mathcal{A} = [2, 3, 6, 5, 4, 1; 3, 4, 7, 6, 5, 2], \tag{9}$$

which is marked as red dots in Figs. 6(a) and (b) and visualized in Fig. 6(c). The corresponding

[3] This number (292) of valid length-7 schedules is inconsistent with the number (234) reported in Ref. [20]. We discussed it with the first author of Ref. [20], but could not identify the precise reason of this discrepancy. We have verified using *Stim* that all of the 292 schedules that we identify give the same detectors as intended. We also note that, should we have mistakenly included additional redundant schedules, this would not affect finding an optimal schedule.

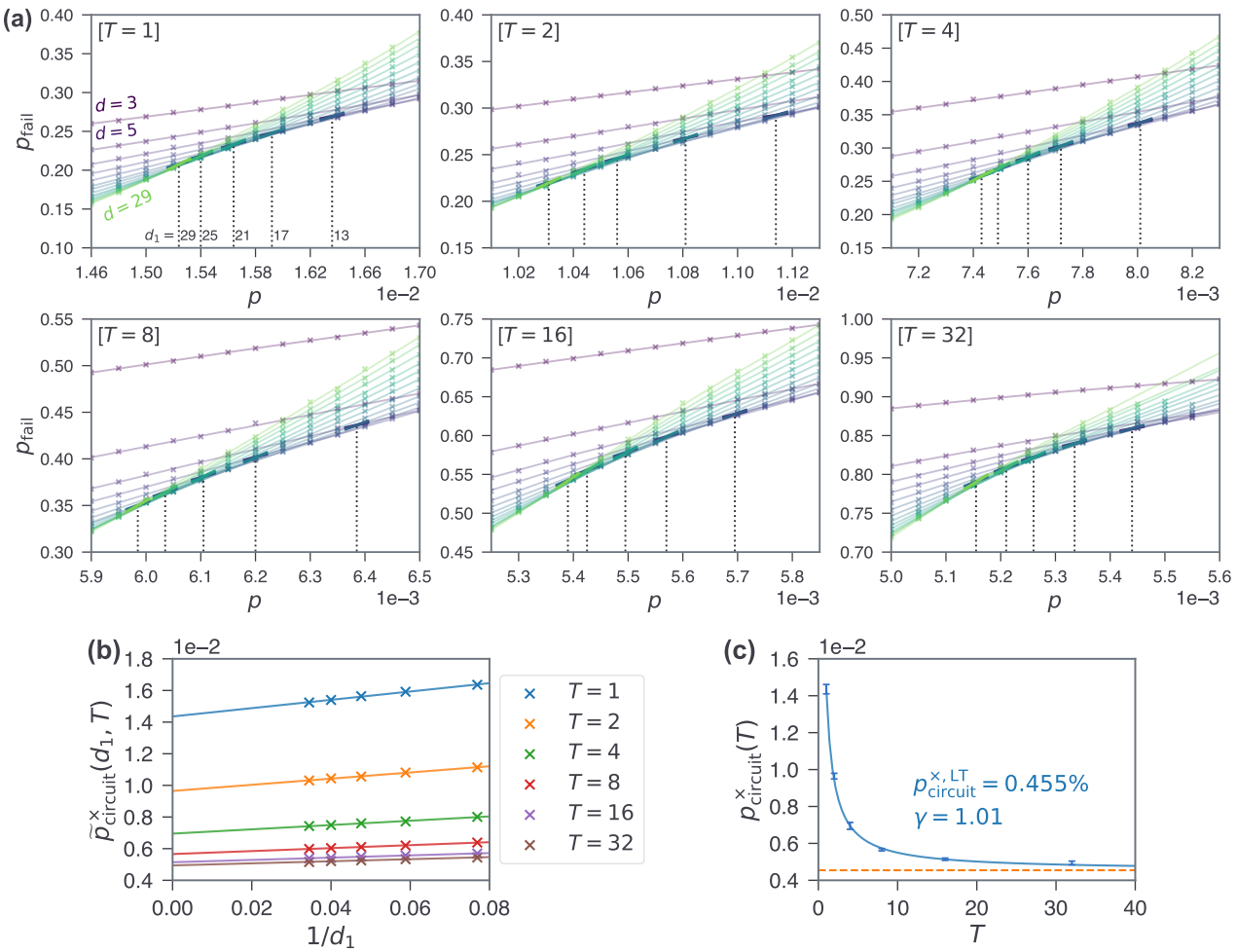

Figure 7: **Numerical analysis of the decoder under near-threshold circuit-level noise.** We consider $T \in \{1, 2, 4, 8, 16, 32\}$ rounds of the logical idling gate of the triangular color code with code distance $d \in \{3, 5, 7, \cdots, 29\}$ under the circuit-level noise model of strength p. **(a)** Logical failure rates $p_{\text{fail}} = p_{\text{fail}}^{(X)} + p_{\text{fail}}^{(Z)}$ are plotted over p for each T and d. Each data point (marked as 'X') has the 99% CI of $\pm 10^{-3}$. For each d, the LOWESS regression of p_{fail} with fraction $2/3$ is drawn as a solid line. For each $d_1 \in \{13, 17, 21, 25, 29\}$, the crossing of the two regression lines of $d = d_1$ and $d = (d_1 + 1)/2$ is highlighted in dark color. **(b)** The values of p at these crossings $\widetilde{p}^{\times}_{\text{circuit}}(d_1, T)$ are plotted over $1/d_1$ for each T with their linear regressions. The cross thresholds $p^{\times}_{\text{circuit}}(T)$ are estimated by the intersections of these regressions with the vertical axis. **(c)** The 99% CIs of the estimations of $p^{\times}_{\text{circuit}}(T)$ are presented with a fitted curve in the form of Eq. (2), where $p^{\times,\text{LT}}_{\text{circuit}} \approx 0.455\%$ and $\gamma \approx 1.01$. The orange dashed line plots $p^{\times}_{\text{circuit}}(T) = p^{\times,\text{LT}}_{\text{circuit}}$.

failure rates and bias (99% CI) are

$$\begin{aligned} p_{\text{fail}} &= (1.437 \pm 0.005) \times 10^{-3}, \\ p_{\text{fail}}^{(Z)} = p_{\text{fail}}^{(X)} &= (7.19 \pm 0.04) \times 10^{-4}, \\ R_{\text{bias}} &= 0.000 \pm 0.003. \end{aligned}$$

Note that the failure rate for the worst-performing CNOT schedule is $p_{\text{fail}} = (2.9 \pm 0.1) \times 10^{-3}$, which is approximately twice that of the best case. In Appendix B, we discuss the origin of the bias and explain why the degree of this bias varies with the CNOT schedule.

Although we here choose the schedule to have a sufficiently small bias, biased schedules may be intentionally used depending on the task. For instance, in the 15-to-1 magic state distillation scheme [49], $\overline{X}$ errors on ancillary logical qubits are more detrimental than $\overline{Z}$ errors, as the latter can be detected by the final $\overline{X}$ measurements [43]. Therefore, using CNOT schedules biased towards $\overline{Z}$ errors (i.e., $R_{\text{bias}} > 0$) for the ancillary logical qubits can be advantageous.

4.4 Performance analysis

We now analyze the performance of the decoder under circuit-level noise models when the CNOT schedule is set to the optimal one in Eq. (9). As in Sec. 3.1, we first examine the near-threshold region to determine the cross threshold and then investigate the sub-threshold scaling of the logical failure rate.

For near-threshold simulations, we consider $T \in \{1, 2, 4, 8, 16, 32\}$ rounds of the logical idling gate with code distance $d \in \{3, 5, 7, \cdots, 29\}$. The evaluated logical failure rates p_{fail} are plotted over the noise strength p in Fig. 7(a) for each T and d. Unlike the case of bit-flip noise plotted in Fig. 3(a), the regression lines for each value of T do not clearly intersect at a single point. Therefore, to determine the cross threshold $p^{\times}_{\text{circuit}}(T)$, we evaluate $\widetilde{p}^{\times}_{\text{circuit}}(d_1, T)$, which is the crossing of the two lines of $d = d_1$ and $d = (d_1 + 1)/2$ for $d_1 \equiv 1 \pmod 4$, while varying d_1 and fit the data into the ansatz

$$\widetilde{p}^{\times}_{\text{circuit}}(d_1, T) = \frac{A}{d_1} + p^{\times}_{\text{circuit}}(T),$$

where A is a free parameter. The values of $\widetilde{p}^{\times}_{\text{circuit}}(d_1, T)$ for $d_1 \in \{13, 17, 21, 25, 29\}$ are highlighted in dark color in Fig. 7(a) and plotted in Fig. 7(b) against $1/d_1$ with their linear regressions. The obtained cross thresholds $p^{\times}_{\text{circuit}}(T)$ are visualized in Fig. 7(c). By fitting them into Eq. (2), we get the long-term cross threshold $p^{\times,\text{LT}}_{\text{circuit}} \approx 0.455\%$ with $\gamma \approx 1.01$.

We next analyze the sub-threshold scaling of logical failure rates. We consider $T = 4d$ rounds of the logical idling gate with code distance $d \in \{3, 5, 7, \cdots, 21\}$ and compute its logical failure rate per round (p_{fail}/T) as presented in Fig. 8(a). For each d, the data with $p \leq 10^{-3}$ are fitted into the ansatz of Eq. (4), whose gradient $G(d)$ and constant $C(d)$ are plotted in Fig. 8(b). We obtain the regressions of $G(d)$ and $C(d)$ against d as

$$G(d) = (0.50 \pm 0.05)(d - 12) + (5.4 \pm 0.3),$$
$$C(d) = (2.9 \pm 0.4)(d - 12) + (26 \pm 2),$$

by selecting $d_0 = 12$, where the error terms are the 99% CIs of the estimations. The corresponding ansatz parameters in Eq. (3) are

$$p^{\star}_{\text{circuit}} \approx 3.2 \times 10^{-3}, \quad \alpha \approx 8.4 \times 10^{-3},$$
$$\beta \approx 0.50, \quad \eta \approx 5.4, \quad d_0 = 12.$$

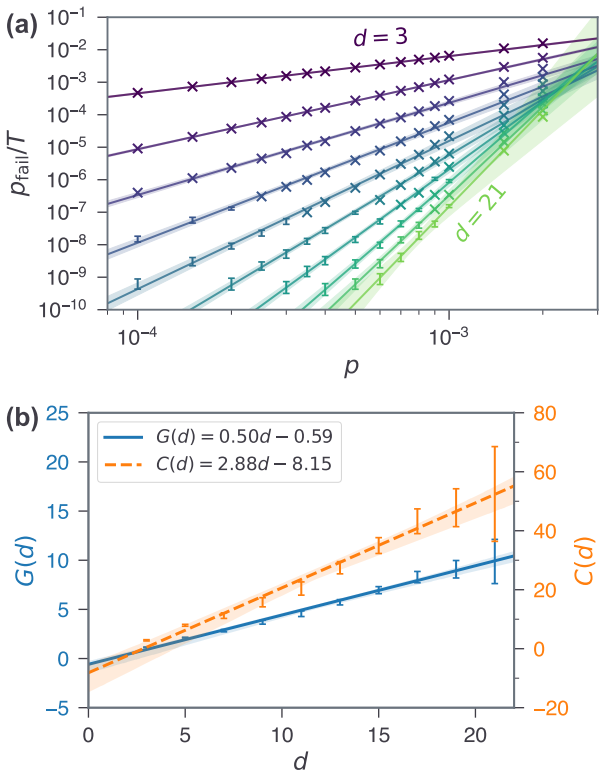

Figure 8: **Numerical analysis of the decoder under sub-threshold circuit-level noise.** We consider $T = 4d$ rounds of the logical idling gate of the triangular color code with code distance $d \in \{3, 5, \cdots, 21\}$ under the circuit-level noise model of strength p. **(a)** Logical failure rates per round (p_{fail}/T) are plotted over p for various code distances. Each data point is marked as an error bar indicating its 99% CI or as an 'X' symbol if its relative margin of error (i.e., the ratio of half the width of its CI to its center) is less than 10%. For each d, the data within the range of $p \leq 10^{-3}$ are used for the regression in the form of Eq. (4), which is drawn as a solid line. The 99% CIs of the regression estimates are depicted as shaded regions around the lines. **(b)** The 99% CIs of $G(d)$ and $C(d)$ in Eq. (4) are plotted over d. Their linear regressions are shown as solid/dashed lines with shaded regions indicating the 99% CIs of the estimates.

We lastly compare the performance of the concatenated MWPM decoder with that of previous decoders: the projection and Möbius decoders. In Fig. 9, we present the logical failure rates per round p_{fail}/T estimated by using these decoders across different code distances d at three noise strengths $p \in \{10^{-3}, 5 \times 10^{-4}, 10^{-4}\}$. The data for the projection and Möbius decoders are originated from Refs. [20] and [24], respectively. The figure shows that the projection decoder significantly underperforms the other two due to its

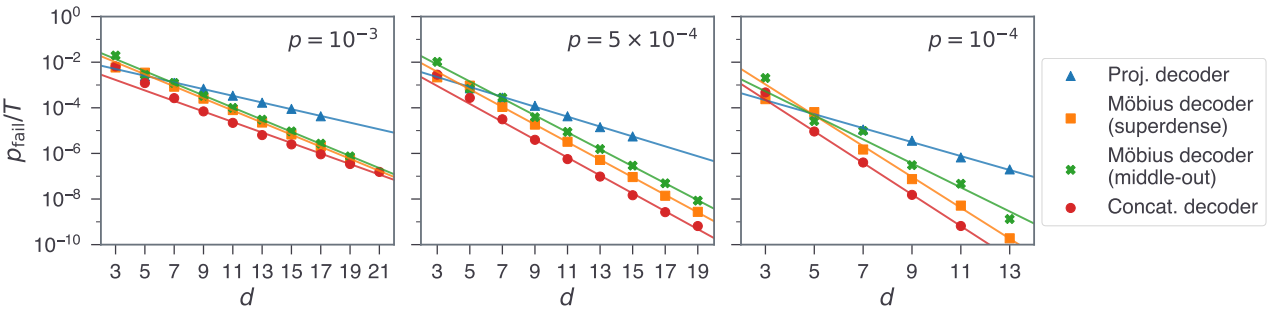

Figure 9: **Comparison of three matching-based decoders under circuit-level noise.** Logical failure rates per round p_{fail}/T are estimated by using three different decoders (projection, Möbius, and concatenated MWPM decoders) across different code distances d at varying noise strengths $p \in \{10^{-3}, 5 \times 10^{-4}, 10^{-4}\}$. The data for the projection decoder are from Fig. 17 of Ref. [20]. The data for the Möbius decoder are from Figs. 11 and 12 of Ref. [24], which respectively correspond to the superdense and middle-out color code circuits (two types of syndrome extraction circuits proposed in Ref. [24]). The solid lines represent the linear regressions of $\log(p_{\mathrm{fail}}/T)$ against d.

suboptimal scaling against d. The scaling against d is comparably similar for the other two decoders (when $p \lessapprox 5 \times 10^{-4}$); however, the concatenated MWPM decoder achieves logical failure rates that are approximately 3–7 times lower than the Möbius decoder.

In Figs. 15–17 of Appendix D, we present separate analyses of the $\overline{Z}$- and $\overline{X}$-failure rates of the decoder, together with analysis of the bias R_{bias} defined in Eq. (8), while setting the CNOT schedule to Eq. (9) as well. The results are summarized as follows: The long-term cross thresholds are estimated as 0.460% for the $\overline{Z}$-failure and 0.452% for the $\overline{X}$-failure, which differ by about 1.8% (0.008%p). However, this difference likely falls within the range of statistical error, as the 99% CIs for the estimations of R_{bias} almost always include zero, implying that the null hypothesis of $R_{\mathrm{bias}} = 0$ cannot be rejected at the 99% significance level for these cases.

5 Remarks

In this work, we introduced the concatenated minimum-weight perfect matching (MWPM) decoder processed by the concatenation of two rounds of MWPM per color, which is applicable not only to simple bit-flip noise but also to realistic circuit-level noise. The decoder is based on the idea that the outcome obtained from decoding on a restricted lattice can serve as additional 'virtual syndrome data', which undergo a subsequent decoding round in conjunction with remaining syndrome data.

We numerically analyzed the performance of the decoder in various aspects: We considered both bit-flip and circuit-level noise models and investigated near-threshold and sub-threshold behaviors of the logical failure rate p_{fail}. We found that the decoder has the thresholds of 8.2% for bit-flip noise and 0.46% for circuit-level noise, which are comparable with those of previous matching-based decoders such as the projection decoder [18–22] and Möbius MWPM decoder [23, 24]. Remarkably, we verified that the decoder approaches a scaling of $p_{\mathrm{fail}} \sim p^{d/2}$, where p is the noise strength and d is the code distance, at least within our simulation range ($d \lessapprox 31$ for bit-flip noise and $d \lessapprox 21$ for circuit-level noise). As a consequence, it outperforms previous matching-based decoders in terms of their sub-threshold failure rates across both bit-flip and circuit-level noise, as visualized in Figs. 4 and 9. We therefore anticipate that our decoder enhances the practicality of employing color codes in quantum computing, which has been considered less viable than surface codes due to its logical failure rate performance despite its advantage in resource efficiency [6]. We distributed a python module implementing the decoder on Github [34] so that other researchers can use it.

We consider several future directions related to this work. All the analyses in this work are based on the logical idling gate, which is generally suitable for initial studies of a decoder. However, to use it in real implementations, we should also consider nontrivial operations. In particular, it will be worth investigating how the decoder should be modified to handle domain walls required for lattice surgery [50]. Additionally,

syndrome extraction circuits may be able to be optimized further beyond the simple circuit in Fig. 5. For example, Ref. [24] suggests two circuits: superdense and middle-out circuits. Superdense circuits use two ancilla qubits per face, which are prepared in a Bell pair so that superdense coding can be employed. Middle-out circuits do not have ancilla qubits, thereby significantly reducing resource overheads at the cost of a slight decrease in fault tolerance. It will be interesting to see how the concatenated MWPM decoder performs with such circuits.

Another promising direction for future research is to apply our concatenation approach to the Möbius decoder [23, 24]. Specifically, instead of considering the three colors independently, we could run the first-round MWPM on the unified lattice, as done in the Möbius decoder. A key challenge lies in constructing the lattice for the second-round MWPM and lifting the first-round MWPM outcomes from the Möbius decoder to this lattice. We have two preliminary ideas for this: We could either (i) combine the three monochromatic lattices by introducing extra edges to account for errors at the boundaries between different colors, or (ii) keep each monochromatic lattice independent and lift each boundary error to two nodes belonging to distinct monochromatic lattices. Developing and testing these ideas further would be a worthwhile direction for future work.

Acknowledgements

We thank Felix Thomsen, Nicholas Fazio, Samuel C. Smith, Benjamin J. Brown, Michael E. Beverland, and Kaavya Sahay for helpful discussions and comments. This work is supported by the Australian Research Council via the Centre of Excellence in Engineered Quantum Systems (EQUS) project number CE170100009. This article is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001122C0063. Any opinions, findings and conclusions or recommendations expressed in this article are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Projects Agency (DARPA).

References

[1] H. Bombin and M. A. Martin-Delgado. "Topological quantum distillation". Phys. Rev. Lett. **97**, 180501 (2006).

[2] Héctor Bombín. "Topological codes". In Daniel A. Lidar and Todd A. Brun, editors, Quantum Error Correction. Chapter 19, pages 455–481. Cambridge (2013).

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

[4] Eric Dennis, Alexei Kitaev, Andrew Landahl, and John Preskill. "Topological quantum memory". J. Math. Phys. **43**, 4452–4505 (2002).

[5] Andrew J. Landahl, Jonas T. Anderson, and Patrick R. Rice. "Fault-tolerant quantum computing with color codes" (2011). arXiv:1108.5738.

[6] Felix Thomsen, Markus S. Kesselring, Stephen D. Bartlett, and Benjamin J. Brown. "Low-overhead quantum computing with the color code". Phys. Rev. Res. **6**, 043125 (2024).

[7] Christopher Chamberland and Kyungjoo Noh. "Very low overhead fault-tolerant magic state preparation using redundant ancilla encoding and flag qubits". npj Quantum Inf. **6**, 91 (2020).

[8] Tomohiro Itogawa, Yugo Takada, Yutaka Hirano, and Keisuke Fujii. "Even more efficient magic state distillation by zero-level distillation" (2024). arXiv:2403.03991.

[9] Craig Gidney, Noah Shutty, and Cody Jones. "Magic state cultivation: Growing T states as cheap as CNOT gates" (2024). arXiv:2409.17595.

[10] Yutaka Hirano, Tomohiro Itogawa, and Keisuke Fujii. "Leveraging zero-level distillation to generate high-fidelity magic states" (2024). arXiv:2404.09740.

[11] Seok-Hyung Lee, Felix Thomsen, Nicholas Fazio, Benjamin J. Brown, and Stephen D. Bartlett. "Low-overhead magic state distillation with color codes" (2024). arXiv:2409.07707.

[12] Lukas Postler, Sascha Heuβen, Ivan Pogorelov, Manuel Rispler, Thomas Feldker, Michael Meth, Christian D. Marciniak, Roman Stricker, Martin Ringbauer, Rainer Blatt, Philipp Schindler, Markus Müller,

and Thomas Monz. "Demonstration of fault-tolerant universal quantum gate operations". Nature **605**, 675–680 (2022).

[13] C. Ryan-Anderson, N. C. Brown, M. S. Allman, B. Arkin, G. Asa-Attuah, C. Baldwin, J. Berg, J. G. Bohnet, S. Braxton, N. Burdick, J. P. Campora, A. Chernoguzov, J. Esposito, B. Evans, D. Francois, J. P. Gaebler, T. M. Gatterman, J. Gerber, K. Gilmore, D. Gresh, A. Hall, A. Hankin, J. Hostetter, D. Lucchetti, K. Mayer, J. Myers, B. Neyenhuis, J. Santiago, J. Sedlacek, T. Skripka, A. Slattery, R. P. Stutz, J. Tait, R. Tobey, G. Vittorini, J. Walker, and D. Hayes. "Implementing fault-tolerant entangling gates on the five-qubit code and the color code" (2022). arXiv:2208.01863.

[14] Lukas Postler, Friederike Butt, Ivan Pogorelov, Christian D. Marciniak, Sascha Heußen, Rainer Blatt, Philipp Schindler, Manuel Rispler, Markus Müller, and Thomas Monz. "Demonstration of fault-tolerant Steane quantum error correction". PRX Quantum **5**, 030326 (2024).

[15] Dolev Bluvstein, Simon J. Evered, Alexandra A. Geim, Sophie H. Li, Hengyun Zhou, Tom Manovitz, Sepehr Ebadi, Madelyn Cain, Marcin Kalinowski, Dominik Hangleiter, J. Pablo Bonilla Ataides, Nishad Maskara, Iris Cong, Xun Gao, Pedro Sales Rodriguez, Thomas Karolyshyn, Giulia Semeghini, Michael J. Gullans, Markus Greiner, Vladan Vuletić, and Mikhail D. Lukin. "Logical quantum processor based on reconfigurable atom arrays". Nature **626**, 58–65 (2024).

[16] Pedro Sales Rodriguez, John M. Robinson, Paul Niklas Jepsen, Zhiyang He, Casey Duckering, Chen Zhao, Kai-Hsin Wu, Joseph Campo, Kevin Bagnall, Minho Kwon, Thomas Karolyshyn, Phillip Weinberg, Madelyn Cain, Simon J. Evered, Alexandra A. Geim, Marcin Kalinowski, Sophie H. Li, Tom Manovitz, Jesse Amato-Grill, James I. Basham, Liane Bernstein, Boris Braverman, Alexei Bylinskii, Adam Choukri, Robert DeAngelo, Fang Fang, Connor Fieweger, Paige Frederick, David Haines, Majd Hamdan, Julian Hammett, Ning Hsu, Ming-Guang Hu, Florian Huber, Ningyuan Jia, Dhruv Kedar, Milan Kornjača, Fangli Liu, John Long, Jonathan Lopatin, Pedro L. S. Lopes, Xiu-Zhe Luo, Tommaso Macrì, Ognjen Marković, Luis A. Martínez-Martínez, Xianmei Meng, Stefan Ostermann, Evgeny Ostroumov, David Paquette, Zexuan Qiang, Vadim Shofman, Anshuman Singh, Manuj Singh, Nandan Sinha, Henry Thoreen, Noel Wan, Yiping Wang, Daniel Waxman-Lenz, Tak Wong, Jonathan Wurtz, Andrii Zhdanov, Laurent Zheng, Markus Greiner, Alexander Keesling, Nathan Gemelke, Vladan Vuletić, Takuya Kitagawa, Sheng-Tao Wang, Dolev Bluvstein, Mikhail D. Lukin, Alexander Lukin, Hengyun Zhou, and Sergio H. Cantú. "Experimental demonstration of logical magic state distillation" (2024). arXiv:2412.15165.

[17] Nathan Lacroix, Alexandre Bourassa, Francisco J. H. Heras, Lei M. Zhang, Johannes Bausch, Andrew W. Senior, Thomas Edlich, Noah Shutty, Volodymyr Sivak, Andreas Bengtsson, Matt McEwen, Oscar Higgott, Dvir Kafri, Jahan Claes, Alexis Morvan, Zijun Chen, Adam Zalcman, Sid Madhuk, Rajeev Acharya, Laleh Aghababaie Beni, Georg Aigeldinger, Ross Alcaraz, Trond I. Andersen, Markus Ansmann, Frank Arute, Kunal Arya, Abraham Asfaw, Juan Atalaya, Ryan Babbush, Brian Ballard, Joseph C. Bardin, Alexander Bilmes, Sam Blackwell, Jenna Bovaird, Dylan Bowers, Leon Brill, Michael Broughton, David A. Browne, Brett Buchea, Bob B. Buckley, Tim Burger, Brian Burkett, Nicholas Bushnell, Anthony Cabrera, Juan Campero, Hung-Shen Chang, Ben Chiaro, Liang-Ying Chih, Agnetta Y. Cleland, Josh Cogan, Roberto Collins, Paul Conner, William Courtney, Alexander L. Crook, Ben Curtin, Sayan Das, Sean Demura, Laura De Lorenzo, Agustin Di Paolo, Paul Donohoe, Ilya Drozdov, Andrew Dunsworth, Alec Eickbusch, Aviv Moshe Elbag, Mahmoud Elzouka, Catherine Erickson, Vinicius S. Ferreira, Leslie Flores Burgos, Ebrahim Forati, Austin G. Fowler, Brooks Foxen, Suhas Ganjam, Gonzalo Garcia, Robert Gasca, Élie Genois, William Giang, Dar Gilboa, Raja Gosula, Alejandro Grajales Dau, Dietrich Graumann, Alex Greene, Jonathan A.

Gross, Tan Ha, Steve Habegger, Monica Hansen, Matthew P. Harrigan, Sean D. Harrington, Stephen Heslin, Paula Heu, Reno Hiltermann, Jeremy Hilton, Sabrina Hong, Hsin-Yuan Huang, Ashley Huff, William J. Huggins, Evan Jeffrey, Zhang Jiang, Xiaoxuan Jin, Chaitali Joshi, Pavol Juhas, Andreas Kabel, Hui Kang, Amir H. Karamlou, Kostyantyn Kechedzhi, Trupti Khaire, Tanuj Khattar, Mostafa Khezri, Seon Kim, Paul V. Klimov, Bryce Kobrin, Alexander N. Korotkov, Fedor Kostritsa, John Mark Kreikebaum, Vladislav D. Kurilovich, David Landhuis, Tiano Lange-Dei, Brandon W. Langley, Pavel Laptev, Kim-Ming Lau, Justin Ledford, Kenny Lee, Brian J. Lester, Loïck Le Guevel, Wing Yan Li, Yin Li, Alexander T. Lill, William P. Livingston, Aditya Locharla, Erik Lucero, Daniel Lundahl, Aaron Lunt, Ashley Maloney, Salvatore Mandrà, Leigh S. Martin, Orion Martin, Cameron Maxfield, Jarrod R. McClean, Seneca Meeks, Anthony Megrant, Kevin C. Miao, Reza Molavi, Sebastian Molina, Shirin Montazeri, Ramis Movassagh, Charles Neill, Michael Newman, Anthony Nguyen, Murray Nguyen, Chia-Hung Ni, Murphy Y. Niu, Logan Oas, William D. Oliver, Raymond Orosco, Kristoffer Ottosson, Alex Pizzuto, Rebecca Potter, Orion Pritchard, Chris Quintana, Ganesh Ramachandran, Matthew J. Reagor, Rachel Resnick, David M. Rhodes, Gabrielle Roberts, Eliott Rosenberg, Emma Rosenfeld, Elizabeth Rossi, Pedram Roushan, Kannan Sankaragomathi, Henry F. Schurkus, Michael J. Shearn, Aaron Shorter, Vladimir Shvarts, Spencer Small, W. Clarke Smith, Sofia Springer, George Sterling, Jordan Suchard, Aaron Szasz, Alex Sztein, Douglas Thor, Eifu Tomita, Alfredo Torres, M. Mert Torunbalci, Abeer Vaishnav, Justin Vargas, Sergey Vdovichev, Guifre Vidal, Catherine Vollgraff Heidweiller, Steven Waltman, Jonathan Waltz, Shannon X. Wang, Brayden Ware, Travis Weidel, Theodore White, Kristi Wong, Bryan W. K. Woo, Maddy Woodson, Cheng Xing, Z. Jamie Yao, Ping Yeh, Bicheng Ying, Juhwan Yoo, Noureldin Yosri, Grayson Young, Yaxing Zhang, Ningfeng Zhu, Nicholas Zobrist, Hartmut Neven, Pushmeet Kohli, Alex Davies, Sergio Boixo, Julian Kelly, Cody Jones, Craig Gidney, and Kevin J. Satzinger. "Scaling and logic in the color code on a superconducting quantum processor" (2024). arXiv:2412.14256.

[18] Nicolas Delfosse. "Decoding color codes by projection onto surface codes". Phys. Rev. A **89**, 012317 (2014).

[19] Christopher Chamberland, Aleksander Kubica, Theodore J. Yoder, and Guanyu Zhu. "Triangular color codes on trivalent graphs with flag qubits". New J. Phys. **22**, 023019 (2020).

[20] 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).

[21] Aleksander Kubica and Nicolas Delfosse. "Efficient color code decoders in $d \geq 2$ dimensions from toric code decoders". Quantum **7**, 929 (2023).

[22] Jiaxuan Zhang, Yu-Chun Wu, and Guo-Ping Guo. "Facilitating practical fault-tolerant quantum computing based on color codes". Phys. Rev. Res. **6**, 033086 (2024).

[23] Kaavya Sahay and Benjamin J. Brown. "Decoder for the triangular color code by matching on a Möbius strip". PRX Quantum **3**, 010310 (2022).

[24] Craig Gidney and Cody Jones. "New circuits and an open source decoder for the color code" (2023). arXiv:2312.08813.

[25] Christopher T. Chubb. "General tensor network decoding of 2D Pauli codes" (2021). arXiv:2101.04125.

[26] Yugo Takada, Yusaku Takeuchi, and Keisuke Fujii. "Ising model formulation for highly accurate topological color codes decoding". Phys. Rev. Res. **6**, 013092 (2024).

[27] Lucas Berent, Lukas Burgholzer, Peter-Jan H.S. Derks, Jens Eisert, and Robert Wille. "Decoding quantum color codes with MaxSAT". Quantum **8**, 1506 (2024).

[28] Eric Sabo, Arun B. Aloshious, and Kenneth R. Brown. "Trellis decoding for qudit stabilizer codes and its application to qubit topological codes" (2022). arXiv:2106.08251.

[29] Nishad Maskara, Aleksander Kubica, and Tomas Jochym-O'Connor. "Advantages of versatile neural-network decoding for topological codes". Phys. Rev. A **99**, 052351 (2019).

[30] Nicolas Delfosse and Naomi H. Nickerson. "Almost-linear time decoding algorithm for topological codes". Quantum **5**, 595 (2021).

[31] Pradeep Sarvepalli and Robert Raussendorf. "Efficient decoding of topological color codes". Phys. Rev. A **85**, 022317 (2012).

[32] Christophe Piveteau, Christopher T. Chubb, and Joseph M. Renes. "Tensor-network decoding beyond 2D". PRX Quantum **5**, 040303 (2024).

[33] Seok-Hyung Lee and Hyunseok Jeong. "Universal hardware-efficient topological measurement-based quantum computation via color-code-based cluster states". Phys. Rev. Res. **4**, 013010 (2022).

[34] Seok-Hyung Lee. "color-code-stim". `https://github.com/seokhyung-lee/color-code-stim` (2024).

[35] Austin G. Fowler. "Two-dimensional color-code quantum computation". Phys. Rev. A **83**, 1–8 (2011).

[36] Pavithran Iyer and David Poulin. "Hardness of decoding quantum stabilizer codes". IEEE Trans. Inf. Theory **61**, 5209–5223 (2015).

[37] Jack Edmonds. "Paths, trees, and flowers". Can. J. of Math. **17**, 449–467 (1965).

[38] Craig Gidney. "Stim: A fast stabilizer circuit simulator". Quantum **5**, 497 (2021).

[39] Austin G. Fowler, Simon J. Devitt, and Cody Jones. "Surface code implementation of block code state distillation". Scientific reports **3**, 1939 (2013).

[40] Austin G. Fowler. "Analytic asymptotic performance of topological codes". Phys. Rev. A **87**, 040301 (2013).

[41] Austin G. Fowler and Craig Gidney. "Low overhead quantum computation using lattice surgery" (2019). arxiv:1808.06709.

[42] Daniel Litinski. "A game of surface codes: Large-scale quantum computing with lattice surgery". Quantum **3**, 128 (2019).

[43] Daniel Litinski. "Magic state distillation: Not as costly as you think". Quantum **3**, 205 (2019-12-02). arxiv:1905.06903.

[44] R Raussendorf, J Harrington, and K Goyal. "Topological fault-tolerance in cluster state quantum computation". New J. Phys. **9**, 199 (2007).

[45] Austin G. Fowler. "Proof of finite surface code threshold for matching". Phys. Rev. Lett. **109**, 180502 (2012).

[46] Fern H. E. Watson and Sean D. Barrett. "Logical error rate scaling of the toric code". New J. Phys. **16**, 093045 (2014).

[47] Oscar Higgott. "PyMatching: A python package for decoding quantum codes with minimum-weight perfect matching". ACM Trans. Quantum Comput.**3** (2022).

[48] Skipper Seabold and Josef Perktold. "Statsmodels: Econometric and statistical modeling with python". In 9th Python in Science Conference. (2010).

[49] Sergey Bravyi and Alexei Kitaev. "Universal quantum computation with ideal clifford gates and noisy ancillas". Phys. Rev. A **71**, 022316 (2005).

[50] 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". PRX Quantum **5**, 010342 (2024).

A Proof of the validity of the concatenated MWPM decoder

In this appendix, we formally prove that the 2D version of the concatenated MWPM decoder described in Sec. 3 indeed works well. Namely, we show that the decoder always identifies a proper prediction of X errors consistent with the syndrome. We additionally validate that any outcome obtained from the projection decoder is a valid matching for the concatenated MWPM decoder, which implies that the former cannot outperform the latter. We here use the same mathematical notations as used in Sec. 3.

A.1 Notations and definitions

Let $\mathcal{L}$ be a trivalent three-colorable lattice. We assume that the lattice does not have boundaries; we will consider boundaries later. We consider the lattices: the dual lattice $\mathcal{L}^*$, the decoding hypergraph $\mathcal{H}$, the red/green/blue-restricted lattice $\mathcal{L}^*_{\neg \mathtt{r}/\neg \mathtt{g}/\neg \mathtt{b}}$, the red/green/blue-only lattice $\mathcal{L}^*_{\mathtt{r}/\mathtt{g}/\mathtt{b}}$.

Let $\mathcal{L}^*$ be the dual lattice of $\mathcal{L}$, whose vertices are three-colorable so that endpoints of an edge are distinctly colored and whose faces are of the form $\{\{v_\mathtt{r}, v_\mathtt{g}\}, \{v_\mathtt{g}, v_\mathtt{b}\}, \{v_\mathtt{b}, v_\mathtt{r}\}\}$, where $v_\mathtt{r}$, $v_\mathtt{g}$, and $v_\mathtt{b}$ are respectively red, green, and blue vertices. The color of an edge is chosen to be distinct from the colors of its endpoints. We associate with the dual lattice its cellular homology complex $(C_0(\mathcal{L}^*), C_1(\mathcal{L}^*), C_2(\mathcal{L}^*))$, where $C_0(\mathcal{L}^*)$ is an $\mathbb{F}_2$ (binary) vector space with the basis set $\Delta_0(\mathcal{L}^*)$, and accordingly for $C_1(\mathcal{L}^*)$, $\Delta_1(\mathcal{L}^*)$ and $C_2(\mathcal{L}^*)$, $\Delta_2(\mathcal{L}^*)$. We equip the complex with linear boundary maps $\partial_1^* : C_1(\mathcal{L}^*) \to C_0(\mathcal{L}^*)$ and $\partial_2^* : C_2(\mathcal{L}^*) \to C_1(\mathcal{L}^*)$ such that $\partial_1^*(\{u, v\}) = u + v$ and $\partial_2^*(\{e_1, e_2, ..., e_m\}) = e_1 + e_2 + ... + e_m$. For each color $\mathtt{c} \in \{\mathtt{r}, \mathtt{g}, \mathtt{b}\}$, we denote $\Delta_0^{(\mathtt{c})}(\mathcal{L}^*) = \{v \in \Delta_0(\mathcal{L}^*) \mid v \text{ is } \mathtt{c}\text{-colored}\}$ and accordingly for $\Delta_1^{(\mathtt{c})}(\mathcal{L}^*)$.

The decoding hypergraph $\mathcal{H}$ has vertices $\mathcal{V} = \Delta_0(\mathcal{L}^*)$, hyper-edges $\varepsilon = \{\cup_{e \in f} e \mid f \in \Delta_2(\mathcal{L}^*)\}$ (such that each hyper-edge is of the form $\{v_\mathtt{r}, v_\mathtt{g}, v_\mathtt{b}\}$), and hyper-faces $\mathcal{F} = \{\{e \in \Delta_1(\mathcal{L}^*) \mid v \in e\} \mid v \in \Delta_0(\mathcal{L}^*)\}$. On this lattice, we define a 2-complex with an $\mathbb{F}_2$ vector space $C_0(\mathcal{H})$ spanned by the $\mathcal{V}$, $C_1(\mathcal{H})$ spanned by ε, and $C_2(\mathcal{H})$ spanned by $\mathcal{F}$. The boundary maps on this complex are defined as

$$\partial_1^{\mathcal{H}}(\{v_\mathtt{r}, v_\mathtt{g}, v_\mathtt{b}\}) = v_\mathtt{r} + v_\mathtt{g} + v_\mathtt{b},$$
$$\partial_2^{\mathcal{H}}(\{e_1, e_2, \cdots, e_m\}) = e_1 + e_2 + \cdots + e_m.$$

The red-restricted lattice $\mathcal{L}^*_{\neg \mathtt{r}}$ is the subgraph of $\mathcal{L}^*$ over vertices $\Delta_0(\mathcal{L}^*_{\neg \mathtt{r}}) = \Delta_0^{(\mathtt{g})}(\mathcal{L}^*) \cup \Delta_0^{(\mathtt{b})}(\mathcal{L}^*)$. We associate it with its cellular homology complex $C_0(\mathcal{L}^*_{\neg \mathtt{r}})$. We define a linear projection operator $\pi_0^{(\mathtt{r})} : C_0(\mathcal{L}^*) \to C_0(\mathcal{L}^*)$ that satisfies

$$\pi_0^{(\mathtt{r})}(v) = \begin{cases} v & \text{if } v \notin \Delta_0^{(\mathtt{r})}(\mathcal{L}^*), \\ 0 & \text{otherwise} \end{cases}$$

for each $v \in \Delta_0(\mathcal{L}^*)$ such that $\mathrm{Im}(\pi_0^{(\mathtt{r})}) = C_0(\mathcal{L}^*_{\neg \mathtt{r}})$.

The red-only lattice $\mathcal{L}^*_\mathtt{r}$ is defined as

$$\begin{aligned}
\Delta_0(\mathcal{L}^*_\mathtt{r}) &= \Delta_0^{(\mathtt{r})}(\mathcal{L}^*) \cup \Delta_1^{(\mathtt{r})}(\mathcal{L}^*), \\
\Delta_1(\mathcal{L}^*_\mathtt{r}) &= \{\{v_\mathtt{r}, \{v_\mathtt{g}, v_\mathtt{b}\}\} \subset \Delta_0(\mathcal{L}^*_\mathtt{r}) \mid \ \{v_\mathtt{r}, v_\mathtt{g}, v_\mathtt{b}\} \in \varepsilon\}, \\
\Delta_2(\mathcal{L}^*_\mathtt{r}) &= \left\{ \{\{v_\mathtt{r}, \{v_\mathtt{g}, v_\mathtt{b}\}\} \in \Delta_1(\mathcal{L}^*_\mathtt{r}) \mid v \in \{v_\mathtt{r}, v_\mathtt{g}, v_\mathtt{b}\}\} \;\middle|\; v \in \Delta_0^{(\mathtt{g})}(\mathcal{L}^*) \cup \Delta_0^{(\mathtt{b})}(\mathcal{L}^*) \right\},
\end{aligned}$$

which respectively span $C_0(\mathcal{L}^*_\mathtt{r})$, $C_1(\mathcal{L}^*_\mathtt{r})$, and $C_2(\mathcal{L}^*_\mathtt{r})$ that form its cellular homology complex. Note that $C_1(\mathcal{L}^*_\mathtt{r})$ and $C_1(\mathcal{H})$ are isomorphic under the mapping of bases $\{v_\mathtt{r}, \{v_\mathtt{g}, v_\mathtt{b}\}\} \leftrightarrow \{v_\mathtt{r}, v_\mathtt{g}, v_\mathtt{b}\}$. Therefore, we will treat edges in the red-only lattice as hyper-edges when considering complexes. We also note that $C_0(\mathcal{L}^*_\mathtt{r}) = C_1(\mathcal{L}^*_{\neg \mathtt{r}}) \oplus \mathrm{Im}(\mathbb{1} - \pi_0^{(\mathtt{r})})$. We define $P_{C_1(\mathcal{L}^*_{\neg \mathtt{r}})}$ and $P_{\mathrm{Im}(\mathbb{1} - \pi_0^{(\mathtt{r})})}$ as the projectors from $C_0(\mathcal{L}^*_\mathtt{r})$ onto these two subspaces $C_1(\mathcal{L}^*_{\neg \mathtt{r}})$ and $\mathrm{Im}(\mathbb{1} - \pi_0^{(\mathtt{r})})$, respectively. The boundary map for 1-cells,

$$\partial_1^{\mathcal{L}^*_\mathtt{r}}(\{v_\mathtt{r}, v_\mathtt{g}, v_\mathtt{b}\}) = v_\mathtt{r} + \{v_\mathtt{g}, v_\mathtt{b}\},$$

can be separated into two functions $p_{\mathrm{vert}}^{(\mathtt{r})}$ and $p_{\mathrm{edge}}^{(\mathtt{r})}$ as

$$p_{\mathrm{vert}}^{(\mathtt{r})} = P_{\mathrm{Im}(\mathbb{1} - \pi_0^{(\mathtt{r})})} \partial_1^{\mathcal{L}^*_\mathtt{r}}, \tag{10a}$$
$$p_{\mathrm{edge}}^{(\mathtt{r})} = P_{C_1(\mathcal{L}^*_{\neg \mathtt{r}})} \partial_1^{\mathcal{L}^*_\mathtt{r}}, \tag{10b}$$
$$\partial_1^{\mathcal{L}^*_\mathtt{r}} = p_{\mathrm{vert}}^{(\mathtt{r})} + p_{\mathrm{edge}}^{(\mathtt{r})}.$$

These operators act on basis elements as

$$p^{(\mathtt{r})}_{\mathrm{vert}}(\{v_{\mathtt{r}}, v_{\mathtt{g}}, v_{\mathtt{b}}\}) = v_{\mathtt{r}},$$
$$p^{(\mathtt{r})}_{\mathrm{edge}}(\{v_{\mathtt{r}}, v_{\mathtt{g}}, v_{\mathtt{b}}\}) = \{v_{\mathtt{g}}, v_{\mathtt{b}}\}.$$

All the above notations are naturally extended to other two colors $\mathtt{g}$ and $\mathtt{b}$.

The decoding problem can be reformulated as follows: Given a set σ of syndrome vertices in $\mathcal{H}$, identify a set of hyper-edges whose boundary is equal to the syndrome vertices.

The concatenated decoder first finds a matching $b_{\mathtt{r}} \in C_1(\mathcal{L}^*_{\neg\mathtt{r}})$ in the red-restricted lattice such that $\partial_1^{\mathcal{L}^*_{\neg\mathtt{r}}}(b_{\mathtt{r}}) = \pi_0^{(\mathtt{r})}(\sigma)$. It then lifts this matching and the red vertices from the original syndrome into the red-only lattice to form a set of vertices $(\mathbb{1} - \pi_0^{(\mathtt{r})})(\sigma) + b_{\mathtt{r}}$, for which it then finds a matching x such that $\partial_1^{\mathcal{L}^*_{\mathtt{r}}}(x) = (\mathbb{1} - \pi_0^{(\mathtt{r})})(\sigma) + b_{\mathtt{r}}$. This is repeated for the other two colors and the least-weight correction is selected as the final outcome, but we here only consider the red sub-decoding procedure (without loss of generality) as our goal is showing the validity of the decoder, not its optimality.

A.2 Proof of validity when no boundaries

We first show that, for a given syndrome and error, if each of the first- and second-round decoding processes works correctly (i.e. returns a valid matching), the concatenated MWPM decoder returns a valid matching. Namely, we prove the following claim:

Claim 1. For $\sigma \in C_0(\mathcal{H})$, $x \in C_1(\mathcal{H})$, and $b_{\mathtt{r}} \in C_1(\mathcal{L}^*_{\neg\mathtt{r}})$, if

$$\partial_1^{\mathcal{L}^*_{\neg\mathtt{r}}}(b_{\mathtt{r}}) = \pi_0^{(\mathtt{r})}(\sigma), \tag{11a}$$
$$\partial_1^{\mathcal{L}^*_{\mathtt{r}}}(x) = \left(\mathbb{1} - \pi_0^{(\mathtt{r})}\right)(\sigma) + b_{\mathtt{r}}, \tag{11b}$$

then

$$\partial_1^{\mathcal{H}}(x) = \sigma. \tag{12}$$

Proof. Using the projected boundary maps in Eq. (10), we rewrite Eq. (11b) as

$$p^{(\mathtt{r})}_{\mathrm{vert}}(x) = \left(\mathbb{1} - \pi_0^{(\mathtt{r})}\right)(\sigma),$$
$$p^{(\mathtt{r})}_{\mathrm{edge}}(x) = b_{\mathtt{r}}.$$

From the latter equation we get $\partial_1^{\mathcal{L}^*_{\neg\mathtt{r}}} p^{(\mathtt{r})}_{\mathrm{edge}}(x) = \pi_0^{(\mathtt{r})}(\sigma)$, and thus $(\partial_1^{\mathcal{L}^*_{\neg\mathtt{r}}} p^{(\mathtt{r})}_{\mathrm{edge}} + p^{(\mathtt{r})}_{\mathrm{vert}})(x) = \sigma$. It suffices to prove that $(\partial_1^{\mathcal{L}^*_{\neg\mathtt{r}}} p^{(\mathtt{r})}_{\mathrm{edge}} + p^{(\mathtt{r})}_{\mathrm{vert}})(x) = \partial_1^{\mathcal{H}}(x)$. Since its both sides are linear in x, we only need to show this over the basis of hyper-edges that span $C_1(\mathcal{H})$:

$$\begin{aligned}\left(p^{(\mathtt{r})}_{\mathrm{vert}} + \partial_1^{\mathcal{L}^*_{\neg\mathtt{r}}} p^{(\mathtt{r})}_{\mathrm{edge}}\right)(\{v_{\mathtt{r}}, v_{\mathtt{g}}, v_{\mathtt{b}}\}) &= v_{\mathtt{r}} + \partial_1^{\mathcal{L}^*_{\neg\mathtt{r}}}(\{v_{\mathtt{g}}, v_{\mathtt{b}}\}) \\ &= v_{\mathtt{r}} + v_{\mathtt{g}} + v_{\mathtt{b}} \\ &= \partial_1^{\mathcal{H}}(\{v_{\mathtt{r}}, v_{\mathtt{g}}, v_{\mathtt{b}}\}).\end{aligned}$$

□

A.3 Consideration of boundaries

We now assume that $\mathcal{L}$ is a color-code lattice with boundaries. To handle them, its dual lattice $\mathcal{L}^*$ is augmented with three distinctly colored boundary vertices $v^{\mathtt{r}}_{\mathrm{bdry}}$, $v^{\mathtt{g}}_{\mathrm{bdry}}$, and $v^{\mathtt{b}}_{\mathrm{bdry}}$. Vertices in $\mathcal{L}^*$ corresponding to green and blue faces of $\mathcal{L}$ adjacent to the red boundary are connected to $v^{\mathtt{r}}_{\mathrm{bdry}}$, and so on for the other two colors, and all pairs of boundary vertices are connected.

Despite the existence of the boundary vertices, almost all the notations and definitions in Appendix A.1 do not change. Denoting the set of boundary vertices of a lattice $\mathcal{L}$ as $V_{\mathrm{bdry}}(\mathcal{L})$, the boundary vertices of various lattices described above are set as follows:

$$\begin{aligned} V_{\mathrm{bdry}}(\mathcal{L}^*) = V_{\mathrm{bdry}}(\mathcal{H}) &= \left\{v^{\mathtt{r}}_{\mathrm{bdry}}, v^{\mathtt{g}}_{\mathrm{bdry}}, v^{\mathtt{b}}_{\mathrm{bdry}}\right\}, \\ V_{\mathrm{bdry}}(\mathcal{L}^*_{\neg\mathtt{r}}) &= \left\{v^{\mathtt{g}}_{\mathrm{bdry}}, v^{\mathtt{b}}_{\mathrm{bdry}}\right\}, \\ V_{\mathrm{bdry}}(\mathcal{L}^*_{\mathtt{r}}) &= \left\{v^{\mathtt{r}}_{\mathrm{bdry}}, \left\{v^{\mathtt{g}}_{\mathrm{bdry}}, v^{\mathtt{b}}_{\mathrm{bdry}}\right\}\right\}, . \end{aligned}$$

We define $C_{\mathrm{bdry}}(\mathcal{L})$ as the $\mathbb{F}_2$ vector space spanned by $V_{\mathrm{bdry}}(\mathcal{L})$. Note that, from a decoding perspective, it does not matter if we contract multiple boundary vertices into a single boundary vertex since edges connecting them are given zero weight when applying MWPM; we thus displayed only one boundary vertex in Figs. 1 and 2 for simplicity. However, we here do not merge distinct boundary vertices for mathematical clarity.

Since syndromes can be matched with the boundary vertices, the mathematical description of MWPM in Eq. (11) and the validity condition of decoding in Eq. (12) should be modified. The modified claim and its proof are as follows:

Claim 2. For $\sigma \in C_0(\mathcal{H})$, $x \in C_1(\mathcal{H})$, and $b_{\mathtt{r}} \in C_1(\mathcal{L}^*_{\neg\mathtt{r}})$, if

$$\partial_1^{\mathcal{L}^*_{\neg\mathtt{r}}}(b_{\mathtt{r}}) - \pi_0^{(\mathtt{r})}(\sigma) \in C_{\mathrm{bdry}}(\mathcal{L}^*_{\neg\mathtt{r}}), \tag{13a}$$

$$\partial_1^{\mathcal{L}^*_{\mathtt{r}}}(x) - \left(\mathbb{1} - \pi_0^{(\mathtt{r})}\right)(\sigma) - b_{\mathtt{r}} \in C_{\mathrm{bdry}}(\mathcal{L}^*_{\mathtt{r}}), \tag{13b}$$

then $\partial_1^{\mathcal{H}}(x) - \sigma \in C_{\mathrm{bdry}}(\mathcal{H})$.

Proof. Let $c \in C_{\mathrm{bdry}}(\mathcal{L}^*_{\neg\mathtt{r}})$ and $c' = c'_{\mathrm{vert}} + c'_{\mathrm{edge}} \in C_{\mathrm{bdry}}(\mathcal{L}^*_{\mathtt{r}})$ denote the left-hand sides of Eqs. (13a) and (13b), respectively, where $c'_{\mathrm{vert}} = P_{\mathrm{Im}(\mathbb{1}-\pi_0^{(\mathtt{r})})}c' \in \left\{0, v^{\mathtt{r}}_{\mathrm{bdry}}\right\}$ and $c'_{\mathrm{edge}} = P_{C_1(\mathcal{L}^*_{\neg\mathtt{r}})}c' \in \left\{0, \left\{v^{\mathtt{g}}_{\mathrm{bdry}}, v^{\mathtt{b}}_{\mathrm{bdry}}\right\}\right\}$. We rewrite Eq. (13b) as

$$\begin{aligned} p^{(\mathtt{r})}_{\mathrm{vert}}(x) &= \left(\mathbb{1} - \pi_0^{(\mathtt{r})}\right)(\sigma) + c'_{\mathrm{vert}}, \\ p^{(\mathtt{r})}_{\mathrm{edge}}(x) &= b_{\mathtt{r}} + c'_{\mathrm{edge}}. \end{aligned}$$

From the latter equation we get $\partial_1^{\mathcal{L}^*_{\neg\mathtt{r}}} p^{(\mathtt{r})}_{\mathrm{edge}}(x) = \pi_0^{(\mathtt{r})}(\sigma) + c + \partial_1^{\mathcal{L}^*_{\neg\mathtt{r}}} c'_{\mathrm{edge}}$, and thus $(\partial_1^{\mathcal{L}^*_{\neg\mathtt{r}}} p^{(\mathtt{r})}_{\mathrm{edge}} + p^{(\mathtt{r})}_{\mathrm{vert}})(x) = \sigma + c + c'_{\mathrm{vert}} + \partial_1^{\mathcal{L}^*_{\neg\mathtt{r}}} c'_{\mathrm{edge}}$. Since $c + c'_{\mathrm{vert}} + \partial_1^{\mathcal{L}^*_{\neg\mathtt{r}}} c'_{\mathrm{edge}} \in C_{\mathrm{bdry}}(\mathcal{H})$ (as $\partial_1^{\mathcal{L}^*_{\neg\mathtt{r}}} c'_{\mathrm{edge}}$ is either 0 or $v^{\mathtt{g}}_{\mathrm{bdry}} + v^{\mathtt{b}}_{\mathrm{bdry}}$), it suffices to prove that

$$(\partial_1^{\mathcal{L}^*_{\neg\mathtt{r}}} p^{(\mathtt{r})}_{\mathrm{edge}} + p^{(\mathtt{r})}_{\mathrm{vert}} - \partial_1^{\mathcal{H}})(x) \in C_{\mathrm{bdry}}(\mathcal{H}).$$

Because of linearity, we only need to show this over the basis of hyper-edges that span $C_1(\mathcal{H})$:

$$\begin{aligned} &\left(p^{(\mathtt{r})}_{\mathrm{vert}} + \partial_1^{\mathcal{L}^*_{\neg\mathtt{r}}} p^{(\mathtt{r})}_{\mathrm{edge}} - \partial_1^{\mathcal{H}}\right)(\{v_{\mathtt{r}}, v_{\mathtt{g}}, v_{\mathtt{b}}\}) \\ &= v_{\mathtt{r}} + \partial_1^{\mathcal{L}^*_{\neg\mathtt{r}}}(\{v_{\mathtt{g}}, v_{\mathtt{b}}\}) - (v_{\mathtt{r}} + v_{\mathtt{g}} + v_{\mathtt{b}}) \\ &= 0 \in C_{\mathrm{bdry}}(\mathcal{H}). \end{aligned}$$

□

A.4 Superiority of the concatenated MWPM decoder over the projection decoder

In addition, we show that the projection decoder cannot outperform the concatenated MWPM decoder. In other words, we analytically prove that the matching obtained from the concatenated MWPM

decoder always has a weight not greater than that from the projection decoder. We here only consider the cases without boundaries.

Given a syndrome s, the projection decoder finds matchings $b_{\mathtt{r}}$, $b_{\mathtt{g}}$, and $b_{\mathtt{b}}$ in the restricted lattices such that $\partial_1^{\mathcal{L}^*_{\neg \mathtt{c}}}(b_{\mathtt{c}}) = \pi_0^{(\mathtt{c})}(\sigma)$ for each $\mathtt{c} \in \{\mathtt{r}, \mathtt{g}, \mathtt{b}\}$. It then lifts the three restricted lattice matchings to find a hypergraph matching $x \in C_1(\mathcal{H})$ such that $\partial_2^*(x) = b_{\mathtt{r}} + b_{\mathtt{g}} + b_{\mathtt{b}}$.

The following claim asserts that, for the projection and concatenated MWPM decoders sharing the same result on the red-restricted lattice, any matching on the projection decoder must also be a proper matching on the red-only lattice of the concatenated MWPM decoder. Hence, running MWPM on the red-only lattice always gives a matching whose weight is equal to or less than the wieght of any matching obtained from the projection decoder.

Claim 3. Given $\sigma \in C_0(\mathcal{H})$, $x \in C_1(\mathcal{H})$, and $b_{\mathtt{c}} \in C_1(\mathcal{L}^*_{\neg \mathtt{c}})$ for each $\mathtt{c} \in \{\mathtt{r}, \mathtt{g}, \mathtt{b}\}$, suppose

$$\begin{aligned} \partial_1^{\mathcal{L}^*_{\neg \mathtt{c}}}(b_{\mathtt{c}}) &= \pi_0^{(\mathtt{c})}(\sigma) \quad \forall \mathtt{c} \in \{\mathtt{r}, \mathtt{g}, \mathtt{b}\}, \\ \partial_2^*(x) &= b_{\mathtt{r}} + b_{\mathtt{g}} + b_{\mathtt{b}}, \\ \partial_1^{\mathcal{H}}(x) &= \sigma. \end{aligned}$$

Then $\partial_1^{\mathcal{L}^*_{\mathtt{r}}}(x) = b_{\mathtt{r}} + \left(\mathbb{1} - \pi_0^{(\mathtt{r})}\right)(\sigma)$.

Proof. The assertion is equivalent to

$$\begin{aligned} p_{\text{vert}}^{(\mathtt{r})}(x) &= \left(\mathbb{1} - \pi_0^{(\mathtt{r})}\right)(\sigma) \\ p_{\text{edge}}^{(\mathtt{r})}(x) &= b_{\mathtt{r}} \end{aligned}$$

As $\partial_2^*(\{v_{\mathtt{r}}, v_{\mathtt{g}}, v_{\mathtt{b}}\}) = \{v_{\mathtt{r}}, v_{\mathtt{g}}\} + \{v_{\mathtt{g}}, v_{\mathtt{b}}\} + \{v_{\mathtt{r}}, v_{\mathtt{b}}\} = (p_{\text{edge}}^{(\mathtt{r})} + p_{\text{edge}}^{(\mathtt{g})} + p_{\text{edge}}^{(\mathtt{b})})(\{v_{\mathtt{r}}, v_{\mathtt{g}}, v_{\mathtt{b}}\})$, applying linearity we get that $\partial_2^*(x) = (p_{\text{edge}}^{(\mathtt{r})} + p_{\text{edge}}^{(\mathtt{g})} + p_{\text{edge}}^{(\mathtt{b})})(x) = b_{\mathtt{r}} + b_{\mathtt{g}} + b_{\mathtt{b}}$. As the subspaces of red, green, and blue edges are disjoint, $p_{\text{edge}}^{(\mathtt{r})}(x) = b_{\mathtt{r}}$. Similarly, $\partial_1^{\mathcal{H}}(\{v_{\mathtt{r}}, v_{\mathtt{g}}, v_{\mathtt{b}}\}) = v_{\mathtt{r}} + v_{\mathtt{g}} + v_{\mathtt{b}} = (p_{\text{vert}}^{(\mathtt{r})} + p_{\text{vert}}^{(\mathtt{g})} + p_{\text{vert}}^{(\mathtt{b})})(\{v_{\mathtt{r}}, v_{\mathtt{g}}, v_{\mathtt{b}}\})$. By linearity, $\partial_1^{\mathcal{H}}(x) = (p_{\text{vert}}^{(\mathtt{r})} + p_{\text{vert}}^{(\mathtt{g})} + p_{\text{vert}}^{(\mathtt{b})})(x) = \sigma = [(\mathbb{1} - \pi_0^{(\mathtt{r})}) + (\mathbb{1} - \pi_0^{(\mathtt{g})}) + (\mathbb{1} - \pi_0^{(\mathtt{b})})](\sigma)$ as $\sum_{\mathtt{c}}(\mathbb{1} - \pi_0^{(\mathtt{c})}) = \mathbb{1}$. Again, as the subspaces of red, green, and blue vertices are disjoint, $p_{\text{vert}}^{(\mathtt{r})}(x) = (\mathbb{1} - \pi_0^{(\mathtt{r})})(\sigma)$. □

B Discussion on the origin of the bias between $\overline{X}$ and $\overline{Z}$ failures

In this appendix, we discuss the origin of the bias between $\overline{X}$ and $\overline{Z}$ failures and explain why the degree of this bias varies with the CNOT schedule. Errors occurring during syndrome extraction propagate to other data qubits via CNOT gates, and the propagation pattern strongly depends on the CNOT schedule. As a result, it is not surprising that failure rates are influenced by the schedule. In particular, hook errors, which are errors on ancillary qubits that occur midway through syndrome extraction and propagate to more than one data qubit, can significantly impact the decoder's performance, as they may reduce the fault distance.

To understand the origin of the bias, it is helpful to compare the optimal schedule,

$$\mathcal{A}_{\text{optimal}} = [2, 3, 6, 5, 4, 1; 3, 4, 7, 6, 5, 2], \tag{14}$$

which exhibits a very small bias (discussed in Sec. 4.3), with the schedule that produces the highest bias,

$$\mathcal{A}_{\text{biased}} = [1, 6, 7, 5, 4, 2; 2, 3, 6, 7, 5, 4],$$

where $R_{\text{bias}} \approx 0.99$ and $p_{\text{fail}}^{(X)} \approx 3p_{\text{fail}}^{(Z)}$ for $d = T = 7$, $p = 10^{-3}$. The optimal schedule has the property that its Z-part and X-part follow the same pattern, differing consistently by 1. This symmetry ensures

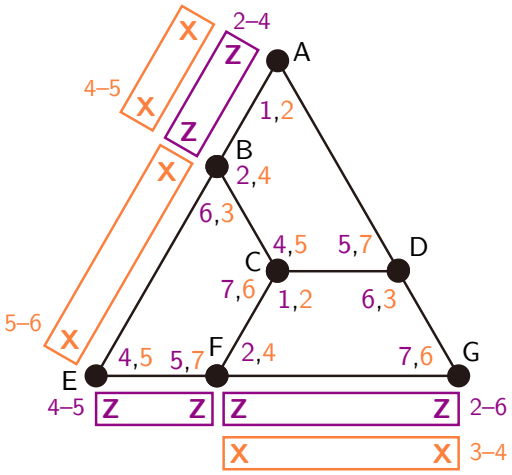

Figure 10: **Possible hook errors on the distance-3 color code patch with the CNOT schedule in Eq. (14), producing the highest bias.** Each purple (orange) box labeled 'x–y' adjacent to an edge between two qubits represents a $Z \otimes Z$ ($X \otimes X$) error on these qubits, which is caused by a Pauli-Z (X) hook error occurring between time steps x and y on the nearby Z-type (X-type) ancillary qubit.

that hook errors affect data qubits in the same way for the two ancillary qubits on a single face. In contrast, the schedule $\mathcal{A}_{\text{biased}}$ lacks such consistency; its Z- and X-parts are entirely different, resulting in hook errors exhibiting distinct characteristics for the two Pauli types.

To illustrate this effect explicitly, Fig. 10 shows a distance-3 patch for schedule $\mathcal{A}_{\text{biased}}$, with possible hook errors visualized using purple and orange boxes. For instance, the purple box adjacent to an edge connecting qubits A and B (labeled '2–4' with 'Z Z' inside) represents a $Z_{\text{A}} \otimes Z_{\text{B}}$ error caused by a hook error between time steps 2 and 4 on the nearby Z-type ancillary qubit (located on face A-B-C-D). It is evident that Z-type hook errors are more likely to occur than X-type hook errors, as they involve a greater number of time steps (e.g., $Z_{\text{F}} \otimes Z_{\text{G}}$ involves time steps 2 to 6). Consequently, $\overline{Z}$ errors are expected to be more frequent than $\overline{X}$ errors, which aligns with the numerical results. Generalizing this argument to higher code distances might be slightly more complex due to weight-6 checks, but the basic principles remain the same.

C Discussion on small-weight uncorrectable errors

In this appendix, we discuss errors with weights $< O(d/2)$ that cannot be corrected by the concatenated MWPM decoder. We first consider weight-$O(d/3)$ errors, which may be difficult to correct via the projection decoder. We verify that a specific family of weight-$O(d/3)$ errors that are uncorrectable via the projection decoder are correctable via the concatenated MWPM decoder. We then show that there exist weight-$O(3d/7)$ errors that cannot be corrected by the concatenated MWPM decoder, which implies that the value of β in the ansatz of Eq. (3) may converge to $3/7$ for sufficiently large values of d.

C.1 Small-weight errors uncorrectable via the projection decoder but correctable via the concatenated MWPM decoder

In Fig. 11(a), we schematize an error with weight $O(d/3)$ that is uncorrectable via the projection decoder [19, 20, 23]. The error is composed of a single-qubit error at the center of the triangular patch and a red string operator connecting the center and the red boundary, which is drawn as red, green, and blue solid lines. This error makes a blue check and a green check (marked as circles) violated. In the green-restricted (blue-restricted) lattice, the green (blue) check is paired up with the green (blue) boundary, thus the decoder makes a wrong prediction that causes a logical error. See Appendix A of Ref. [19] for an explicit example of such an error.

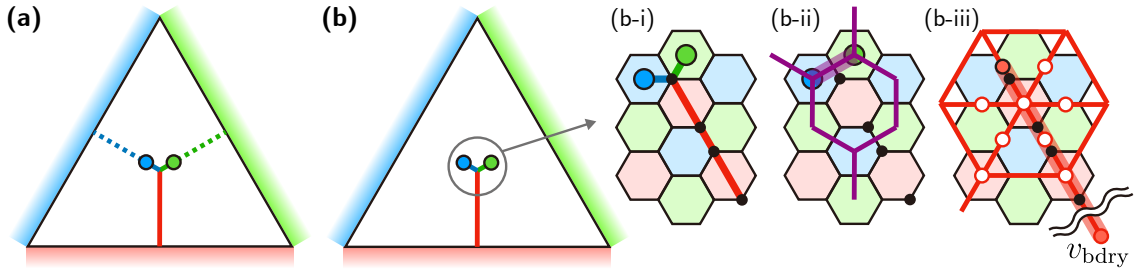

Figure 11: **Example of an error with weight $O(d/3)$ that is uncorrectable via the projection decoder but correctable via the concatenated MWPM decoder. (a)** The error is drawn as red, green, and blue solid lines, which makes a blue check and a green check marked as circles violated. The projection decoder pairs up these checks with the green/blue boundaries (dotted lines), thus it makes a logical error. **(b)** The same error is correctable via the concatenated MWPM decoder. The microscopic picture of the vicinity of the violated checks is described in (b-i), where black dots indicate qubits with errors. In (b-ii) and (b-iii), MWPMs on the red-restricted and red-only lattices are schematized with the notations used in Fig. 2, where only some of the lattice elements are visualized for simplicity. The outcome of the second-round MWPM, which is visualized as a red thick line ending at the boundary vertex v_{bdry}, coincides with the error, thus it makes no logical error.

Notably, the above error is correctable via the concatenated MWPM decoder, as described in Fig. 11(b). Roughly speaking, it is thanks to the strategy to select the least-weight one among the three predictions $\left\{\widetilde{V}^{(\mathtt{r})}_{\mathrm{pred}}, \widetilde{V}^{(\mathtt{g})}_{\mathrm{pred}}, \widetilde{V}^{(\mathtt{b})}_{\mathrm{pred}}\right\}$, which are obtained from the sub-decoding procedures of red, green, and blue, respectively. Namely, MWPM on the red-restricted and red-only lattices, which are microscopically described in (b-ii) and (b-iii), gives a correct prediction $\widetilde{V}^{(\mathtt{r})}_{\mathrm{pred}}$. Although other two predictions $\widetilde{V}^{(\mathtt{g})}_{\mathrm{pred}}$ and $\widetilde{V}^{(\mathtt{b})}_{\mathrm{pred}}$ make logical errors, their weights are $O(2d/3)$, thus $\widetilde{V}^{(\mathtt{r})}_{\mathrm{pred}}$ is selected as the final outcome of the decoder.

C.2 Small-weight errors uncorrectable via the concatenated MWPM decoder

Alike the Möbius MWPM decoder [23], the concatenated MWPM decoder fails to overcome the limitation that $O(3d/7)$-weight errors may not be correctable. We consider an error with weight $w = O(\xi d)$ for $\xi < 1/2$ as presented in Fig. 12(a), which is composed of three string operators (one for each color) that terminate at the corresponding boundaries and make one check violated for each color. These string operators have the weights of $w_{\mathtt{r}} = O(\xi_{\mathtt{r}} d)$, $w_{\mathtt{g}} = O([(\xi - \xi_{\mathtt{r}})/2]d)$, and $w_{\mathtt{b}} = w - w_{\mathtt{r}} - w_{\mathtt{g}} = O([(\xi - \xi_{\mathtt{r}})/2]d)$, where $\xi_{\mathtt{r}} \leq \xi$. Additionally, they are arranged in a way that each pair of violated checks (say, green and blue) are separated by $w_{\mathrm{gap}} = O([(1-\xi)/2]d)$ edges of the third color (red); that is, to move from the violated green check to the blue check, we need to pass at least w_{gap} red edges. (Note that a non-trivial string-net operator constructed by moving the violated blue and green checks to the violated red check and fusing them has a weight of $w + 2w_{\mathrm{gap}} = O(d)$.) When this error occurs, we face a problem that a first-round MWPM of a sub-decoding procedure can be wrong. For example, as shown in Fig. 12(b), the MWPM on the red-restricted lattice, which involves only the violated green and blue checks, gives a wrong matching connecting these checks (drawn as a dotted purple line) if and only if the distance between them (w_{gap}) is smaller than the sum of their distances to the corresponding boundaries ($w_{\mathtt{g}} + w_{\mathtt{b}}$) within the red-restricted lattice, i.e.,

$$w_{\mathrm{gap}} = O\left(\frac{1-\xi}{2}d\right) < w_{\mathtt{g}} + w_{\mathtt{b}} = O((\xi - \xi_{\mathtt{r}})d). \tag{15}$$

Then the subsequent MWPM on the red-only lattice produces predictions represented by black triangles in Fig. 12(b), leading to a logical error. Similarly, the green sub-decoding fails if and only if

$$w_{\mathrm{gap}} = O\left(\frac{1-\xi}{2}d\right) < w_{\mathtt{r}} + w_{\mathtt{b}} = O\left(\frac{\xi + \xi_{\mathtt{r}}}{2}d\right), \tag{16}$$

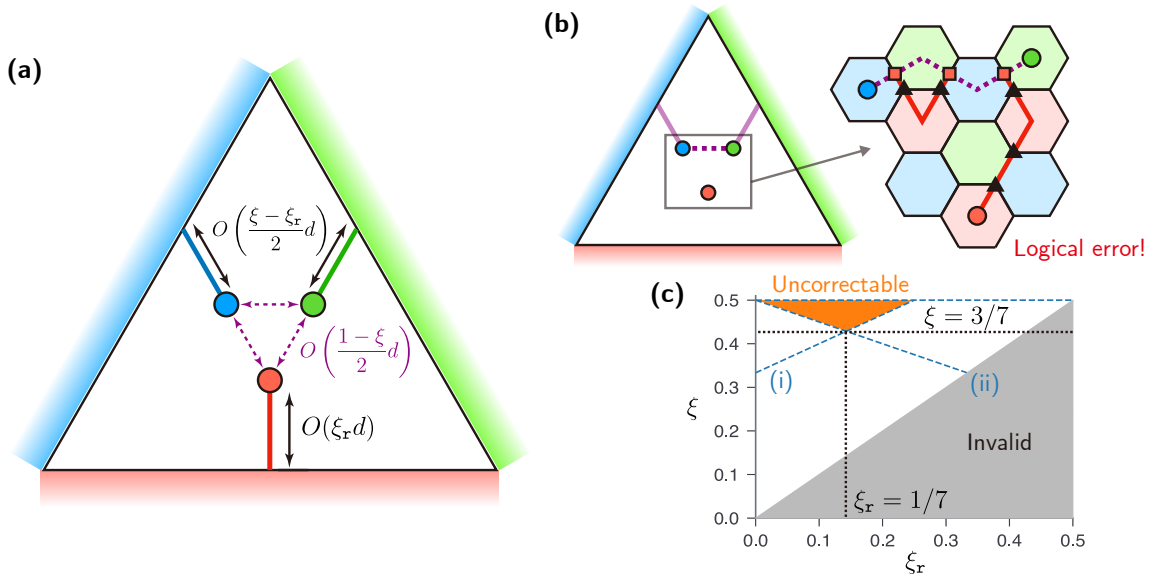

Figure 12: **Example of an error with weight $< d/2$ that may be uncorrectable via the concatenated MWPM decoder. (a)** The error has a weight of $w = O(\xi d)$ for $\xi < 1/2$ and is composed of three string operators (one for each color) terminating at the three boundaries, whose weights are respectively $w_{\mathtt{r}} = O(\xi_{\mathtt{r}} d)$, $w_{\mathtt{g}} = O([(\xi - \xi_{\mathtt{r}})/2]d)$, and $w_{\mathtt{b}} = w - w_{\mathtt{r}} - w_{\mathtt{g}} = O([(\xi - \xi_{\mathtt{r}})/2]d)$, where $\xi_{\mathtt{r}} \leq \xi$. They are arranged in a way that every pair of violated checks (say, green and blue) is separated by $w_{\mathrm{gap}} = O([(1-\xi)/2]d)$ edges of the third color (red). **(b)** If Eq. (15) holds, the first-round MWPM on the red-restricted lattice returns a wrong matching, which leads to a logical error. As an example, the microscopic picture near the violated checks is illustrated for the case of $w_{\mathrm{gap}} = 3$, where the purple dotted line indicates the matching of the first-round MWPM, the red squares are the corresponding red edges $E^{(\mathtt{r})}_{\mathrm{pred}}$, the red solid lines are the matching of the second-round MWPM, and the black triangles indicate the final prediction. Similarly, decoding on green or blue sub-lattices incurs a logical error if Eq. (16) or (17) holds. The decoding eventually fails only when all of Eqs. (15)–(17) are satisfied. **(c)** The region of ξ and $\xi_{\mathtt{r}}$ that give uncorrectable errors is visualized. The blue dashed lines (i) and (ii) respectively indicate the boundaries of the regions where Eqs. (15) and (16)/(17) hold. The gray region corresponds to $\xi_{\mathtt{r}} > \xi$, which is invalid.

and lastly, the blue sub-decoding fails if and only if

$$w_{\mathrm{gap}} = O\left(\frac{1-\xi}{2}d\right) < w_{\mathtt{r}} + w_{\mathtt{g}} = O\left(\frac{\xi+\xi_{\mathtt{r}}}{2}d\right). \tag{17}$$

Since we select the least-weight correction among the three colors, the decoding eventually fails only when all of Eqs. (15)–(17) hold. The region of such uncorrectable errors is visualized in Fig. 12(c) on the plane with the axes of $\xi_{\mathtt{r}}$ and ξ, while neglecting constant factors. The minimum of ξ in this region is $3/7$, which corresponds to $\xi_{\mathtt{r}} = 1/7$; thus, there may exist errors with weights $\geq O(3d/7)$ but smaller than $d/2$ that are uncorrectable via the concatenated MWPM decoder.

We exhaustively search for errors of the above type while increasing d under the following constraints: (i) The red string operator terminates exactly at the center qubit of the red boundary, that is, $(d-1)/2$ qubits are placed on each of the left and right sides of the center qubit along the red boundary. (ii) The weights of the blue and green string operators differ by up to two. (iii) The value of w_{gap} is the same for every pair of the violated checks. We find that $d = 25$ is the smallest code distance that permits such uncorrectable errors, as shown in the example visualized in Fig. 13, where $d = 25$, $w = 12$, $w_{\mathtt{r}} = w_{\mathtt{g}} = w_{\mathtt{b}} = 4$, and $w_{\mathrm{gap}} = 7$.

D Additional numerical analyses

In this appendix, we present the results of additional numerical analyses.

Figure 14 compares our color-selecting strategy (i.e., executing the sub-decoding procedures for all the three colors and selecting the best one) with two other strategies of performing only one or two

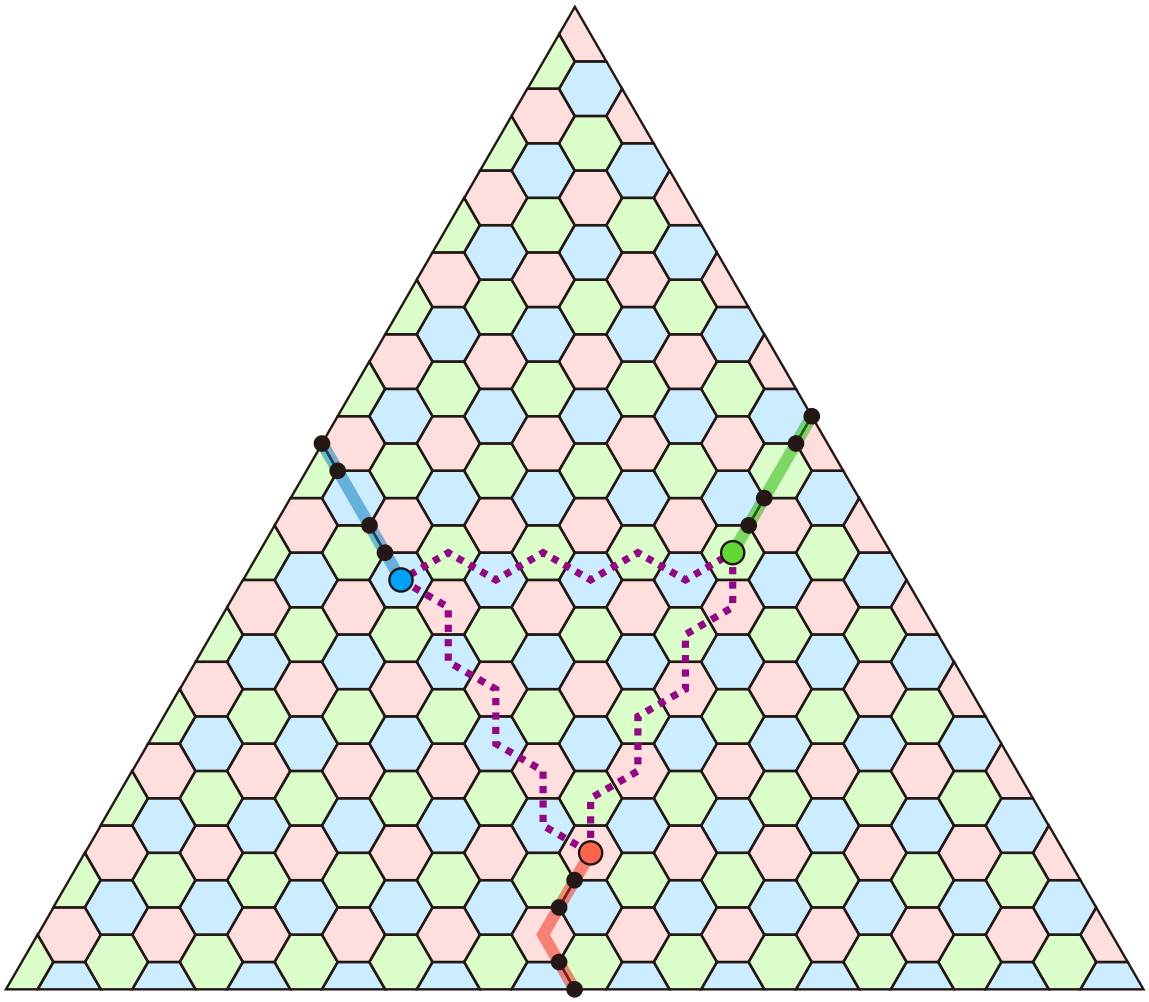

Figure 13: **Smallest example of an uncorrectable error shown in Fig. 12.** The error, marked as black dots, lies on the code of $d = 25$, which is the smallest possible code distance permitting such errors, and has the weight of $w = 12 < d/2$. Three string operators (colored thick lines) constituting the error have the same weight of $w_{\mathtt{r}} = w_{\mathtt{g}} = w_{\mathtt{b}} = 4$ and each pair of violated checks (colored circles) is separated by $w_{\mathrm{gap}} = 7$ edges. The concatenated MWPM decoder cannot correct this error since it satisfies all of Eqs. (15)–(17).

sub-decoding procedures. It plots logical failure rates under the bit-flip noise model of $p \in \{0.02, 0.04\}$ for these three strategies, confirming that our original strategy of using all the three colors significantly outperforms the others.

Figure 15 presents separate analyses for the $\overline{Z}$- and $\overline{X}$-failure of the decoder under near-threshold circuit-level noise, which are analogous to Figs. 7(b) and (c). The long-term cross thresholds $p^{\times,\mathrm{LT}}_{\mathrm{circuit}}$ are estimated as 0.460% for the $\overline{Z}$-failure and 0.452% for the $\overline{X}$-failure, which differ by about 1.8% (0.008%p). Note that $p^{\times,\mathrm{LT}}_{\mathrm{circuit}}$ for the overall failure rate is about 0.455%, which is the average of these two.

Figure 16 presents separate analyses for the $\overline{Z}$- and $\overline{X}$-failure of the decoder under sub-threshold circuit-level noise, which are analogous to Fig. 8. For the $\overline{Z}$-failure, the regressions of $G(d)$ and $C(d)$ against d and the corresponding ansatz parameters are

$$G(d) = (0.48 \pm 0.07)(d - 12) + (5.4 \pm 0.4), \quad C(d) = (2.7 \pm 0.5)(d - 12) + (25 \pm 3),$$
$$p^{\star}_{\mathrm{circuit}} \approx 3.4 \times 10^{-3}, \quad \alpha \approx 5.0 \times 10^{-3}, \quad \beta \approx 0.48, \quad \eta \approx 5.4, \quad d_0 = 12.$$

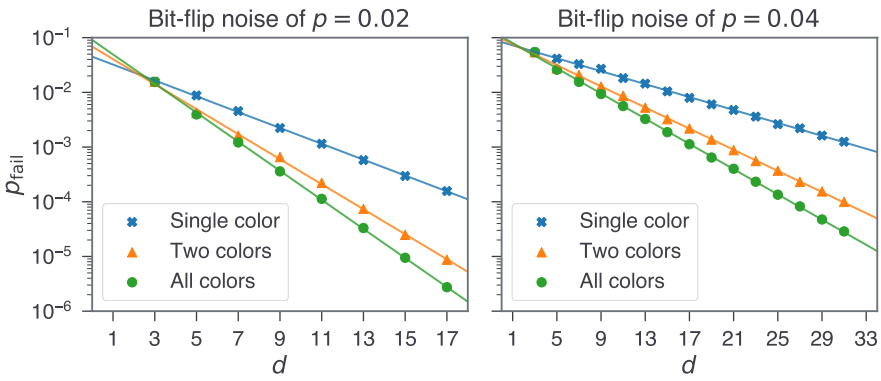

Figure 14: **Comparison of different color-selecting strategies.** We consider three strategies for the color-selecting step of the concatenated MWPM decoder: using only the outcome of the sub-decoding procedure of a single color (r) without a comparison, comparing the outcomes for two colors (r, g), and comparing the outcomes for all the three colors (which is the original strategy). The failure rates p_{fail} under the bit-flip noise models of $p = 0.02$ (left) and $p = 0.04$ (right) are plotted against d for these three strategies. The solid lines indicate the linear regressions of $\log p_{\text{fail}}$ against d: $\log p_{\text{fail}} = -0.33d - 3.1$ ($p = 0.02$, single color), $-0.53d - 2.7$ ($p = 0.02$, two colors), $-0.61d - 2.4$ ($p = 0.02$, all colors), $-0.14d - 2.5$ ($p = 0.04$, single color), $-0.22d - 2.4$ ($p = 0.04$, two colors), and $-0.27d - 2.3$ ($p = 0.04$, all colors), respectively.

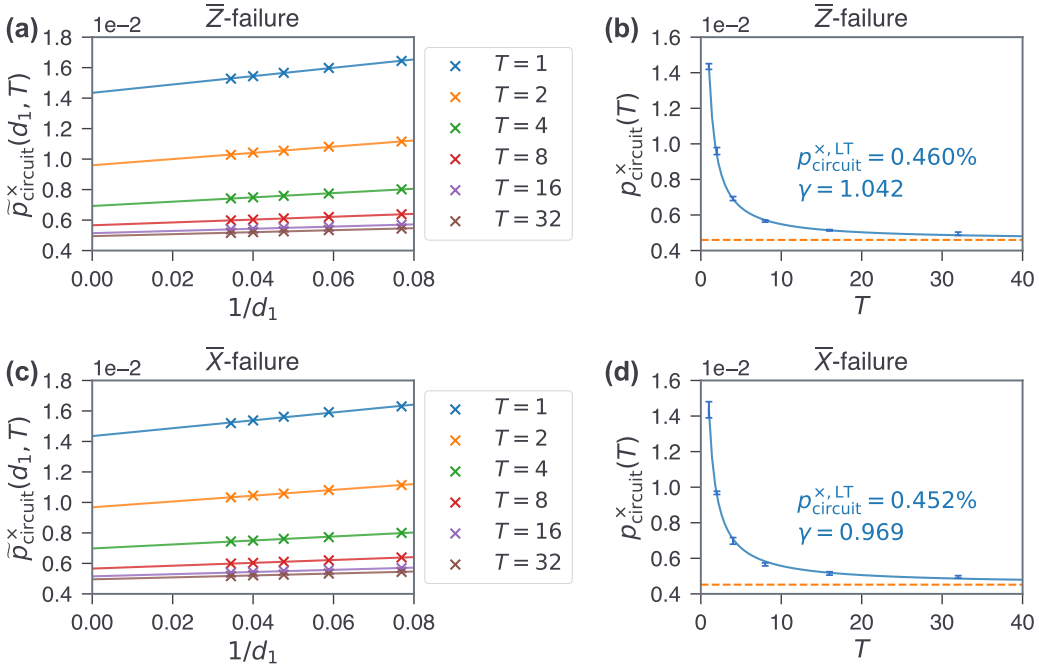

Figure 15: **Separate analyses for the $\overline{Z}$- and $\overline{X}$-failure of the decoder under near-threshold circuit-level noise.** For the $\overline{Z}$-failure, **(a)** and **(b)** are the counterparts of Figs. 7(b) and (c), respectively. Similarly, **(c)** and **(d)** are their counterparts for the $\overline{X}$-failure. The long-term cross thresholds $p^{\times,\text{LT}}_{\text{circuit}}$ are estimated as 0.460% for the $\overline{Z}$-failure and 0.452% for the $\overline{X}$-failure, which differ by about 1.8%.

For the $\overline{X}$-failure, they are

$$G(d) = (0.53 \pm 0.05)(d - 12) + (5.5 \pm 0.3), \quad C(d) = (3.1 \pm 0.5)(d - 12) + (26 \pm 3),$$
$$p^{\star}_{\text{circuit}} \approx 3.0 \times 10^{-3}, \quad \alpha \approx 3.4 \times 10^{-3}, \quad \beta \approx 0.53, \quad \eta \approx 5.5, \quad d_0 = 12.$$

Figure 17 analyzes the bias between $\overline{Z}$- and $\overline{X}$-failure in all the circuit-level simulation outcomes of Fig. 7 and 8. It shows two scatter plots: one for the $\overline{Z}$-failure rate $p^{(Z)}_{\text{fail}}$ and the $\overline{X}$-failure rate $p^{(X)}_{\text{fail}}$ and the other one for the total failure rate p_{fail} and the bias R_{bias} defined in Eq. (8). We observe that

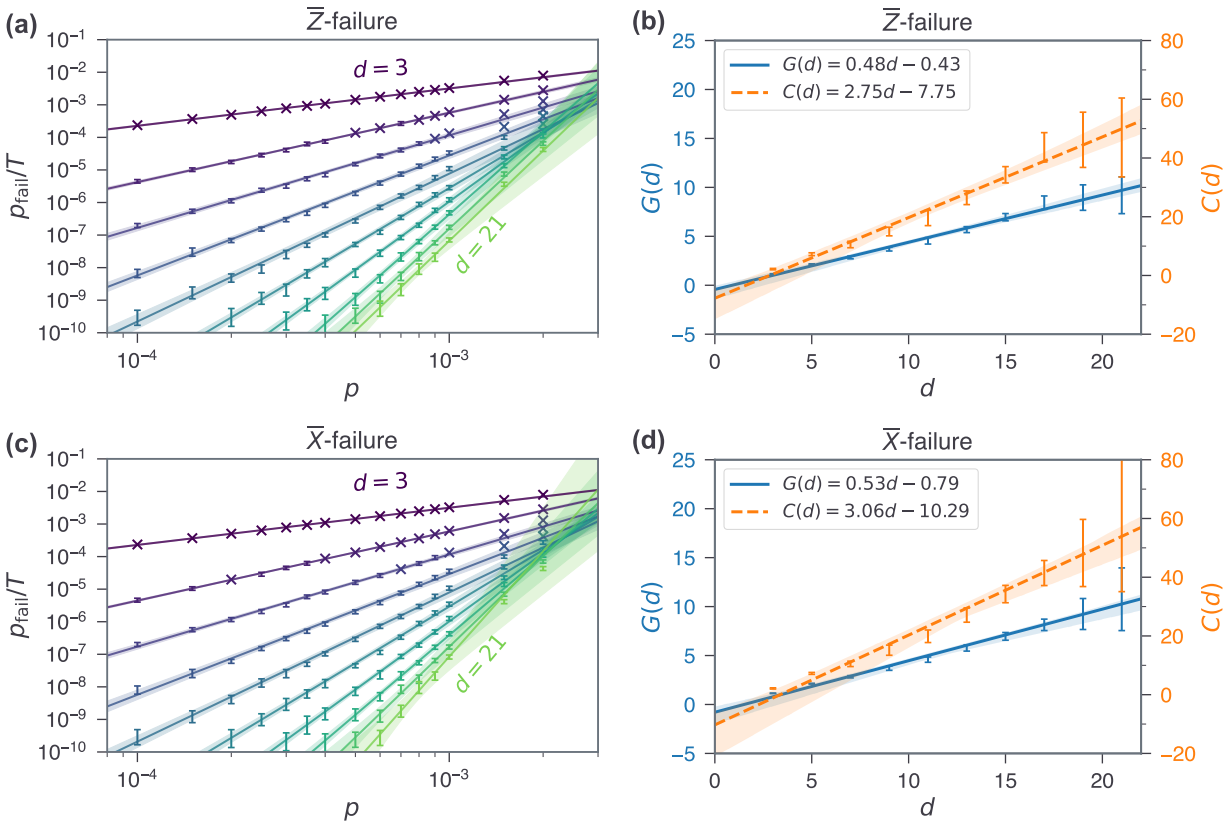

Figure 16: **Separate analyses for the $\overline{Z}$- and $\overline{X}$-failure of the decoder under sub-threshold circuit-level noise.** For the $\overline{Z}$-failure, **(a)** and **(b)** are the counterparts of Figs. 8(a) and (b), respectively. Similarly, **(c)** and **(d)** are their counterparts for the $\overline{X}$-failure.

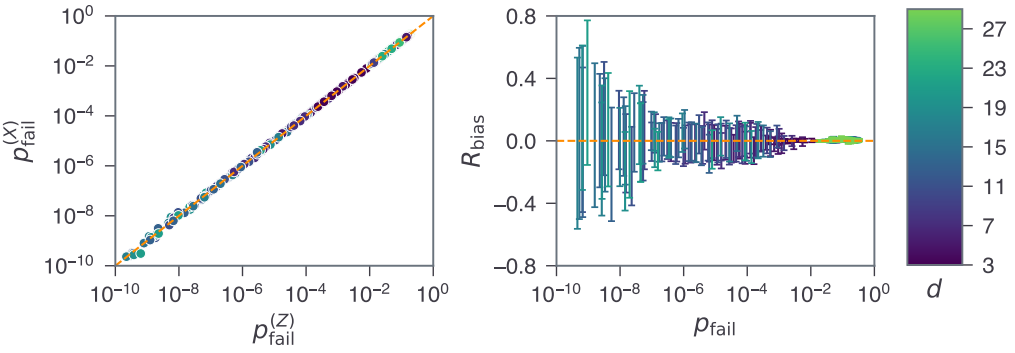

Figure 17: **Analysis of bias between $\overline{Z}$- and $\overline{X}$-failure under circuit-level noise.** The left figure shows the scatter plot of the $\overline{Z}$-failure rate $p_{\text{fail}}^{(Z)}$ and the $\overline{X}$-failure rate $p_{\text{fail}}^{(X)}$ for all the simulation outcomes in Figs. 7 and 8. The right figure shows the scatter plot of the logical failure rate p_{fail} and the bias R_{bias} defined in Eq. (8) for all the simulation outcomes in Figs. 7 and 8. The error bars indicate the 99% confidence intervals of R_{bias}. For both figures, the dots and error bars are colored depending on the code distance d (as shown in the right color bar) and the orange dashed lines represent unbiased cases of $p_{\text{fail}}^{(Z)} = p_{\text{fail}}^{(X)}$.

the 99% confidence intervals of R_{bias} almost always include zero, which implies that we cannot reject the null hypothesis that there is no bias.